Zammad

Zammad

Mar 13, 2023

MANAGE

1 Users **1**
- 1.1 Managing users via the admin panel . . . 1
- 1.2 Managing users via CSV Import . . . 2
- 1.3 LDAP / Active Directory . . . 4
- 1.4 Exchange . . . 7
- 1.5 Reference Guide: User Details . . . 9

2 Groups **13**
- 2.1 Group Settings . . . 13
- 2.2 Group Access Levels . . . 15

3 Roles **19**
- 3.1 What Is a Role? . . . 20
- 3.2 Reference Guide: Permissions . . . 21
- 3.3 Role Details . . . 29

4 Organizations **31**
- 4.1 Managing organizations via the admin panel . . . 32
- 4.2 Managing organizations via CSV Import . . . 32
- 4.3 Reference Guide: Organization Details . . . 34

5 Overviews **37**

6 Text Modules **41**
- 6.1 Delete or clone text modules . . . 43
- 6.2 Import of text modules via CSV file . . . 43

7 Macros **45**
- 7.1 Learn by example . . . 45
- 7.2 How do they work? . . . 47

8 Templates **49**
- 8.1 Managing templates . . . 50

9 Tags **51**
- 9.1 How to tag . . . 51
- 9.2 Tag-Overviews . . . 51
- 9.3 Tag Management Area . . . 52

10 Calendars **53**

11 SLAs **55**
11.1 Learn by example . . . 58
11.2 How do they work? . . . 62

12 Triggers **65**
12.1 Learn by example . . . 66
12.2 How do they work? . . . 69
12.3 Limitations . . . 71
12.4 System Notifications . . . 71

13 Public Links **73**
13.1 Adding new public links . . . 73
13.2 Manage existing public links . . . 74

14 Webhook **79**
14.1 How does it work? . . . 79

15 Scheduler **89**
15.1 Add a new scheduler . . . 89

16 Report Profiles **93**
16.1 Further information about the reporting: . . . 93

17 Time Accounting **97**
17.1 How it works . . . 97
17.2 Reviewing accounted time . . . 98

18 Knowledge Base **101**
18.1 Features . . . 102
18.2 Setup . . . 102

19 Web **109**
19.1 Settings affecting your customer . . . 109
19.2 Global settings affecting *all* users . . . 110

20 Forms **111**
20.1 Limitations . . . 111
20.2 Settings . . . 112
20.3 Designer . . . 112
20.4 Requirements . . . 114
20.5 Potential Spam Issue . . . 116

21 Email **117**
21.1 Accounts . . . 117
21.2 Filters . . . 125
21.3 Signatures . . . 127
21.4 Settings . . . 129
21.5 Email header manipulation . . . 132
21.6 Extra Options for Self-Hosted Users . . . 136

22 Chat **139**
22.1 How to improve Support by Chat? . . . 139
22.2 Our answer: The Zammad Chat Widget . . . 139
22.3 Configuration of the Chat widget . . . 140

23 Google **145**
23.1 Accounts . . . 145
23.2 Filters . . . 154
23.3 Signatures . . . 156
23.4 Settings . . . 158
23.5 Email header manipulation . . . 161

24 Microsoft 365 **167**
24.1 Accounts . . . 167
24.2 Filters . . . 178
24.3 Signatures . . . 180
24.4 Settings . . . 181
24.5 Email header manipulation . . . 184

25 Twitter **189**
25.1 1. Apply for a Twitter Developer account . . . 190
25.2 2. Create a new Twitter app for Zammad . . . 191
25.3 3. Set your new app's permissions . . . 192
25.4 4. Generate a new access token & secret . . . 192
25.5 5. Set up a dev environment . . . 194
25.6 6. Add your new Twitter app in Zammad . . . 194
25.7 7. Add your Twitter account in Zammad . . . 194
25.8 8. Configure filters . . . 194

26 Facebook **197**

27 Telegram **211**
27.1 Register a Telegram bot app . . . 211
27.2 Configure Zammad as Telegram bot . . . 213
27.3 Start using your new channel . . . 214

28 Branding **221**

29 System **225**
29.1 Base . . . 225
29.2 Services . . . 226
29.3 Storage . . . 226
29.4 Network . . . 227
29.5 Frontend . . . 227
29.6 Console based settings . . . 228

30 Security **231**
30.1 Base . . . 231
30.2 Password . . . 235
30.3 Third-Party Applications . . . 236

31 Ticket **261**
31.1 Base . . . 261
31.2 Number . . . 262
31.3 Auto-Assignment . . . 263

32 Subscription (SaaS) **265**
32.1 Plan . . . 265
32.2 Billing . . . 267
32.3 FAQ . . . 269

33 Integrations **271**
33.1 Integrations for phone systems . . . 271
33.2 Integrations for authentication and customers . . . 288
33.3 Integrations for Monitoring Systems . . . 295
33.4 Integrations for Issue Trackers . . . 305
33.5 Other Integrations . . . 307

34 Objects **319**
34.1 Object types . . . 320
34.2 Object permissions . . . 328
34.3 Ordering objects . . . 329
34.4 Updating database after adding or editing objects . . . 330
34.5 System objects . . . 331

35 Core Workflows **333**
35.1 Learn by example . . . 333
35.2 How do they work? . . . 339
35.3 Limitations . . . 343

36 Variables **345**
36.1 Variable Categories . . . 345
36.2 Wait, what about custom objects? . . . 354

37 Translations **357**
37.1 How do I get the most recent translations? . . . 357
37.2 But… I have custom objects?! . . . 357
37.3 So how does this local translation work? . . . 358

38 Data Privacy **361**
38.1 Deleting Users . . . 362
38.2 Frequently Asked Questions . . . 363

39 Maintenance **365**

40 Monitoring **369**

41 Packages **373**

42 Sessions **375**

43 Version **377**

44 Composer Settings **379**

CHAPTER

ONE

USERS

Depending on your organization's IT capabilities, users can be managed individually or in bulk, or even synchronized with third-party directory services.

Fig. 1: The simplest way to manage users is directly in the Admin Panel.

Learn more about managing users…

1.1 Managing users via the admin panel

The "Users" panel provides tools to manually manage user accounts.

Creating and editing users

Fig. 2: Click the **New User** button to open the New User dialog, or click on an existing user to edit.

Hint: See *Reference Guide: User Details* for help with the New/Edit User dialog.

Deleting users

Fig. 3: Use the **Actions** menu to open the **Delete User** dialog.

Warning: Deleting a customer destroys all their associated tickets!

To learn more, see *Data Privacy*.

Filtering the user list

Note: Known bug

The search list displays up to 50 users, from newest to oldest. That means that if there are more than 50 results, the user you're searching for might not be shown.

This is a known bug with a fix underway.

Fig. 4: Use the **Search for users** bar and the **Roles** buttons to filter the list. You may search by name, email, or any other user attribute.

Unlock locked user accounts

Fig. 5: Use the **Actions** menu to unlock accounts after too many failed logins. Locked accounts are indicated with a lock icon on the left side.

Taking over a user's session

Fig. 6: Use the **Actions** menu to **View from user's perspective**.

The **View from user's perspective** button allows you to "hijack" another user's session and confirm firsthand what they can and can't do (or see) when they're logged in.

This is especially useful when you need to verify that you've set up custom permissions correctly for non-admin users.

> **Danger: With great power comes with great responsibility.**
>
> This feature is not a simulation; entering this mode will boot the user from their session, and any actions you take (responding to tickets, changing passwords, *logging hours worked*) will actually be performed from the user's account.
>
> (On the other hand, if the user logs back in, you'll be booted, too.)

Hint: When finished, use the **Back to my view** × button at the top of the page. If you try to exit by logging out, the "hijacked" user session will be restored when you log back in.

1.2 Managing users via CSV Import

If your organization has information about customers (or personnel) already stored in a directory system that can export to CSV, you can batch-import them into Zammad in just three steps.

Hint: CSV import provides one-off batch processing of user records. For persistent, automated user synchronization, consider integration with a third-party directory system like *LDAP / Active Directory* or *Exchange*.

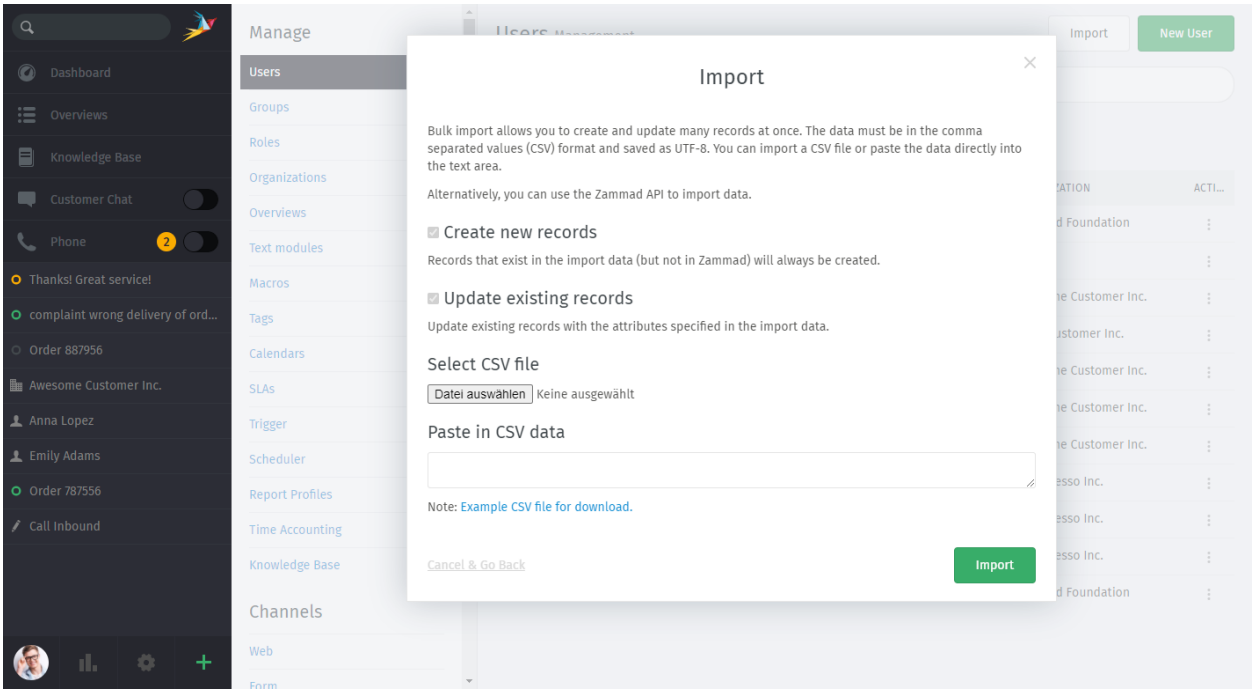


Fig. 7: Use the **Import** button to open the CSV import dialog.

1.2.1 Step 1: Inspect the sample .csv

Use the link at the bottom of the CSV import dialog (**"Example CSV file for download"**) to see how Zammad expects you to format your user data.

1.2.2 Step 2: Export your user data to .csv

Prepare your .csv file according to the format found in Step 1 above. Note that:

- the `id` attribute (column) should be left blank or removed entirely;
- the `firstname` and `lastname` attributes are **required**; and
- any other columns may be safely omitted as long as each row has the same number of fields (commas).

For instance:

```
login,firstname,lastname,email,web,active
jdoe,"John","Doe",jdoe@example.com,"https://zammad.com",true
mmore,"Madeline","Moore",mmore@example.net,"",false
```

1.2.3 Step 3: Import your .csv to Zammad

Once your raw user data has been submitted, Zammad will perform a test run to compute the number of affected records:

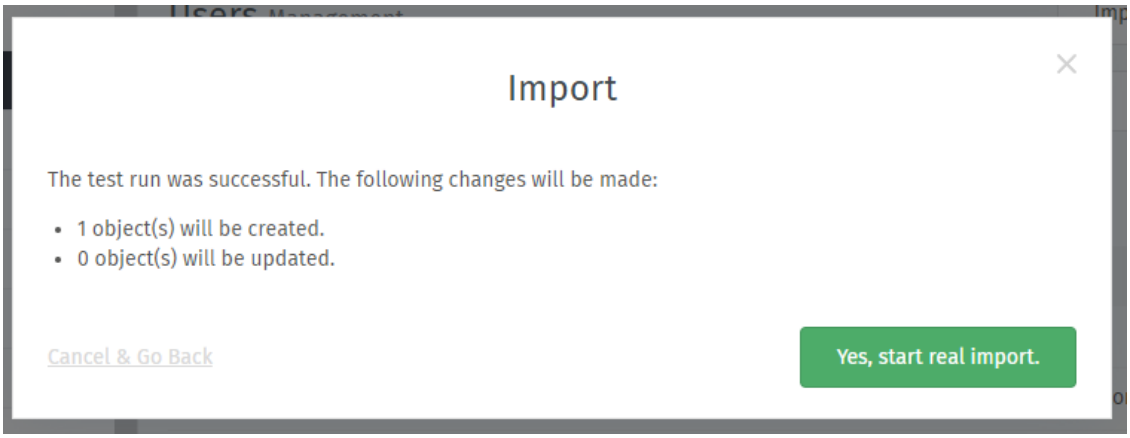


Fig. 8: CSV import always begins with a preview / test run.

Note: How does it know when to create new records and when to update?

Records are updated when the imported data contains an `email` or `login` that matches an existing user account.

Click **Yes, start real import** to proceed. (If you're importing a lot of records, be patient—it may take a minute.)

That's it!

1.3 LDAP / Active Directory

Zammad comes with a powerful LDAP integration that allows you to have a single source of truth. By this you reduce the number of login credentials your user has to remember.

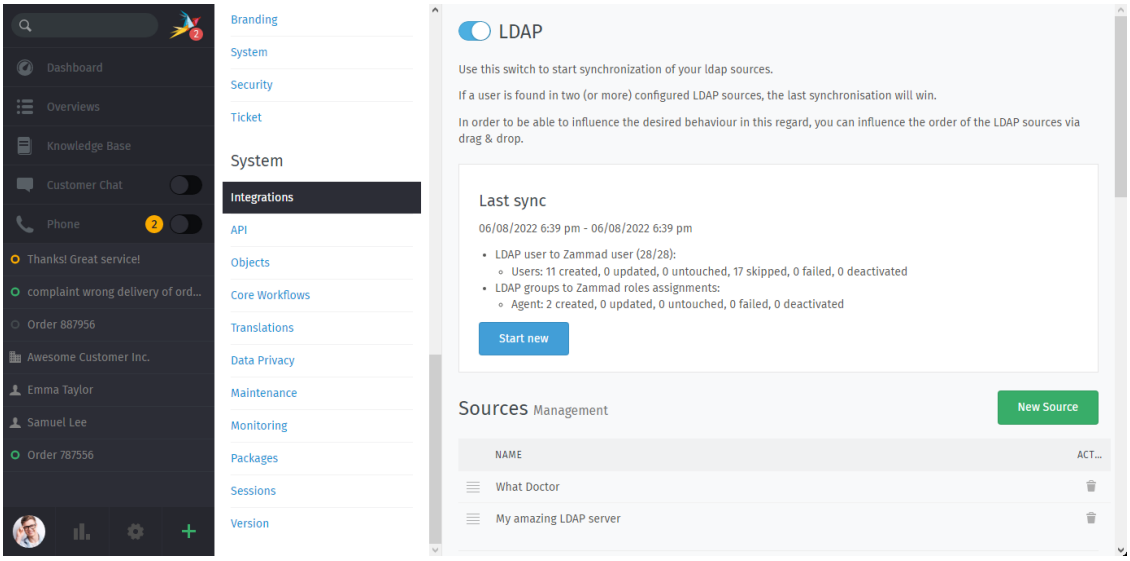
Hint: The LDAP source is also a perfect candidate for Zammad's Kerberos Single Sign-On but also works as addition to other *Third-Party Applications*.

1.3.1 Manage LDAP-Sources

Hint: Please keep in mind all current limitations for the LDAP synchronization. This will help to understand scopes of the following operations better.

Add new source Using the *New Source* button allows you to add new LDAP sources to your installations. You're not limited in the number of sources, however, keep in mind that many sources will also take more time to synchronize.

If you want to use LDAPs, type `ldaps://` instead of `ldap://` in front of your hostname. You can also tell Zammad to use a different port by appending `:<port number>`.



When providing a LDAPs source, Zammad will display an additional option *SSL verification* that allows you to disable the verification for e.g. self-signed SSL certificates.

Tip: Using an user filter can be a good idea if you only require a small subset of your LDAP users in Zammad.

This especially affects inactive users As every LDAP behaves differently on which attributes and how they are set, Zammad doesn't care about any flags.

Users that no longer are returned by your LDAP source will automatically be set to inactive. Zammad assumes that the user was deactivated.

Users will never be removed automatically! If you want to remove obsolete users, use *Data Privacy*.

Active Directory admins Active Directories are quite specific on how to filter for active users only - please see the Microsoft documentation for more information.

Danger: *Do not* manually write pathes of either LDAP attributes or groups. If Zammad does not display them, it either cannot find them or you have a lot of users that don't have the attributes populated.

Zammad will always only return attributes that are filled - this reduces the returned list of attributes greatly.

Review or edit existing source Clicking on a LDAP source will provide a configuration and mapping overview.

If needed you can then use the *Change* button to update either the name, active state or the whole configuration. If you're changing the whole configuration, the dialogue will be identical to the source creation.

Note: Did your LDAP server change? Different LDAP servers have different structures and default attributes. This causes the LDAP synchronization to likely fail. Consider removing the affected source and re-add it.

Re-arrange LDAP source's order You can change the synchronization order for each source at any time. Zammad will synchronize the sources from top to bottom.

Remove a source If you no longer need the a LDAP source or simply want to start over, you can remove them at any time. This will not remove synchronized users, data synchronized or permissions.

Tip: Not sure if you'd need the source later on? Set the source to inactive instead of removing it - see *Review or edit existing source* for more.

Manage LDAP-Sources Add, modify, remove or re-arrange your LDAP-Sources as needed.

1.3.2 Limitations

Before you continue, please note the following limitations.

- Mapping / Synchronizing organizations is not possible

  Tip: You may want to consider using domain based assignments to overcome this issue. Learn more on *Organizations*.

- Zammad's LDAP sync is one way. Editing user settings or permissions may be overwritten upon the next sync depending on your configuration.
- Nested groups are not supported.
- Synchronizing user avatars from LDAP is not supported.
- Unlike user filters, group filters cannot be changed.
- When a user originates from an LDAP server, Zammad will try to verify the login credentials against LDAP first - if this fails Zammad will check its local database.

  Warning: Users can have local passwords even if they're LDAP users! You can learn more about user accounts in general on *Users*.

- When several LDAP sources contain the same user (meaning the same email address), the user in question will be updated with every source configured. *The last LDAP source will win.*

  This is subject to change with Issue 4109 in the future.

- Synchronization statistics currently affect *all* configured LDAP sources. This also applies for newly added or updated sources.

  This is subject to change with Issue 4108 in the future.

- Zammad currently has limited fallback server support. You can workaround this by providing several sources - however, ensure to have the exact same configuration on your fallback.

  This is subject to improve with Issue 4107 in the future.

1.3.3 Recent Logs

This section holds all requests Zammad handled for all LDAP sources. These entries can either contain synchronization information or logins (authentication attempts via Zammad's login interface).

By clicking on any request, Zammad will provide even more information. The provided information can be useful when something does not work as expected.

Note: Especially a LDAP synchronization can cause many log entries. The web interface will always limit the number of shown entries to the last 50 entries.

1.4 Exchange

With our Exchange integration, you can easily use existing address books without having to update more than one source.

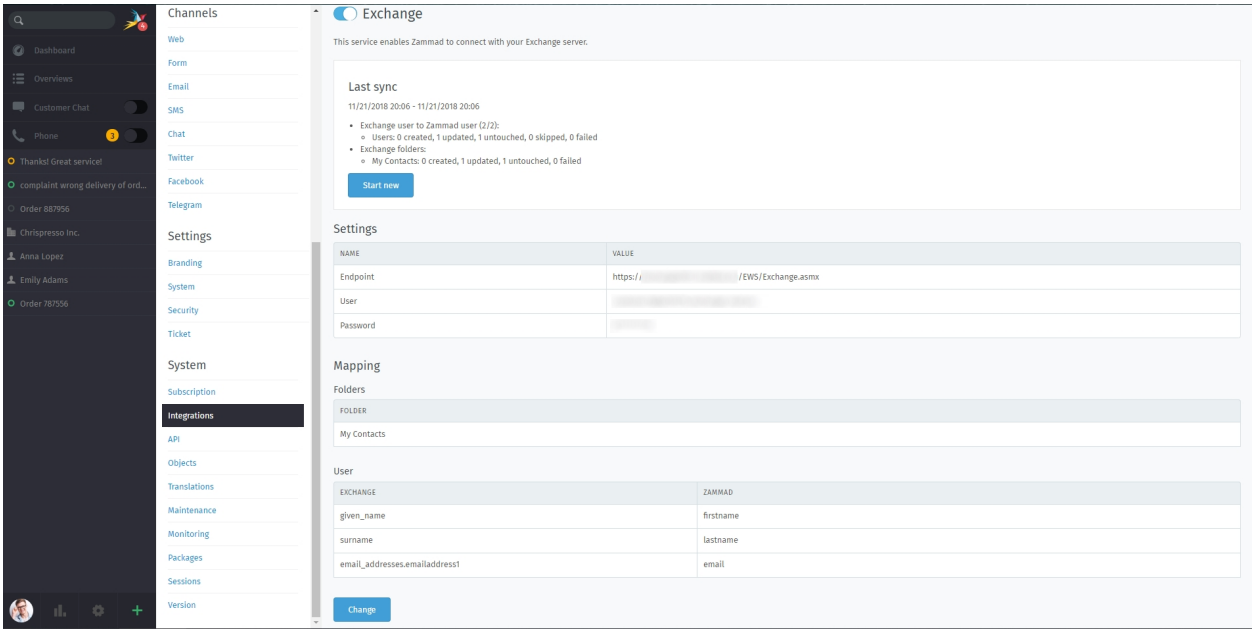
> **Warning:** The exchange sync can be overruled by Zammad's LDAP integration. If you have the same users in both sources, the LDAP version will always be Zammad's pick.

Note: The Exchange sync is one way: Exchange => Zammad. Changes to your users inside of Zammad might be overwritten by the Exchange Sync.

To configure Exchange integration, simply go to the System -> Integrations -> Exchange in the admin panel. Press "change" and follow the wizard for adding the needed Exchange information to Zammad. On the last two step Zammad will ask you for the address book(s) and your wanted Attribute mapping. By default, Zammad only Maps email address, First- and Lastname. Technically you can map any Exchange object to a Zammad user object (this also works for Custom Objects!).

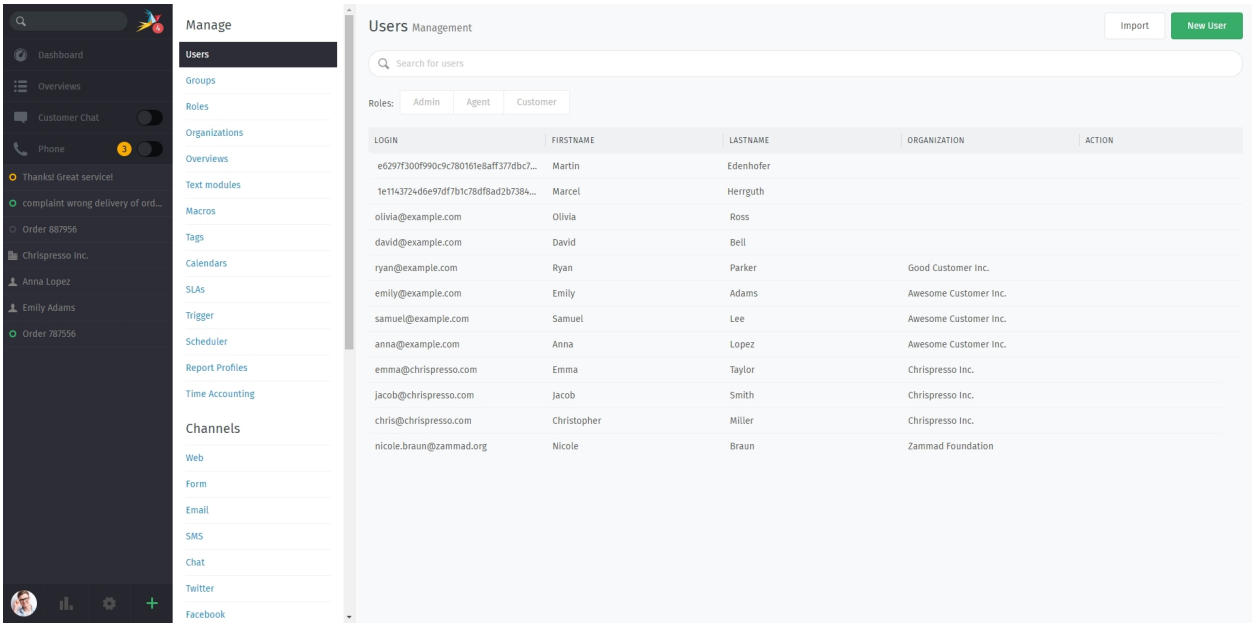
Note: Please refrain from syncing all addresses, as the results may not be what you expect (Exchange collects huge amounts of addresses). A central address book of your company to sync makes more sense, as you can ensure that Zammad gets only the data you need and want.

After pressing Continue, Zammad will check if the configuration is okay. You can then enable Exchange and start your first sync. The sync will then run hourly - if you need to change mappings or the address book(s), you can change the configuration at any time.



After the sync has finished, you can find the new Exchange contacts under "Users". Zammad integrates them just normal users.

Note: In some cases you might see unique IDs as "Login" instead of the email address. This is normal and doesn't affect the login or email mapping for that entry.



Note: Customers get their own user accounts, too?

Yes! Unlike e.g. OTRS, Zammad needs to store accounts for *everyone* who communicates through the system.

Why? It helps us do things like show all tickets from a certain customer.

How? Zammad checks the sender of every incoming message at every inbox it monitors, and if it doesn't recognize the

address, **poof**—new customer account!

(Your customers never need to set a password. Of course, they can if they want to, but the account will be there even if they never use it.)

1.5 Reference Guide: User Details

Most of the attributes you can set on user accounts are self-explanatory. The ones that aren't are described below.

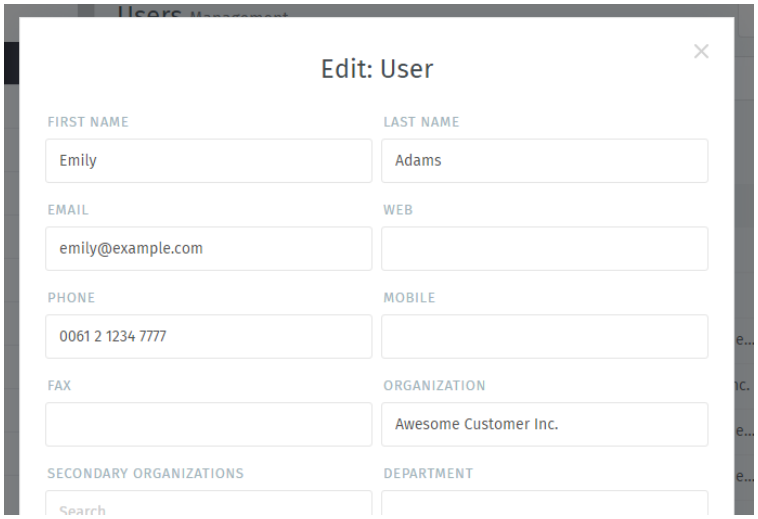


Fig. 9: User details can be set in the **New/Edit User** dialog.

Note: Admins aren't the only ones who can change these settings.

In most cases, agents can, too (using the new ticket dialog, search bar or the ticket pane).

Login A user's email and login may differ, but **either one can be used to sign in.**

Note:

Fig. 10: User logins are **not** shown in the New/Edit User dialog, but they are visible from the user overview.

This attribute **cannot** be set via the Admin Panel. Instead, use the Zammad console, the REST API, or *CSV import*.

Password Yes, administrators really do have the power to change other users' passwords.

(Agents do not, though.)

Organization *Organizations* are a way to group customers together (usually, members of the same company). This allows you to do things like view all tickets for that company or set up special *Triggers* that fire only for those customers.

Hint: You can't assign a customer to an organization that doesn't exist yet.

To add one, go to **Manage > Organizations** in the Admin Panel.

Secondary Organizations This option allows you to assign more organizations, in addition to the user's primary organization.

Secondary organizations behave the same like the primary ones with one exception: Secondaries are not as highlighted like their primaries.

Hint: Listings for all organizational tickets are not affected by this. Zammad will mix primary and secondary organization tickets together.

Warning: While the number of secondary organizations is not limited directly, you may want to keep this to a reasonable number of organizations.

30-40 organizations at maximum *should* be good enough.

VIP This flag is a way for your team to indicate high-status customers. Just as with organizations, you can set up special *Triggers*, *Scheduler* jobs, *SLAs*, and *Overviews* just for VIPs.

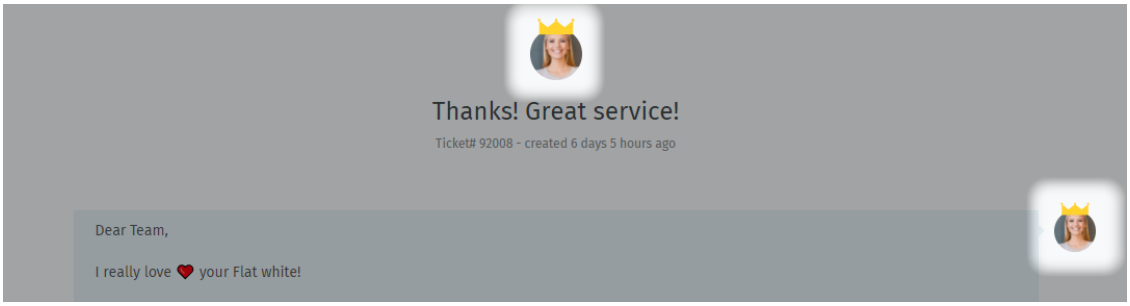


Fig. 11: VIPs are displayed with a crown above their avatars.

Note Notes are visible to all staff members, **including agents**.

Hint: Are you using the Note field to keep track of your own "custom" user attributes?

Wish you could add your own fields to the New/Edit User dialog?

You can! To learn more, see *Objects*.

Active Disabling this flag is a soft alternative to deleting a user. So what's the difference?

1. There is no way to restore a deleted user; inactive users can be reactivated at any time.
2. When a user is deleted, all their associated tickets are lost, as well; deactivating a user keeps all associated tickets intact.
3. Inactive users still appear in search results:

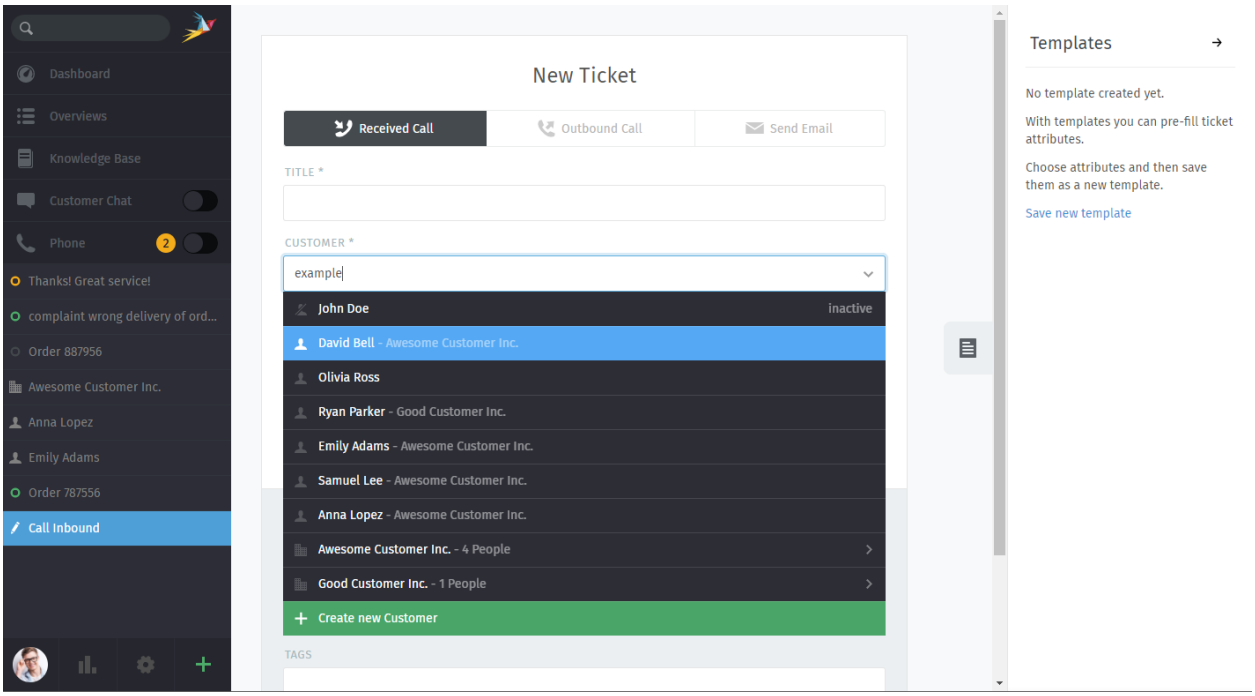


Fig. 12: A slashed-out icon indicates an inactive user. In other cases, inactive users are greyed out.

Permissions Under this heading, you can manage two separate (but related) user details:

1. *Roles* dictate what users can do in the system. If you need to grant someone privileges to edit the knowledge base or access part of the admin panel, roles are the answer.
2. *Group Access Levels* dictate which tickets an agent can work with. If someone's not receiving notifications for incoming tickets or can't be assigned a ticket, group access levels are likely to blame.

Hint: Huh? I don't see the group access table…

The group access table is only visible in **agent profiles**, when there is **more than one active group** in the system.

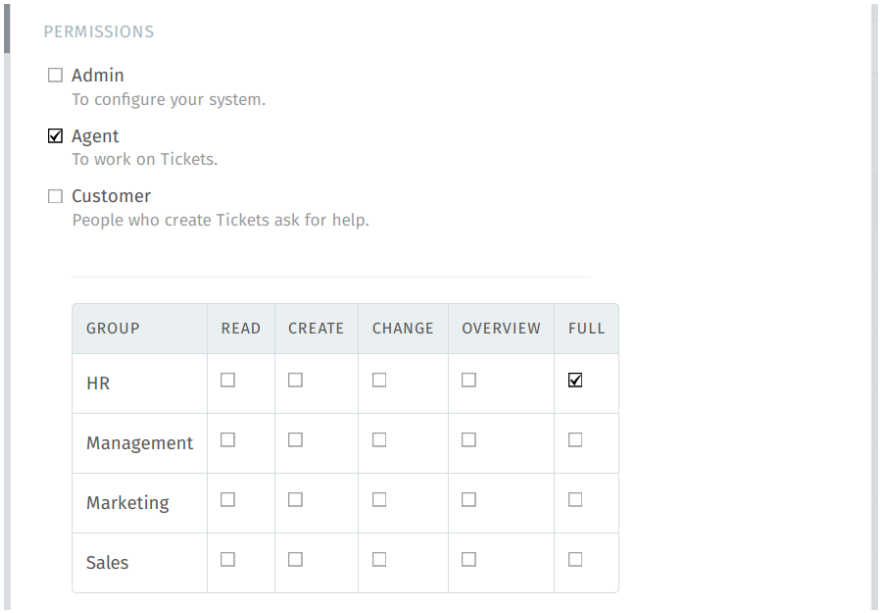


Fig. 13: **Top:** A user's roles decide what kinds of actions they can perform and which *groups* they belong to. **Bottom:** Group assignments can alternately be set on a per-user basis.

CHAPTER

TWO

GROUPS

This is the group management area. From here you can edit existing groups and add new groups.

Groups in Zammad are similar to working groups that deal with different topics within a company. For example, the tickets relevant to the sales department might be available in the *Sales* group, while the tickets for the support department might be available in the *Support* group. These are just examples; how you structure your groups is up to you.

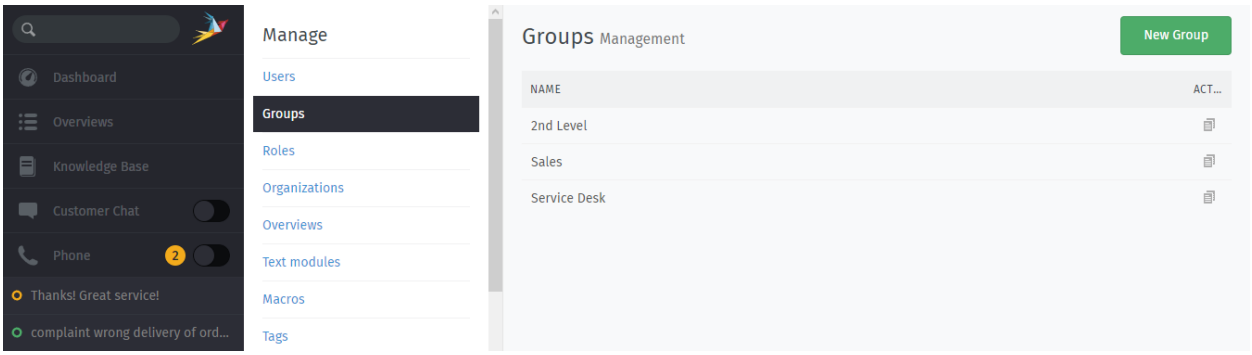
Tickets enter Zammad through various channels (e.g. via email) and are then sorted into these groups. The tickets (cases) are thus made available to the agents responsible for the group. Each ticket can only belong to one group, and you can decide via **access levels** (see below) what access your agents have in each group. For example, you might want set up a group *Management* for confidential tickets; with access levels, you can configure that only a few select agents will have access to these tickets.

For an additional way to categorize tickets, have a look at *Tags*.

Hint: Zammad users are global to the whole instance. Restriction to specific groups is not possible.

2.1 Group Settings

Click on a group to edit it, or click on *New group* to create a new group.



No matter if you're going to edit or create a new group, each group comes with the following settings you can adjust as needed.

Name This is the name your agents (and customers when using *Web*) will see within the Zammad-UI.

Assignment timeout The time in minutes after which the ticket's ownership will revert back to *unassigned* after the assigned agent hasn't worked on the ticket.

> **Note:** This timeout does not take any working hours in account.

Follow-up possible This option allows you to decide how Zammad should react if a customer replies to a closed ticket (no matter if by e.g. email or UI).

yes The ticket will be reopened. This is the default value.

do not reopen Ticket but create new Ticket The ticket will remain closed and Zammad will create a new ticket instead. The new ticket contains the customers reply only.

do not reopen ticket after certain time but create new ticket The ticket can be reopened unless the specified number of days *after last ticket closure* has been exceeded. If the time limit is exceeded, Zammad will create a new ticket instead of reopening.

Choosing this option will provide the option *Reopening time in days* which requires you to provide the number of days you want Zammad to wait until it creates new tickets.

Assign follow-ups This setting allows you to decide if, upon a reopen of a ticket, the last assigned owner should stay assigned or if Zammad should reset the owner to nobody.

yes The ticket will remain to the last agent who owned it. This is the default value

no The owner assignment of the ticket will be removed.

Email Select which sender's email address Zammad will use outbound for replying on a ticket in this group.

> **Note: You don't have an email address configured yet?**
>
> Please configure an email based channel before here
>
> - *Channels → Email*
> - *Channels → Microsoft 365*
> - *Channels → Google*
>
> and come back afterwards.

Signature Choose which *signature* to use when replying to tickets in this group. Leaving this option unset will send emails without any signature.

Shared Drafts Shared drafts allows your agents to share ticket drafts (for new and existing tickets) with their colleagues. By default this setting is enabled, disable it if you don't want your agents to use this function.

Learn more about shared drafts in our user documentation.

Note An internal note about the group that is only visible to people who can access the group management area.

Active Don't need the group any more? If you can't or don't want to recycle (rename) the group, you can also set it to inactive. Agents and customers will no longer be able to see the group and thus can't add, update or read it's tickets.

> **Note:** As of now groups cannot be removed.

> **Warning:** Please keep in mind that you still can route tickets into these groups. This is potentially dangerous, make sure the group is no longer part of email filters or a destination group.

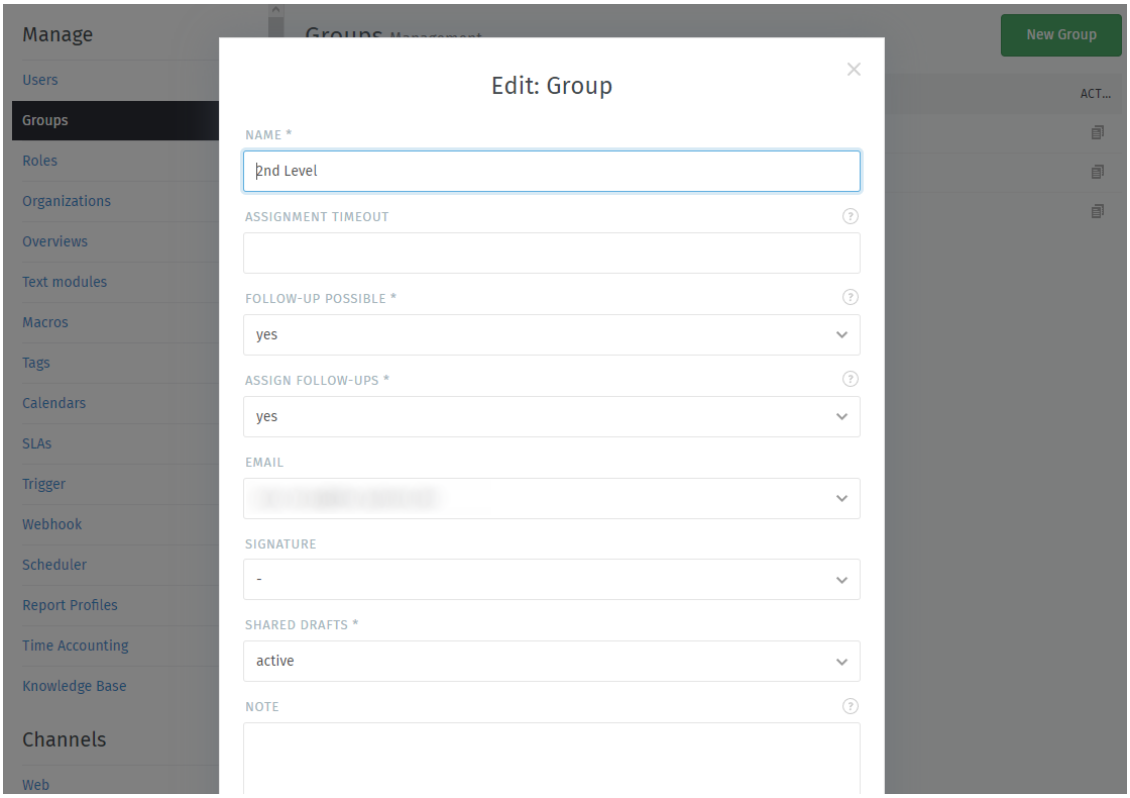


Fig. 1: A sample configuration of a group.

2.2 Group Access Levels

When assigning an agent to a *group*, Zammad gives you fine-grained control over what actions that agent can perform within it:

GROUP	READ	CREATE	CHANGE	OVERVIEW	FULL
HR	☐	☐	☐	☐	☑
Management	☐	☐	☐	☐	☐
Marketing	☐	☐	☐	☐	☐
Sales	☐	☐	☐	☐	☐

Fig. 2: Use the group access table to grant per-group privileges.

Within each group, the different access levels allow an agent to…

READ …view tickets

CREATE …create new tickets

CHANGE …modify existing tickets

OVERVIEW …see ticket overviews (but not ticket details)

FULL …all of the above **and be assigned / receive notifications for tickets**

Note: **Full group access also enables notifications for that group's tickets.**

2.2.1 Setting Access Levels

There are two ways to define an agent's per-group access levels:

1. Directly, in the *Edit User dialog*

Fig. 3: Simply set your access levels right on the target user.

2. Implicitly, by *editing a user's roles*

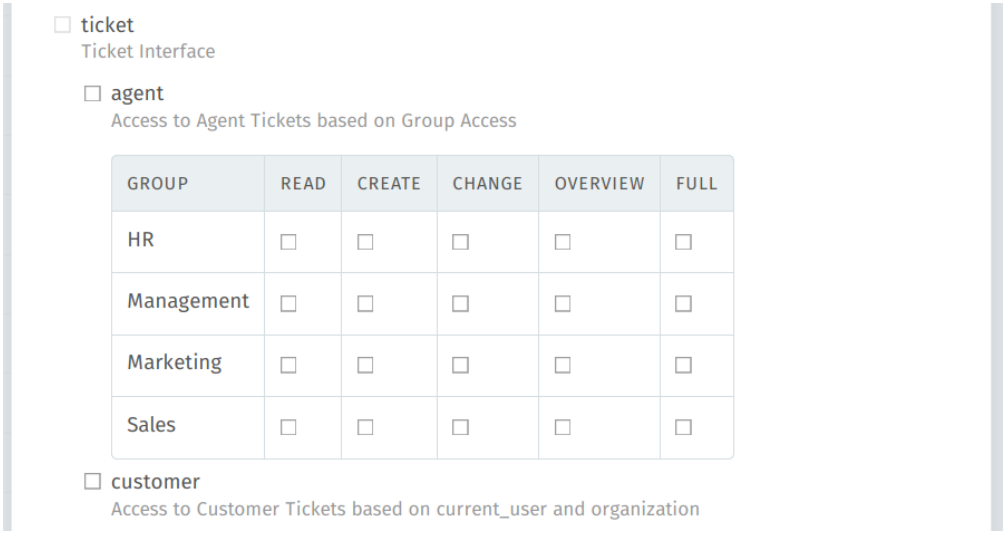


Fig. 4: First, set your access levels on a role…

Fig. 5: …then, add that role to the target user.

Note: **We recommend choosing one or the other**; things can get confusing if you use both at the same time.

So which one is right for you? *Whichever one is less work.* If you're trying to assign multiple agents to the same group with the same access levels, create a role for them to share—that's what roles are for!

2.2.2 Examples

“The Standard Issue”

GROUP	READ	CREATE	CHANGE	OVERVIEW	FULL
Incoming	☐	☐	☐	☐	☑

When a system only has one group, this is the default access level assigned to all agents. Unless you have special needs in mind, this is the way to go.

“The Supervisor”

GROUP	READ	CREATE	CHANGE	OVERVIEW	FULL
Incoming	☑	☑	☑	☑	☐

Agents with all permissions except for “full” **cannot be assigned tickets**. Otherwise, their privileges are identical to agents with “full” access. Great for letting other people do the real work.

“The Meddler”

GROUP	READ	CREATE	CHANGE	OVERVIEW	FULL
Incoming	☑	☐	☑	☑	☐

Agents with “read”, “change”, and “overview” access can do everything except create tickets or be assigned to them. Great for getting involved in other people’s business.

“The Intern”

Agents with only “create” access can do just that, and nothing else—once they hit **Save**, they’ll never see that ticket again. Great for taking phone calls for someone more important than you.

Hint: If the *Group* field does not appear in the ticket view, ensure that:

- you have created more than one group
- the current user has “change” permissions to more than one group

This is necessary because Zammad automatically hides selection fields with only one option.

GROUP	READ	CREATE	CHANGE	OVERVIEW	FULL
Incoming	☐	☑	☐	☐	☐

CHAPTER

THREE

ROLES

If you're already using Zammad, you'll know that users can be **admins**, **agents**, or **customers**. These are Zammad's built-in roles, and they're the tip of the iceberg of its powerful, flexible, and fine-grained permission system.

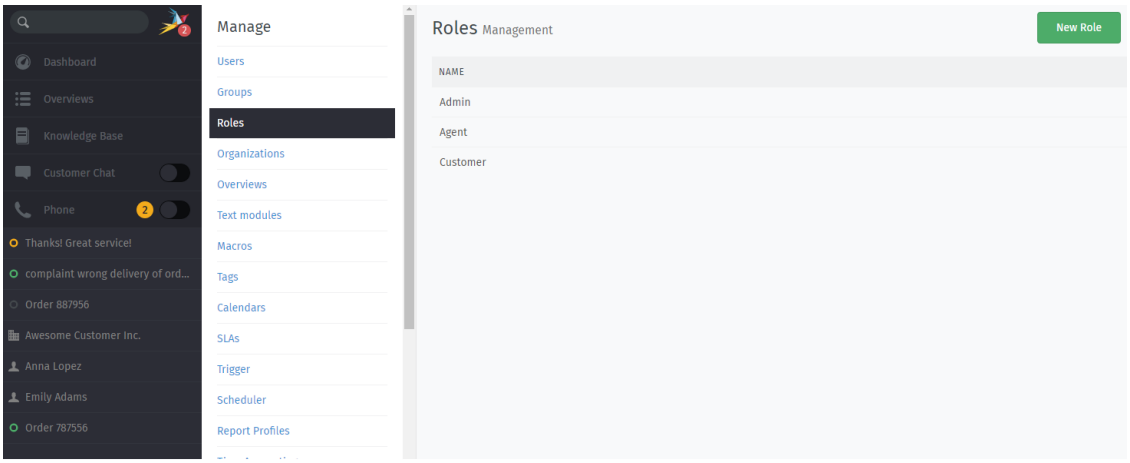


Fig. 1: Assign user privileges in the Admin Panel, under **Manage > Roles**.

Note: Users can have both "agent" and "customer" roles at the same time!

Why would you want this? Agents get *overviews* of all the tickets they're *assigned to* (among other things), while customers get an overview of all the tickets they've *opened*. But some teams use Zammad for both internal and public communication, so their agents need both.

Having both roles also changes what you see in the ticket view, depending on whether you're the "customer" or not.

Tip: LDAP/Active Directory users:

Syncing your LDAP "groups" to Zammad roles can make access management *way* easier. To learn more, see *LDAP / Active Directory*.

3.1 What Is a Role?

tl;dr Some users can do things others can't (like close a ticket). Users have roles, roles have permissions, and permissions are what make those actions possible.

So what exactly are permissions, then?

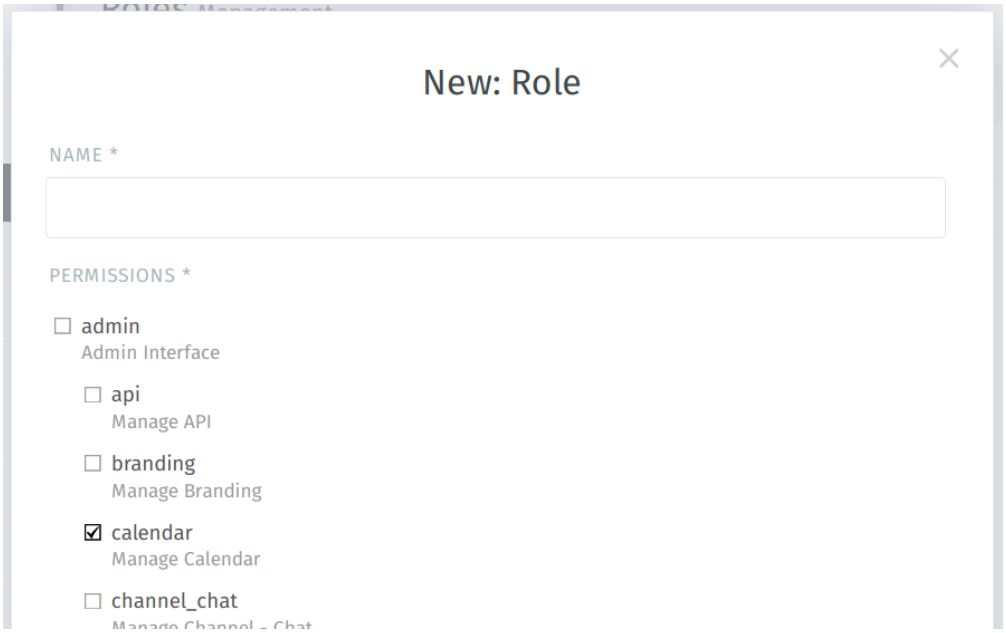


Fig. 2: The `admin.calendar` permission gives you access to the **Manage > Calendars** admin panel.

Simply put, permissions are names for all the different things users might want to do throughout the system, such as:

`chat.agent` respond to live chat messages

`ticket.agent` update tickets

`admin.user` access the **Manage > Users** admin panel

`knowledge_base.editor` create/edit knowledge base articles

Zammad has dozens of these permissions, which is a lot to keep track of. So instead of saying "This user has permissions A, B, and C", Zammad says "The *agent role* has permissions A, B, and C, and this user is an agent."

This makes creating user accounts for new agents a whole lot simpler, and it also makes it easier to invent a new permission D and say "All existing agents can do *that* now, too."

In short, roles are just collections of permissions that you can give to a user. The built-in admin, agent, and customer roles are enough for many teams, but Zammad gives you the freedom to custom-build your own.

And to do that, you'll need to know what each permission does.

3.2 Reference Guide: Permissions

3.2.1 Admin Permissions

Note: **Permissions are namespaced**, which is sort of like having files inside of folders.

The permissions listed on this page all belong to the `admin` namespace. You can select them individually, or you can just select `admin` to enable the whole bunch.

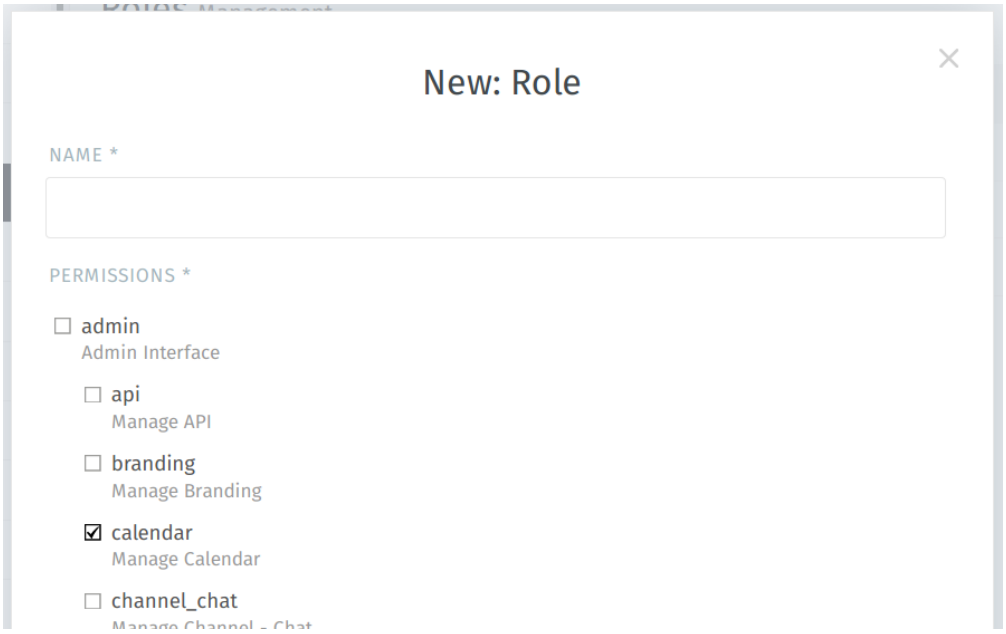


Fig. 3: Admin permissions are shown at the top of the New Role dialog…

x

`admin.api` System > API

`admin.branding` *Settings* > *Branding*

`admin.calendar` *Manage* > *Calendars* (required for *SLAs*)

`admin.channel_chat` *Channels* > *Chat*

> **Hint:** **Trying to grant access to send messages in live chats?**
>
> Use `chat.agent` instead.

`admin.channel_email` *Channels* > *Email*

`admin.channel_facebook` *Channels* > *Facebook*

> **Hint:** **Trying to grant access to view/update tickets from Facebook?**
>
> That's in *Group Access Levels*.

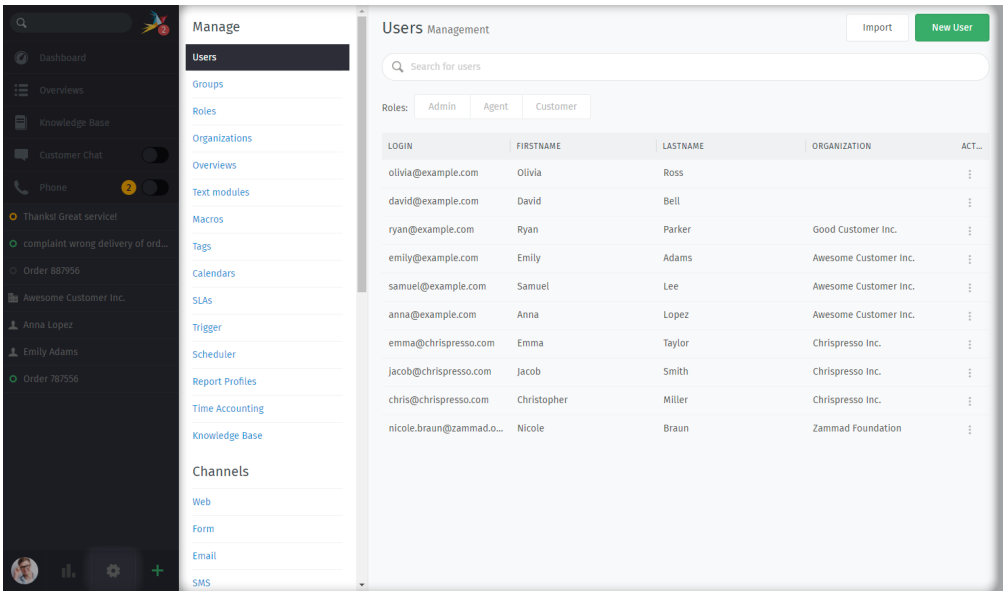


Fig. 4: ...and give users access to the pages of the Admin Panel.

admin.channel_formular *Channels > Form*

admin.channel_google *Channels > Google*

admin.channel_microsoft365 *Channels > Microsoft 365*

admin.channel_sms Channels > SMS

admin.channel_telegram *Channels > Telegram*

> **Hint: Trying to grant access to view/update tickets from Telegram?**
>
> That's in *Group Access Levels*.

admin.channel_twitter *Channels > Twitter*

> **Hint: Trying to grant access to view/update tickets from Twitter?**
>
> That's in *Group Access Levels*.

admin.channel_web *Channels > Web*

admin.core_workflows *System > Core Workflows*

admin.data_privacy *System > Data Privacy*

> **Danger:** This permission allows users to permanently delete data on the system. Proceed with caution!

admin.group *Manage > Groups*

admin.integration *System > Integrations*

admin.knowledge_base *Manage > Knowledge Base*

> **Hint: Trying to grant access to read/edit knowledge base articles?**
>
> Use `knowledge_base.reader` and `knowledge_base.editor` instead, and double-check the answer's visibility.

admin.macro *Manage > Macros*

> **Note:** In some cases, macros may also require `admin.tag`.

admin.maintenance *System > Maintenance*

admin.monitoring *System > Monitoring*

admin.object *System > Objects*

admin.organization *Manage > Organizations*

> **Note:** Agents can access existing organizations from the search bar, even without this permission. They can even edit an organization's name, domain, and notes!

admin.overview *Manage > Overviews*

admin.package *System > Packages*

admin.report_profile *Manage > Report Profiles*

> **Hint: Trying to grant access to view reports?**
>
> Use `report` instead.

admin.role *Manage > Roles*.

admin.scheduler *Manage > Scheduler* for automation on tickets

admin.security *Settings > Security* settings of Zammad This also covers third party authentications.

admin.session *System > Sessions*

admin.setting_system *Settings > System* of Zammad

admin.sla *Manage > SLAs*

admin.tag *Manage > Tags*

admin.template *Manage > Templates*

admin.text_module *Manage > Text Modules*

admin.ticket *Settings > Tickets* (does not grant access to *Composer Settings*)

admin.time_accounting *Manage > Time Accounting*

> **Hint:** This permission may be useful for accounting personnel if they need to be able to export timekeeping records.

admin.translation *System > Translations* (also enables inline translation)

admin.trigger *Manage > Triggers*

admin.user *Manage > Users*

Note: I thought agents could already manage user accounts?

Agents can create and edit **customers**, but they can't:

- modify anyone's permissions (roles or groups)
- modify anyone's passwords
- edit other agent's accounts

Danger: This permission allows users to hijack other user sessions.

To learn more, see *Taking over a user's session*.

3.2.2 Agent Permissions

Note: The permissions listed on this page grant access to features that have to be enabled or configured system-wide in the Admin Panel first.

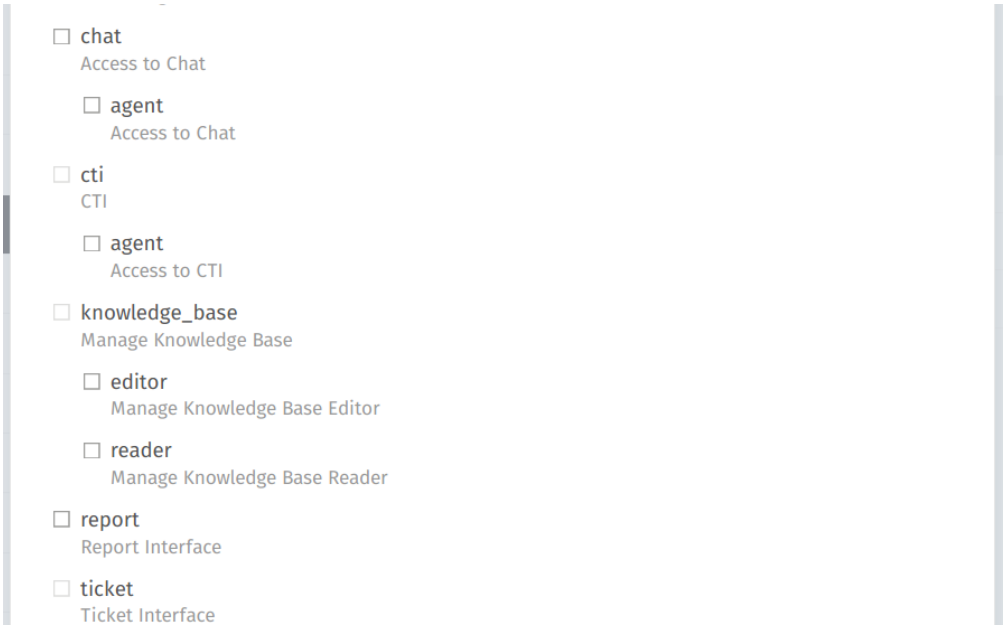


Fig. 5: Agent permissions are shown in the middle of the New Role dialog…

x

chat.agent Customer Chat

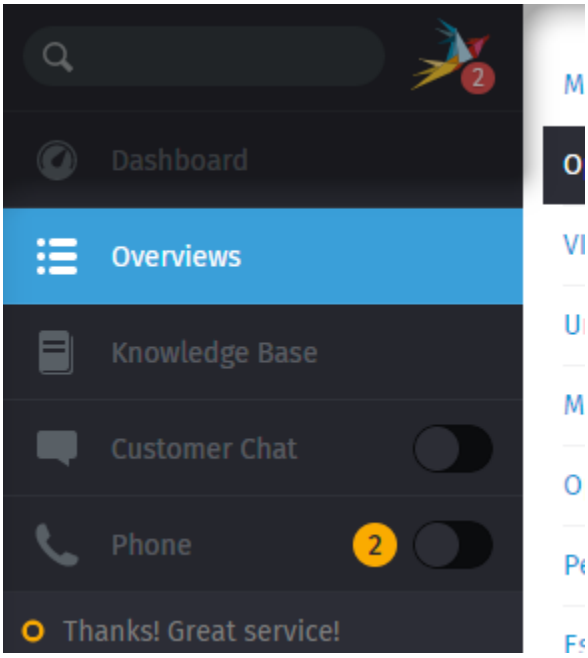


Fig. 6: . . . and give users access to new sidebar tabs for communicating with customers.

Hint: Requires configuration of *Chat Channel*

`cti.agent` Provides access to Caller Log

Hint: Requires configuration of either integrations

- *Generic CTI*
- *placetel*
- *sipgate*

---

knowledge_base Knowledge Base

knowledge_base.editor create/edit privileges

Hint: Editor permissions always include reader permissions.

---

knowledge_base.reader read privileges for internal content

Hint: Public articles are always visible.

Tip: Zammad supports granular permissions on knowledge base categories.

This function allows *agents with editor* permissions to restrict specific internal categories & answers to chosen roles.

In order to allow your agents to set granular role permissions, the roles in question require at least reader permission for the knowledge base.

Danger: Keep in mind that this may be dangerous, as reader permission provides access to internal answers!

`report` *Reporting*

Warning: Never grant this permission to your customers.

Giving customers access to reporting constitutes a serious data breach, as it includes **all ticket and user information across the entire system**!

Note: This permission is the exception to the rule on this page:

1. the feature it enables is not for communicating with customers;
2. the button appears at the *bottom* of the sidebar; and
3. it is typically reserved for admins and supervisors.

`ticket.agent` (Agent) Overviews

Note: What's this big table doing here in the middle of my permissions?

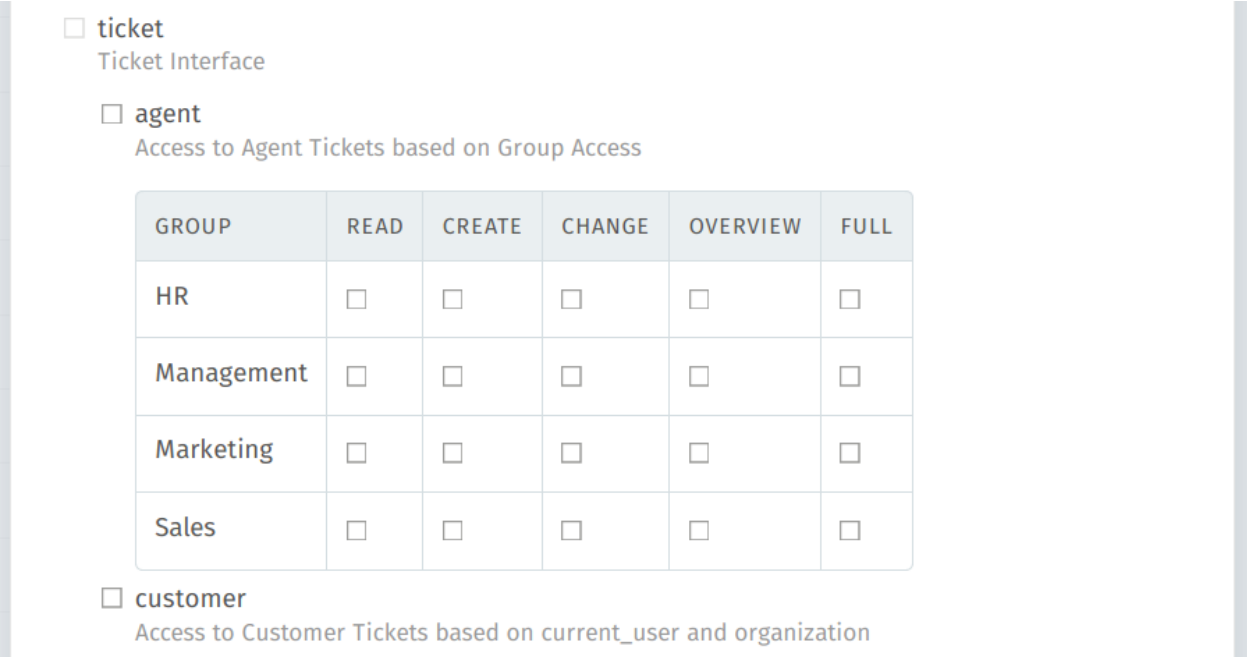


Fig. 7: The group access table is shown when there is **more than one active group** in the system.

Okay, so remember when we said that *"roles are just collections of permissions"*? That wasn't

entirely true—they can also be collections of **group access levels**.

To learn more, see *Group Access Levels*.

Hint: Point of technicality

You can assign both agent and customer roles to the same user — but you **can't** assign both `ticket.agent` and `ticket.customer` *permissions* to the same *role!*

To make it work, you need two separate roles: one with `ticket.agent` and the other with `ticket.customer`.

3.2.3 User Preferences Permissions

Note: Permissions are namespaced, which is sort of like having files inside of folders.

The permissions listed on this page all belong to the `user_preferences` namespace. You can select them individually, or you can just select `user_preferences` to enable the whole bunch.

Fig. 8: User preferences permissions are shown at the bottom of the New Role dialog…

x

`user_preferences.access_token` Generate API tokens to control Zammad via the REST API

Note: Security Tip

Generated tokens will never have more permissions than the user that generated them.

`user_preferences.avatar` Override the default Gravatar with a custom avatar

`user_preferences.calendar` Configure the calendar feed

`user_preferences.device` Manage device login sessions

Note: Security Tip

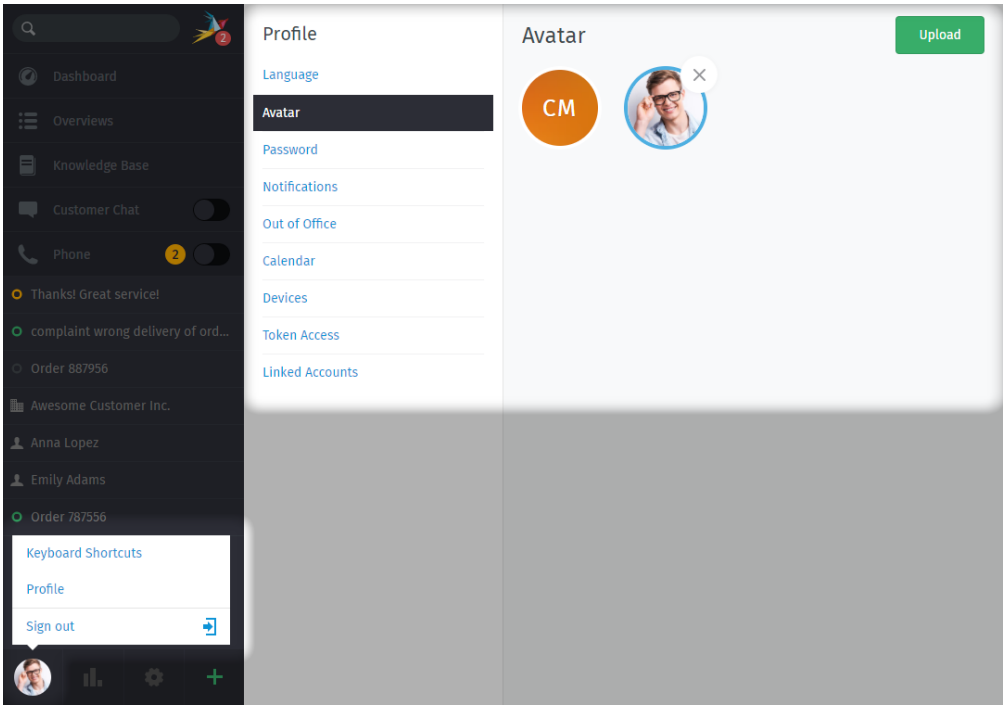


Fig. 9: . . . and give users access to the pages of their User Profile.

Revoking this permission disables "Login detected from a new location" notification emails.

To learn more, see *System Notifications*.

`user_preferences.language` Configure the UI locale/language

`user_preferences.linked_accounts` Manually link accounts after signing in with *third-party authentication*

Note: If *automatic account linking* fails, this is the only way your users can utilize third-party logins.

`user_preferences.notifications` Configure ticket notification settings

Note: Agents only receive ticket notifications for *groups they have "full" access to*.

Customers can't receive ticket notifications at all.

`user_preferences.out_of_office` Designate a substitute for out-of-office hours

Note: Security Tip

Designating a substitute does **not** grant that person the permissions / *group access levels* of the agent they're replacing.

`user_preferences.overview_sorting` Allow your users to define their own overview order.

Note: The order your user chooses here cannot be overwritten by admins. Renaming or resorting overviews has no effect on custom orders.

Hint: This is an optional permission for customers and thus disabled by default.

`user_preferences.password` Change account password

Warning: Third-party authentication / LDAP users:

Be sure to **revoke this permission** for all your users. When using a third-party identity server (like LDAP), the whole point is to let *it* take care of authentication so that passwords never have to live in Zammad's database.

Broadly speaking, there are four types of permissions:

Admin for access to each page of the Admin Panel

Agent for access to customer communications

Customer Without the `ticket.customer` permission, customers can't see the **My Ticket** overview—but they can still log in and open new tickets!

User Preferences for access to your own user profile

Note: Permissions are namespaced, which is sort of like having files inside of folders.

These permissions:

- `admin.api`
- `admin.branding`
- `admin.calendar`
- `admin.channel_chat`
- `admin.channel_email`
- …and 30+ more

all belong to the `admin` namespace. You can select them individually, or you can just select `admin` to enable the whole bunch.

3.3 Role Details

Default at signup Every new user must be assigned at least one role upon creation. This attribute decides which role to give new users by default (which usually happens when creating a new ticket for a new customer).

Warning: Default roles should never provide admin/agent permissions.

Fig. 10: The default role is identified in the overview of the **Manage > Roles** admin panel.

CHAPTER

FOUR

ORGANIZATIONS

Depending on your organization's IT capabilities, organizations can be managed individually or in bulk.

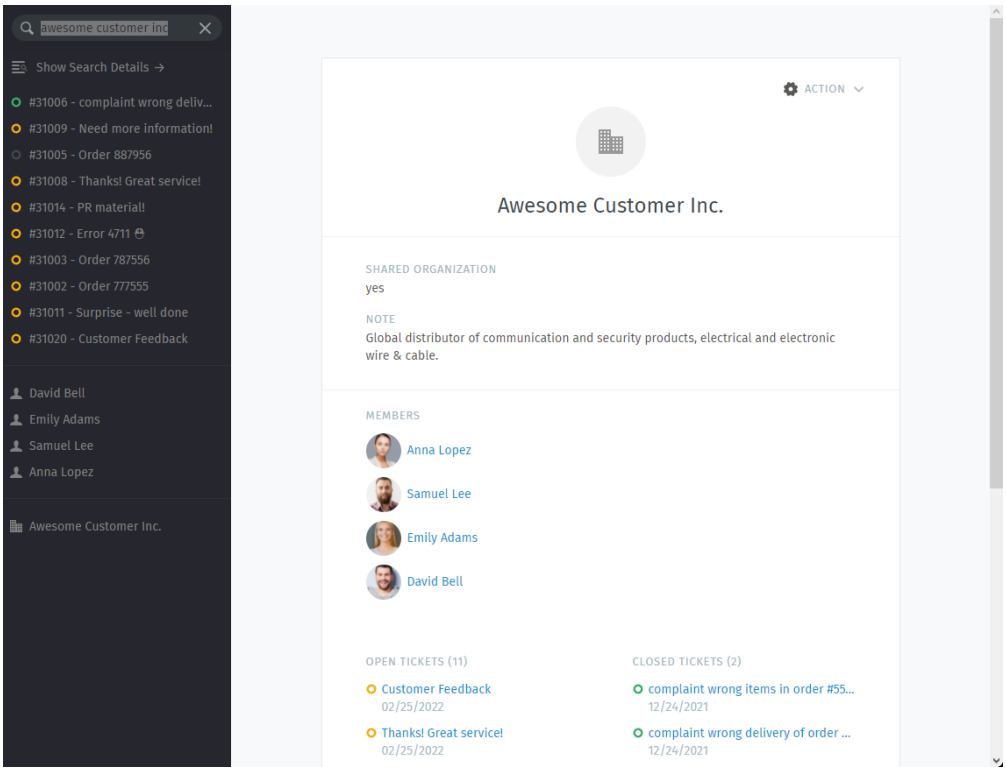


Fig. 1: The simplest way to manage organizations is directly in the Admin Panel.

Learn more about managing organizations…

4.1 Managing organizations via the admin panel

The "Organizations" panel provides tools to manually manage organization entries.

Creating and editing users

Fig. 2: Click the **New Organization** button to open the New Organization dialog, or click on an existing organization to edit.

Hint: See *Reference Guide: Organization Details* for help with the New/Edit Organization dialog.

Deleting organizations Organizations currently can only be removed via data privacy by deleting the last organization member and then choose `yes` for *Delete organization?*.

Fig. 3: Use the **Actions** menu to open the **Delete User** dialog.

> **Warning: Deleting a customer destroys all their associated tickets!**
>
> To learn more, see *Data Privacy*.

Hint: Technically organization removal is possible via Zammad's API, however, this only works in very specific situations. You may want to stick to data privacy as of now.

4.2 Managing organizations via CSV Import

If your organization has information about customers (or personnel) already stored in a directory system that can export to CSV, you can batch-import them into Zammad in just three steps.

4.2.1 Step 1: Inspect the sample .csv

Use the link at the bottom of the CSV import dialog (**"Example CSV file for download"**) to see how Zammad expects you to format your organization data.

4.2.2 Step 2: Export your organization data to .csv

Prepare your .csv file according to the format found in Step 1 above. Note that:

- the `id` attribute (column) should be left blank or removed entirely;
- the `name` attribute is **required**; and
- any other columns may be safely omitted as long as each row has the same number of fields (commas).

For instance:

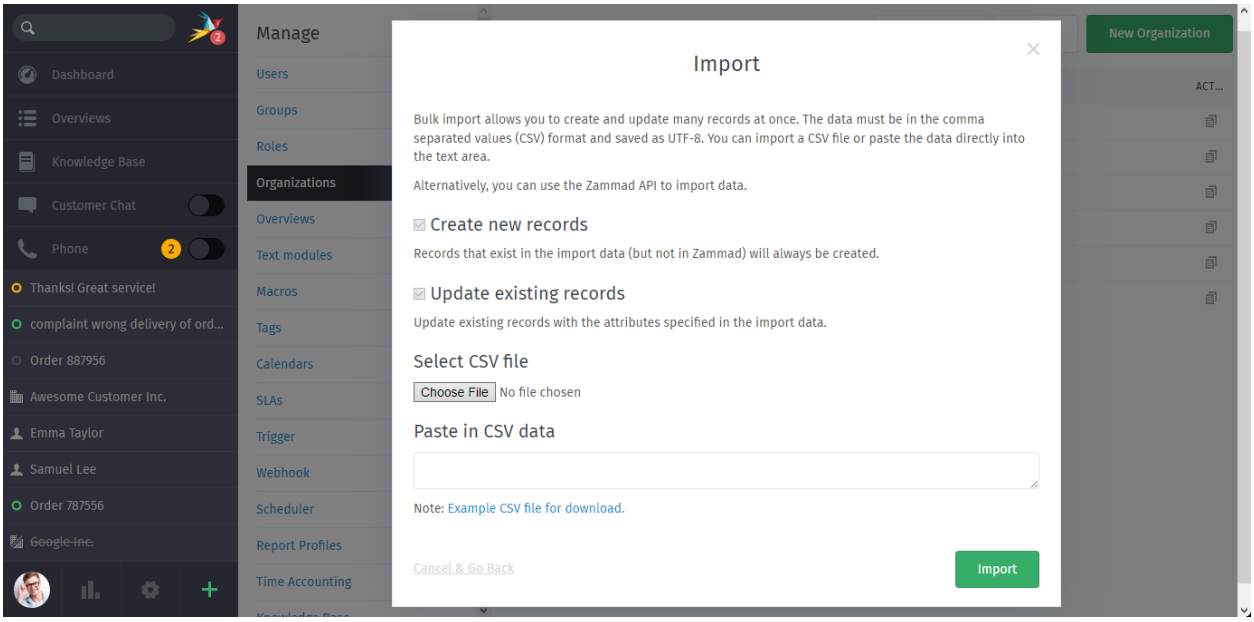


Fig. 4: Use the **Import** button to open the CSV import dialog.

```
name,shared,domain,domain_assignment,active,members
Chrispresso Inc.,true,"",false,true,emma@chrispresso.com
"","","","","",jacob@chrispresso.com
"","","","","",chris@chrispresso.com
Awesome Customer Inc.,true,"",false,true,emily@example.com
"","","","","",samuel@example.com
"","","","","",anna@example.com
Zammad Foundation,true,"",false,true,nicole.braun@zammad.org
```

Hint: Several organization members can be added: Ensure to provide the users email address as shown above with empty values.

4.2.3 Step 3: Import your .csv to Zammad

Once your raw organization data has been submitted, Zammad will perform a test run to compute the number of affected records:

Note: How does it know when to create new records and when to update?

Records are updated when the imported data contains a `name` that matches an existing organization entry.

Click **Yes, start real import** to proceed. (If you're importing a lot of records, be patient—it may take a minute.)

That's it!

Note: Technical Limitations

- Organizations currently cannot be removed. The only exception is Zammad's *Data Privacy* function.

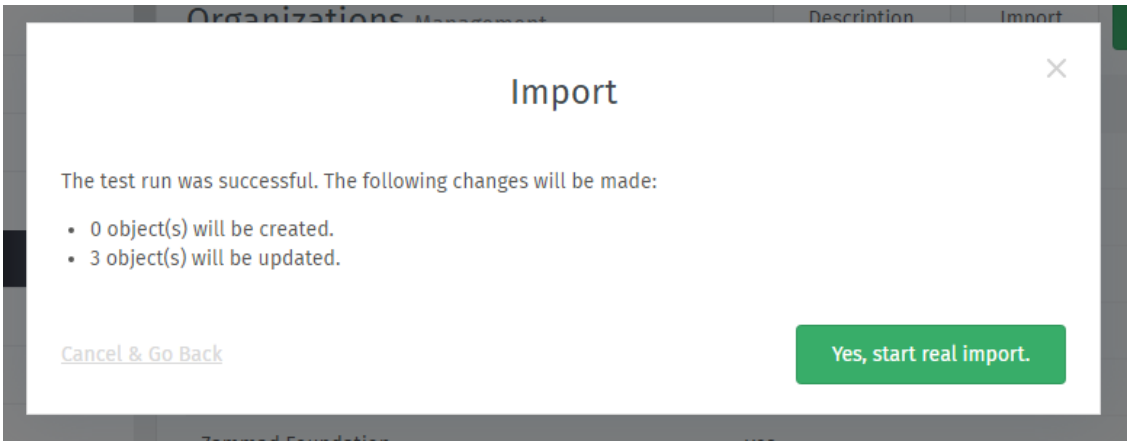


Fig. 5: CSV import always begins with a preview / test run.

- Unlike users, agents cannot just create new organizations. Check the *permission reference* to learn more.
- Because of how organization references work with users, external syncs like LDAP or Exchange *do not* support organization mapping.

> **Hint:** This is relevant to you? Consider domain based assignments.

> **Warning:** **BIG organizations can cause performance issues**
>
> Organizations with many members can cause a fairly high system load in some situations. This especially affects organizations whose members run many updates, for example ticket creations or frequent communication. A lot of linked data syncs may cause an overhead.
>
> Proceed with caution.

4.3 Reference Guide: Organization Details

Most of the attributes you can set on organizations are self-explanatory. The ones that aren't are described below.

> **Note:** **Admins aren't the only ones who can change these settings.**
>
> In most cases, agents can, too (using the ticket pane or organization detail page).

Shared Organization If you set this option to `yes`, all organization members will be able to **view** and **update** tickets of their organizational members in addition to their own.

Setting this option to yes also provides access to overviews being available to shared organizations only. Learn more on *Overviews*.

The default value on creation dialogues is `yes`.

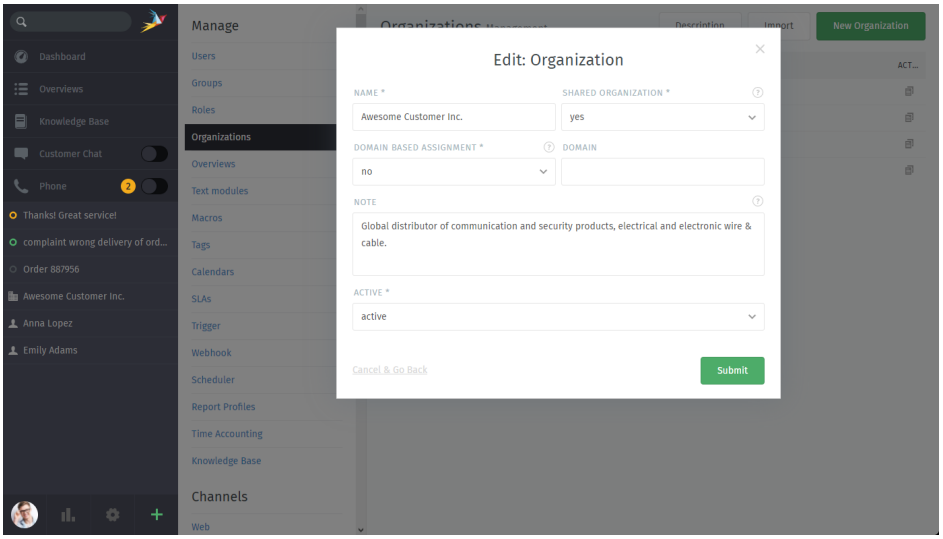


Fig. 6: User details can be set in the **New/Edit Organization** dialog.

Danger: This can cause serious issues if you have e.g.human resources working in the same Zammad instance. Shared organizations usually are relevant for Support companies with fairly big customers and support contingents.

Hint: Sharing organizations don't just affect customers, however, if you want to provide ticket access to agents, please see the *Reference Guide: Permissions*.

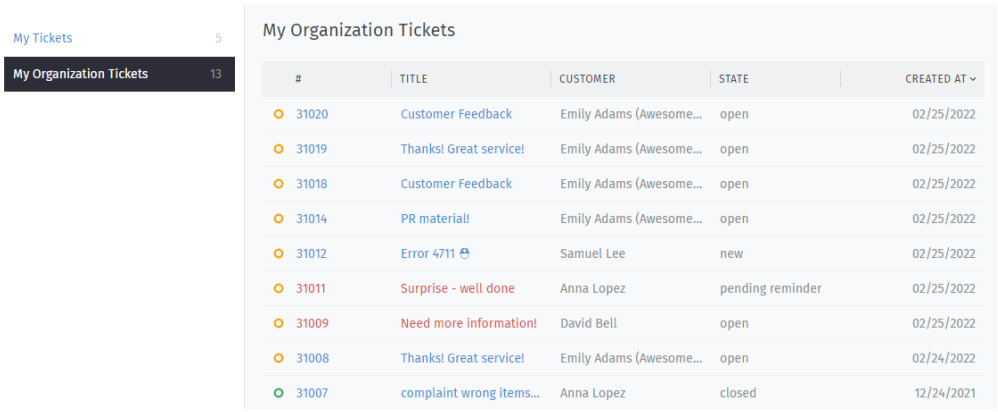


Fig. 7: Members of shared organization have access to organization based overviews

Domain based assignment Activating domain based assignment will cause Zammad to automatically add newly created users to said organization. This can greatly reduce your maintenance effort and is seen as workaround for not being able to map organizations via LDAP.

The default value on creation dialogues is no

Note: Domain based assignment only works for *newly created* users and has no effect on existing users.

Domain Add the email domain of the organization with this option. It's being used on user creation to determine the assignment. This option belongs to domain based assignment and is required if set to `yes`.

Note: At the time Zammad allows *one* domain per organization. You may also want to ensure to not use free mailer domains like `gmail.com` for these assignments.

Note Notes are visible to all staff members, **including agents**.

Hint: Are you using the Note field to keep track of your own "custom" organization attributes?

Wish you could add your own fields to the New/Edit Organization dialog?

You can! To learn more, see *Objects*.

Active Disabling this flag is a soft alternative to deleting an organization. So what's the difference?

1. There is no way to restore a deleted organization; inactive organizations can be reactivated at any time.
2. Inactive organizations still appear in search results:

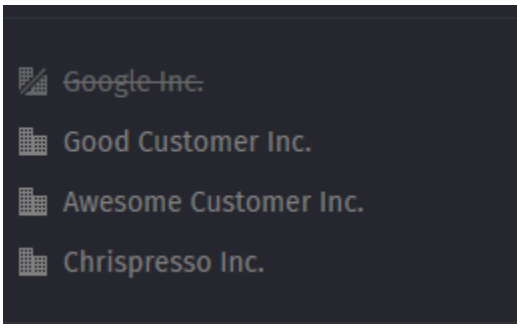


Fig. 8: A slashed-out icon indicates an inactive organization. In other cases, inactive organizations are greyed out.

CHAPTER

FIVE

OVERVIEWS

You can provide overviews to your agents and customers. They can be used as a kind of worklist of tasks that the agent is supposed to work off. You can also create individual reports for individual agents or agent groups.

In the Overview Management Area you can add new overviews, edit or delete them.

> **Warning:** Please note that Overviews can cause performance issues leading to no longer or less often refreshing overviews!
>
> Whenever possible, try to use the same overviews for as many agents and groups as possible to keep the number of overviews low. For best results, you might want to use between 15-20 overviews maximum. Also, any overview will only show a total of 2100 elements.

Note: Overviews will only show tickets to your users, that the user have rights on (group or role based).

The following attributes can be set when creating an overview:

Available for the following roles / Restrict to only the following users

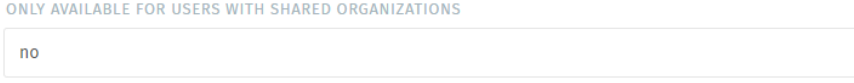
> **Hint:** Roles are assigned to users, per default there are agents, admins and customers. Further information about *Roles*.

> **Note:** The setting "available for the following roles" is mandatory.

Define roles that are supposed to see and use the overview in question.

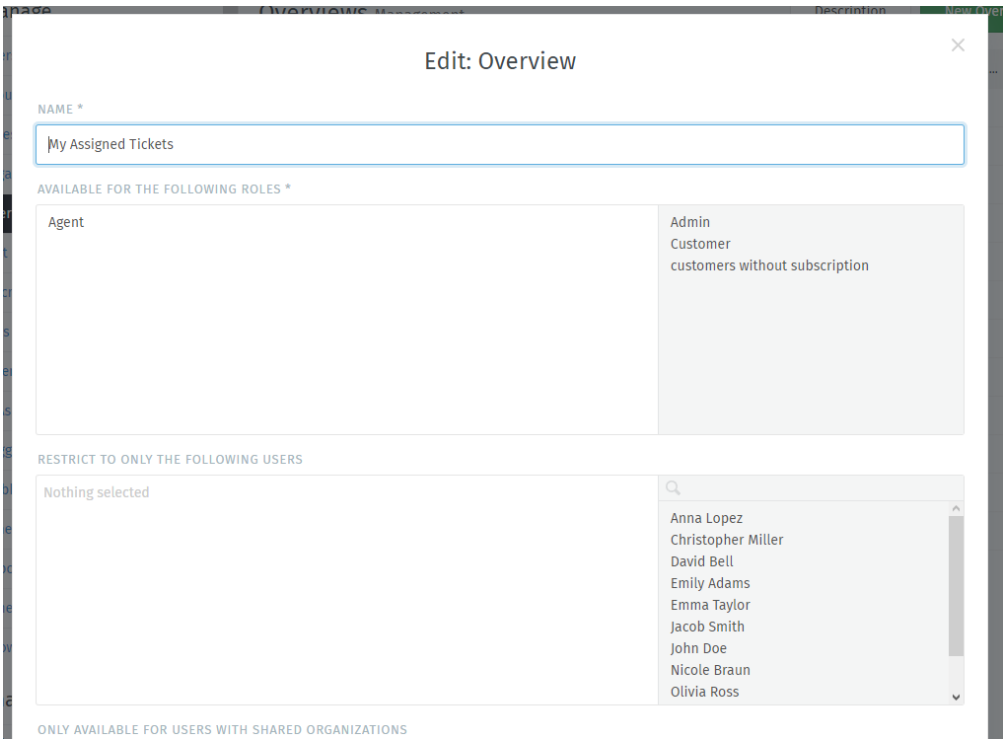
If your overview is rather specific for a sub group of users of your role, use the *"Restrict to only the following users"* option to further restrict the visibility of the overview to defined users.

> **Hint:** You will still have to provide a role!



Only available for users with shared organization

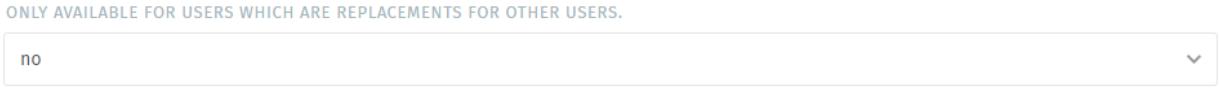
> **Hint:** Shared organization is a setting in the organization management. See *Organizations* for more information.



This is only important if the available role is a customer. When deciding whether yes or no is selected, it must be considered to what extent this makes sense - for example, if a customer sees only his own tickets, many views are usually not necessary…

Note: Users also refers to the customer role in this case.

Available for users which are replacements for other users.



This selection refers to the setting in the user preferences (profile-pic in the left corner –> profile –>) "out of office". If this option is checked, this selection is only displayed if someone has been entered as a substitution.

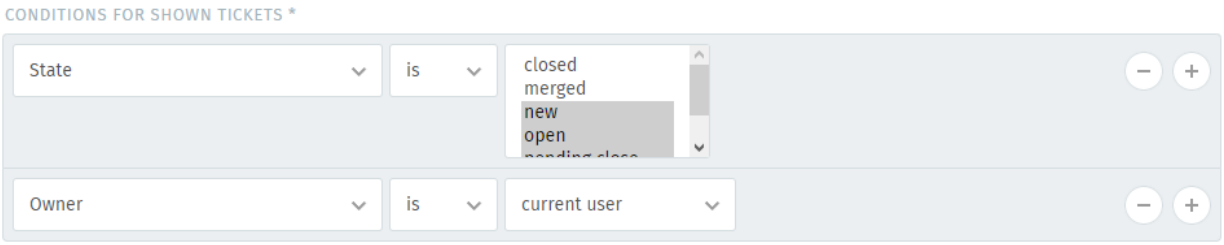
For example: Agent A is on vacation and Agent B will take care of his tickets. Then an overview can be set up, which only shows Agent B all new tickets from Agent A for this period of time, without having to search for them separately.

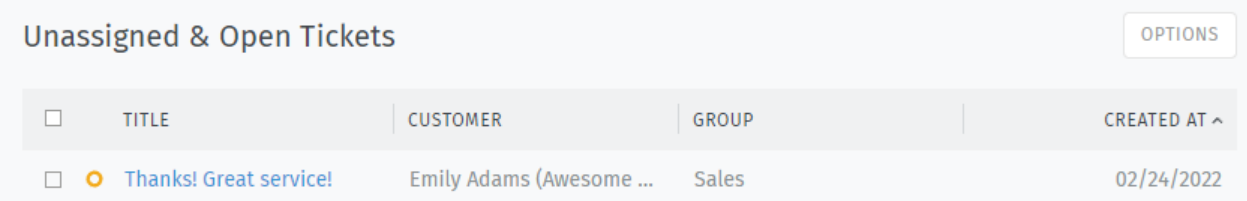
Note: Replacement users are part of our Out of Office function.

Conditions for shown tickets

What conditions should the listed tickets contain? (it is like a filter) You can add more than one condition. In the preview you have the possibility to double check if your entry of the conditions makes sense by directly displaying tickets that match your filtering.

Attributes





Which attributes shall be shown in the overview? (column headers)

With this setting you can select the headlines of your overview. Depending on which information is important in this selection, it can be displayed individually. For this example "Unassigned and open" the overview would look like this:

These settings can also be adjusted individually by admins at a later time (In the overview, top right: Options).

Note: Please note that overview column and sort settings are global settings which affect all users seeing those overviews.

Sorting, Grouping and Active

Sorting by In which order should the tickets be displayed? (Sorted by the attributes)

Sorting order The direction of the sorting.

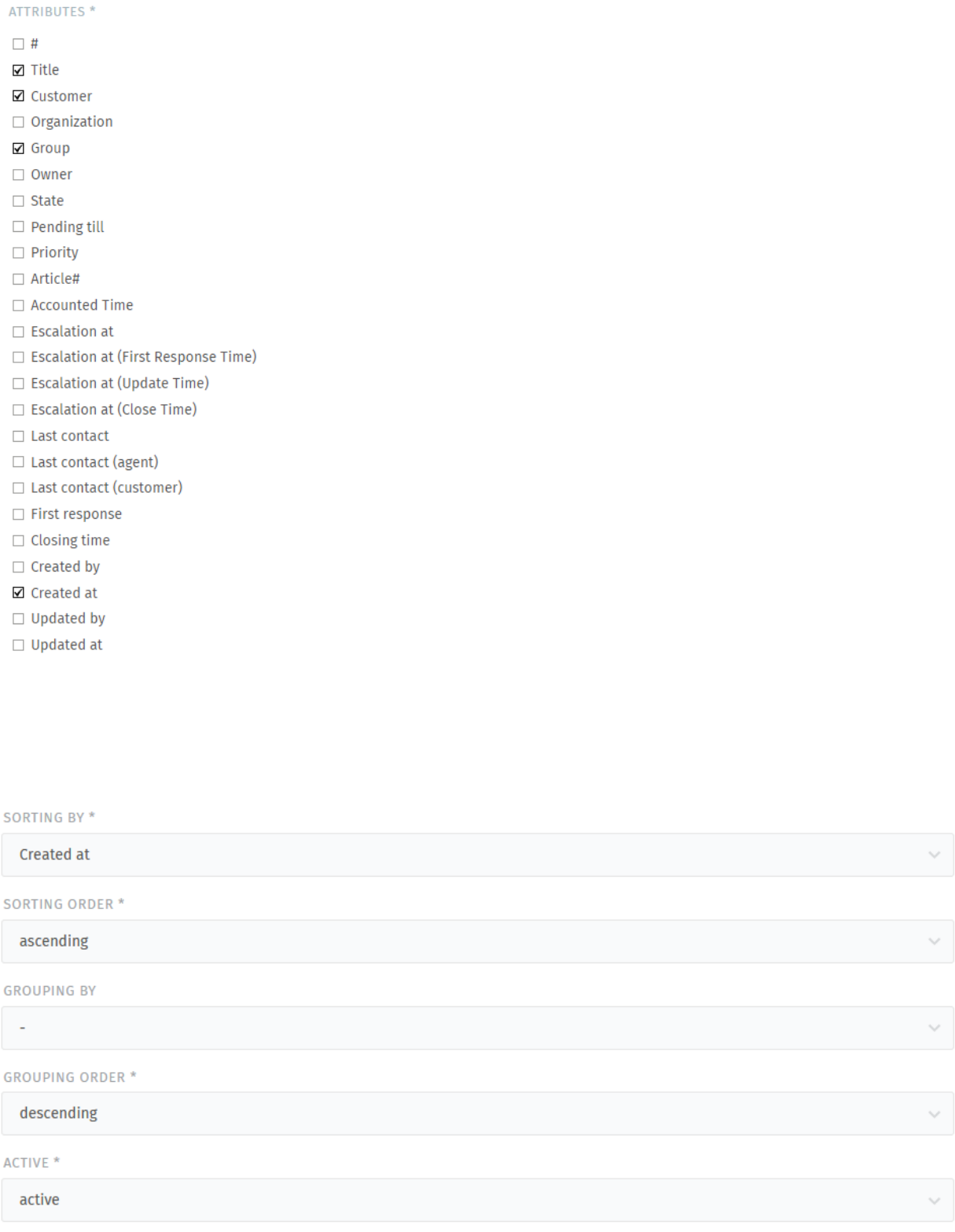
Grouping by Should the tickets be grouped by a specific attribute within the list?

Grouping order The direction of the grouping.

Active Should the overview be active or not? Rather than deleting an overview entirely, you can set it to inactive to make it unavailable to your users.

Note: Users can define their own overview order. Renaming or reordering overviews has no effect on custom orders!

You can learn more about this setting in the user documentation.


ATTRIBUTES *
#
Title
Customer
Organization
Group
Owner
State
Pending till
Priority
Article#
Accounted Time
Escalation at
Escalation at (First Response Time)
Escalation at (Update Time)
Escalation at (Close Time)
Last contact
Last contact (agent)
Last contact (customer)
First response
Closing time
Created by
Created at
Updated by
Updated at
SORTING BY *
Created at
SORTING ORDER *
ascending
GROUPING BY
-
GROUPING ORDER *
descending
ACTIVE *
active

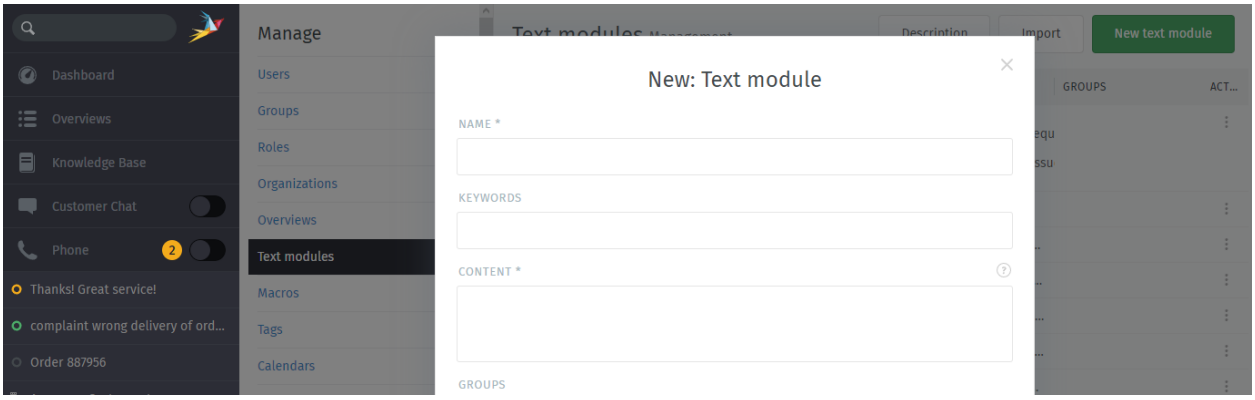
CHAPTER

SIX

TEXT MODULES

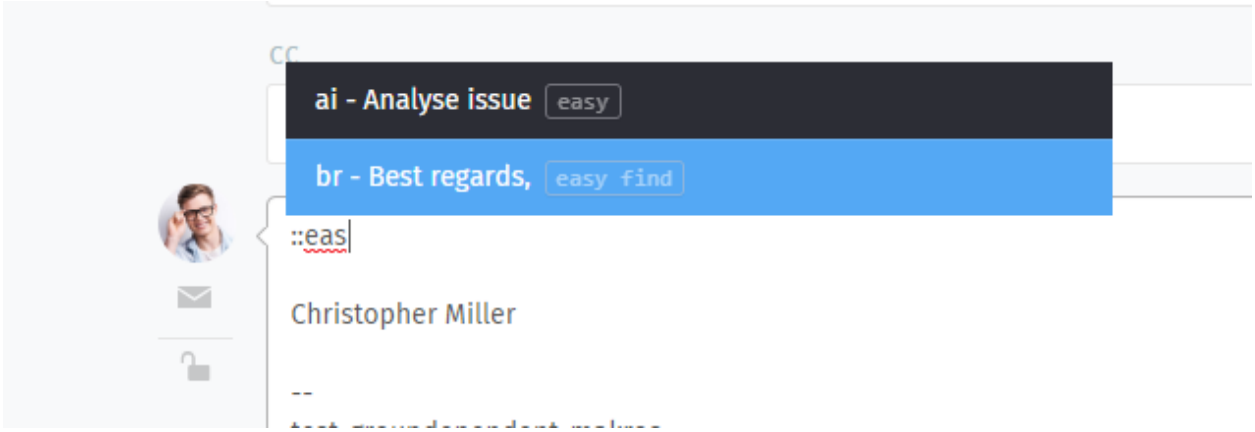
Note: Beside text modules Zammad also allows you to use Ticket Templates for ticket creation.

Text modules can be edited in the admin interface under Manage –> Text modules. Here you will find text snippets already created in the standard version, which can be extended as needed.



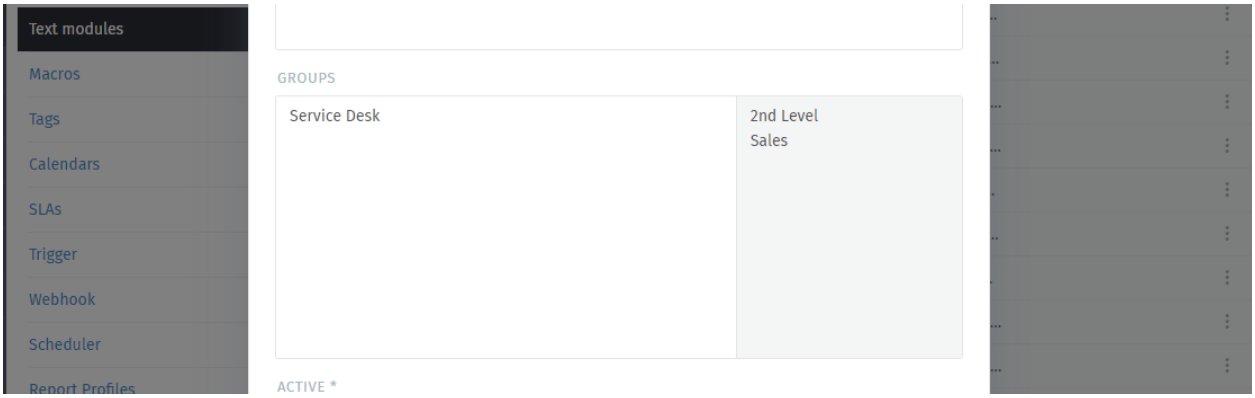
Here you can add new text modules, delete or edit them.

Creating keywords makes it easier to find the right text module.



If needed, you can restrict text modules to specific groups. With this, you can easilly keep text module lists short and dedicate specific texts to where they belong.

You can adjust the group memberships for text modules at any time. This allows you to have the text module available globally (no groups selected) or one or several specific groups.



To select placeholders from a list, just enter `::` in the text block. The list can be searched with the arrow keys after inputting keywords or shortcuts. All text modules can be used in articles as well as in the chat.

Note: You can find more information on how to use text modules on our user documentation.

Tip: If text modules are to be grouped, this can be done using shortcuts. Example country codes:

Text modules are created for the group Germany as follows:

- Ger_Textmodule1
- Ger_Textmodule2
- …

for Austrian-Snippets:

- Aut_Textmodule1
- Aut_Textmodule2

thus only the relevant text modules are displayed for each country.

The example text modules below use *Variables* to dynamically insert information like the customer's or agent's names.

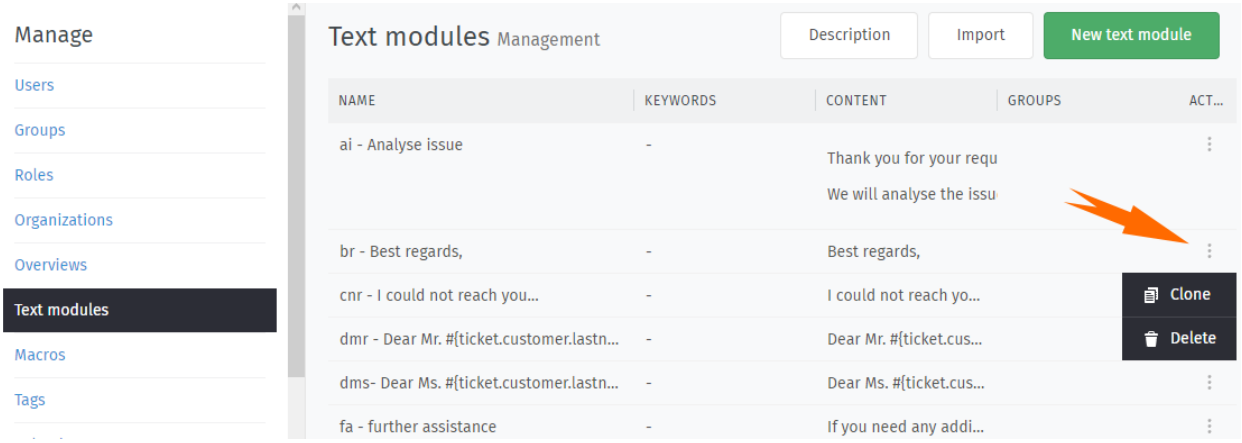
Examples of snippets are:

```
Hello Mrs. #{ticket.customer.lastname},

Hello Mr. #{ticket.customer.lastname},

Hello #{ticket.customer.firstname},

My Name is #{user.firstname},
```

Of course you can also use multi line snippets.

6.1 Delete or clone text modules

Often similar text modules have to be created or unnecessary ones deleted. For these cases you can click on the 3 points in the text module overview on the right side and select the corresponding action:

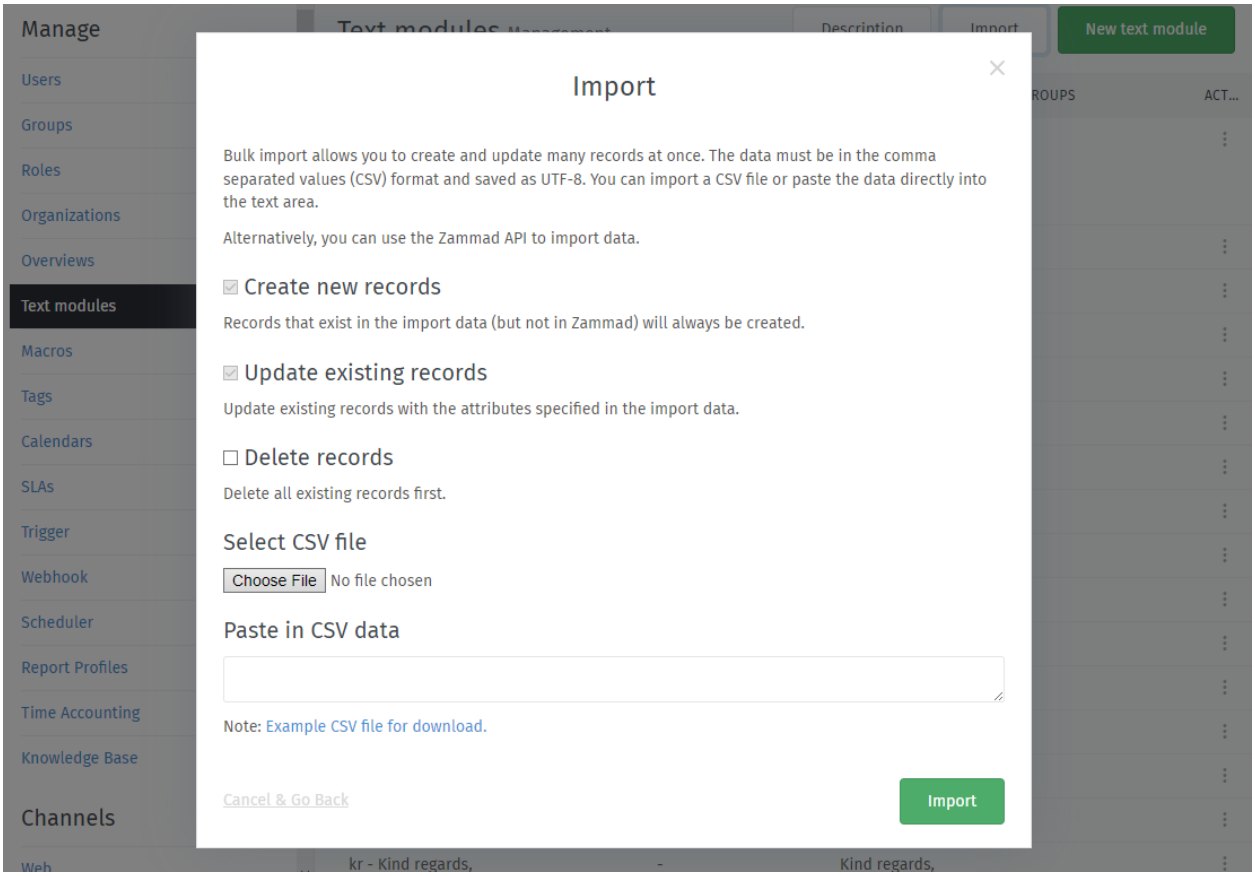


When cloning, text modules with all attributes are duplicated and can be edited later.

6.2 Import of text modules via CSV file

With the import action (since Zammad 2.5) you can download a sample CSV file and upload your own CSV file.

To reduce the error rate of unwanted mass changes, a test import is carried out first and a summary appears at the end. If you agree with the summary, the CSV import will be executed.


Manage
Users
Groups
Roles
Organizations
Overviews
Text modules
Macros
Tags
Calendars
SLAs
Trigger
Webhook
Scheduler
Report Profiles
Time Accounting
Knowledge Base
Channels
Description
Import
New text module
ROUPS
ACT...
Import
Bulk import allows you to create and update many records at once. The data must be in the comma separated values (CSV) format and saved as UTF-8. You can import a CSV file or paste the data directly into the text area.
Alternatively, you can use the Zammad API to import data.
Create new records
Records that exist in the import data (but not in Zammad) will always be created.
Update existing records
Update existing records with the attributes specified in the import data.
Delete records
Delete all existing records first.
Select CSV file
Choose File
No file chosen
Paste in CSV data
Note: Example CSV file for download.
Cancel & Go Back
Import
kr - Kind regards,
Kind regards,

CHAPTER

SEVEN

MACROS

Macros are **one-click shortcuts** for applying changes to a ticket.

If you find yourself making the same changes to lots of tickets (*e.g.,* close-and-tag-as-spam or reassign-to-another-group), you can store those changes in a macro for easy access:

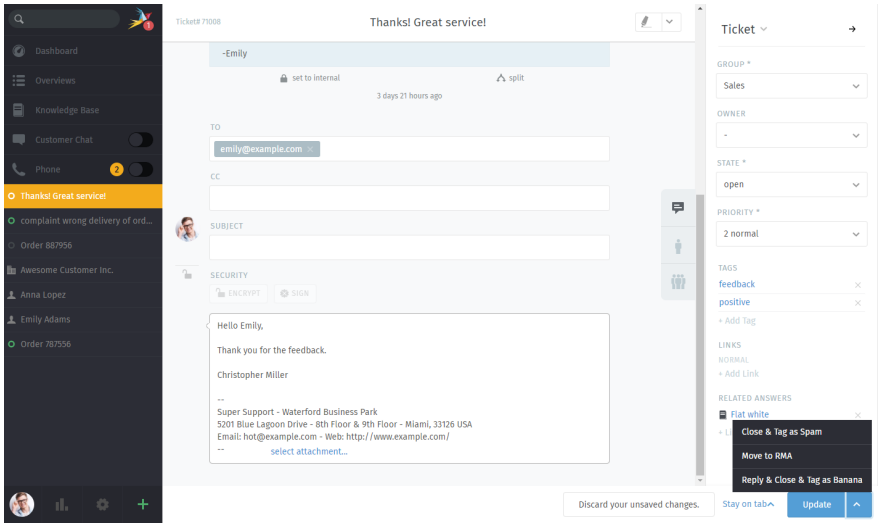


Fig. 1: Choose from the macros **Close & Tag as Spam**, **Move to RMA**, and **Reply & Close & Tag as Banana**.

You can also apply macros in bulk (*i.e.,* to many tickets all at once) via the *Overviews* page:

You can create or edit macros on the Macros page of the admin panel:

7.1 Learn by example

To get you up and running quickly, here are some examples of the kinds of one-click actions you can set up using macros.

Hint: If they don't make sense to you, don't worry—just skip ahead to *How do they work?* to learn about all the options in detail, then come back here to see them in action.

---

1. If you deal with a lot of spam, you could set up a macro that applies the following changes to a ticket:

   State closed

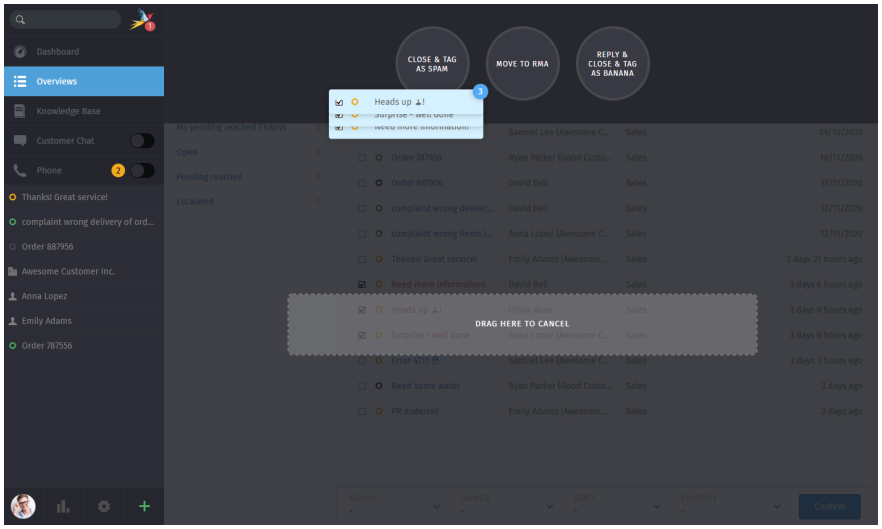
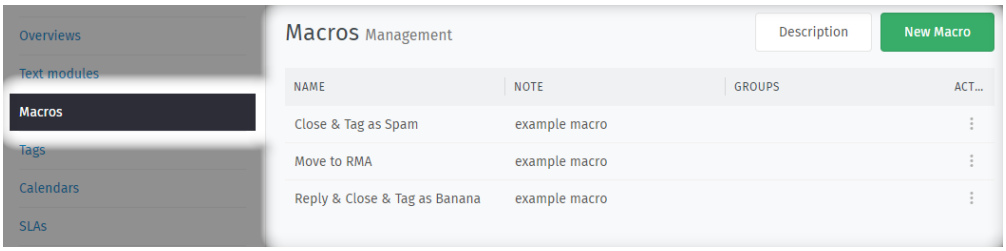


Fig. 2: Select your desired tickets, then click-and-drag to apply a macro to all of them.



Tags add `spam`

Owner current user

Tip: Run this macro in a *Scheduler* to periodically clean up unwanted tickets.

---

2. If you want to set a ticket's state to *pending reminder*, it's usually a two-step process—first select the state, then select a date. To always set a reminder for the same, fixed amount of time (say, seven days later), you can bundle the whole change into a macro:

   Note "Postponing ticket for 7 days." (internal visibility only)

   State pending reminder

   Pending Till relative / 7 / days

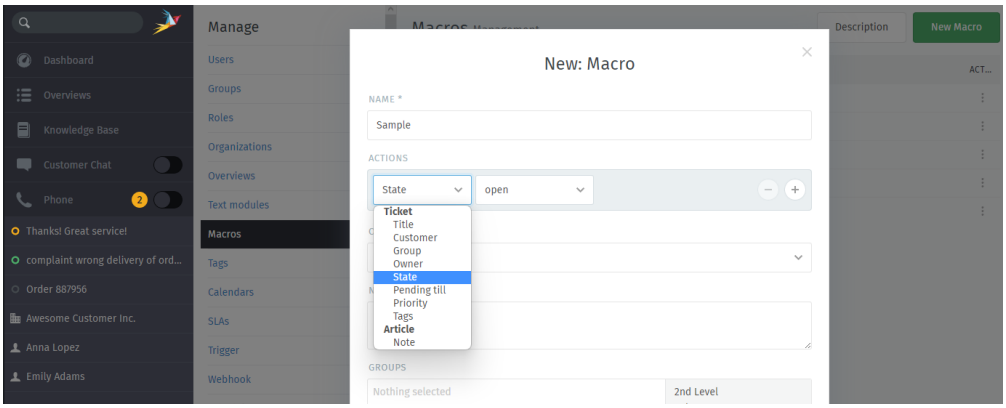
   Owner current user

7.2 How do they work?

Macros are made up of **actions** (changes to a ticket). You can add as many actions to a macro as you want.

There are also a few other settings that affect who can use a macro or how it behaves.

7.2.1 Creating Macros



Actions You can create actions to:

- set ticket attributes (priority, state, group, etc.)
- add new notes to a ticket

There are **no** actions for:

- sending a reply to the customer

Note: Unlike triggers, the scheduler, and text modules, macro actions do **not** support the use of *Variables*.

Warning: If the ticket is missing a required attribute and the macro doesn't set it, then **no actions will be applied**.

Once completed… After running this macro, should Zammad remain on the current tab, close it, or automatically switch to the next ticket? (Does not apply when running macros "in bulk".)

Note What should other Zammad admins know about this macro? (Visible only via the "Edit: Macro" dialog, Rails console, and API.)

Groups Which *Groups* are allowed to see/use this macro?

Active Choose "inactive" to disable this macro without deleting it.

7.2.2 Managing Macros

You can delete or even clone existing macros in the Admin Panel under **Manage > Macros**.

Fig. 3: When cloning a macro, you *must* click “Submit” for the duplicate to be created.

CHAPTER

EIGHT

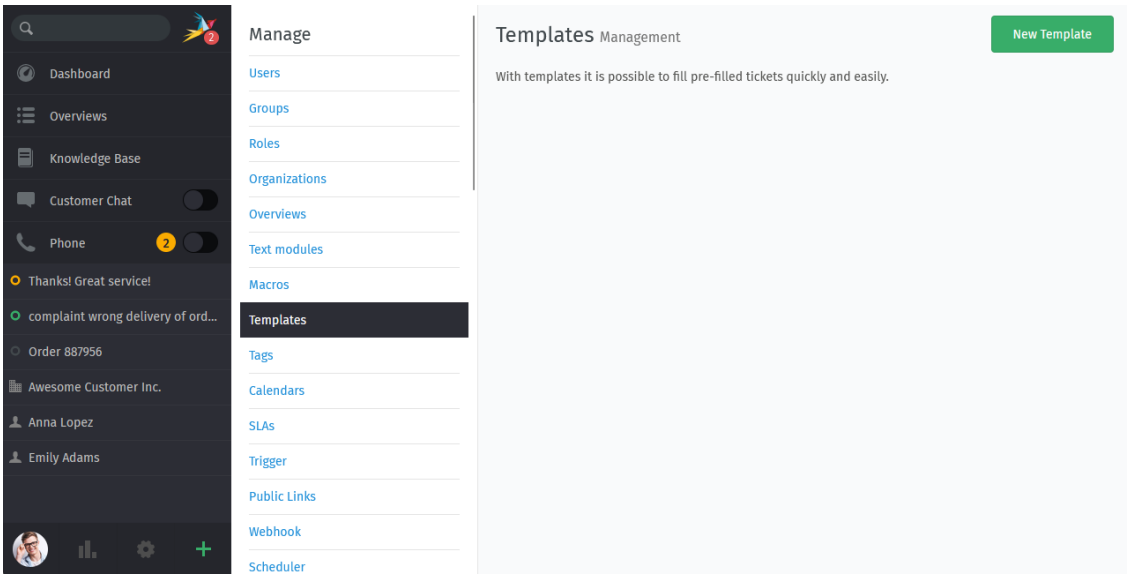
TEMPLATES

Templates allow you to create tickets at a rapid speed by defining a ticket template for recurring tickets.

Note: Sounds familiar

Right! Prior Zammad 5.3 ticket templates where managed entirely within the ticket zoom.

This has changed:
Your agents by default no longer have permission to manage templates by default. However, they can always *load* existing templates into their new ticket dialogue.



Warning: Limitation ahead

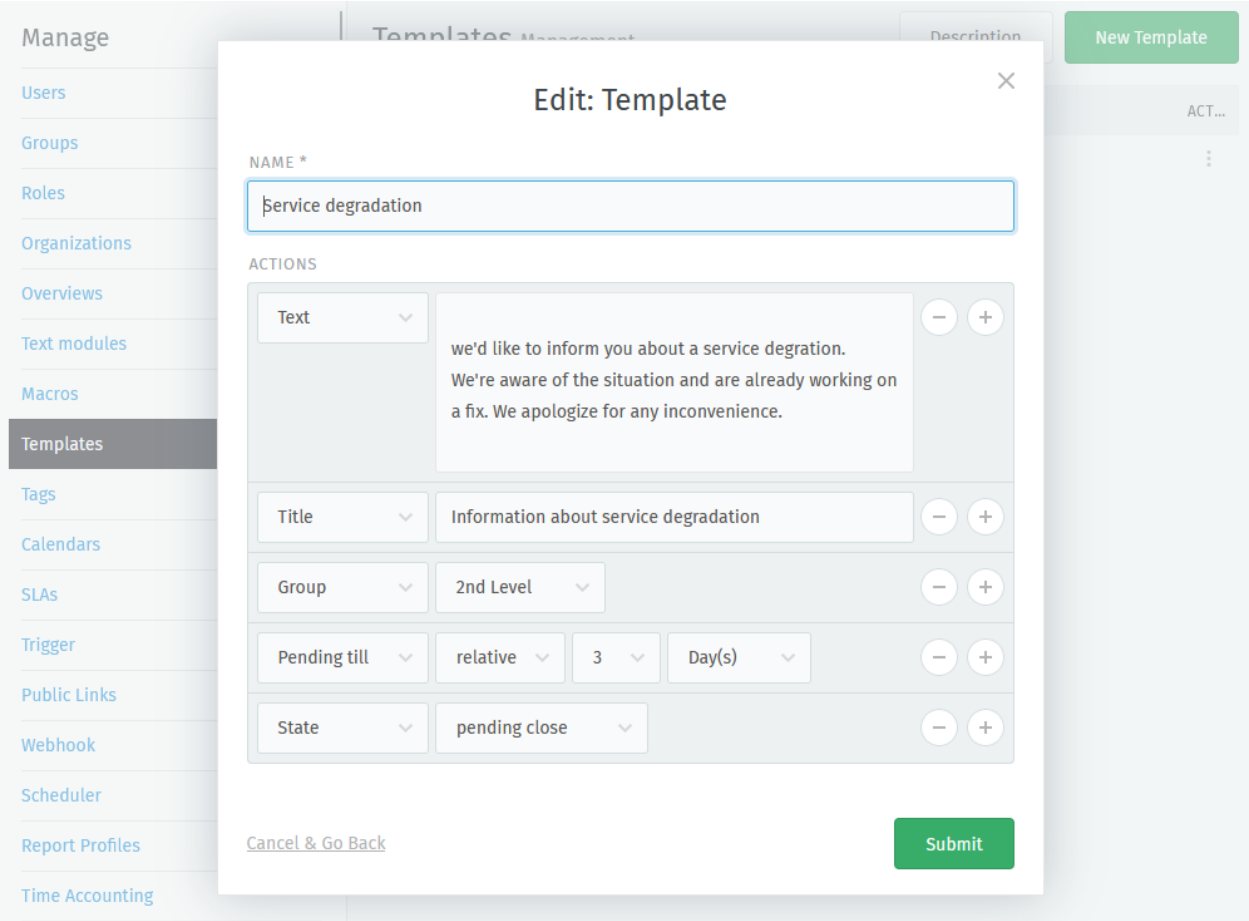
Please note that ticket templates do not allow the use of variables.

8.1 Managing templates

Adding new template Use the *New Template* button to start creating a new template.

Name Select a meaningful template name. This name will be shown to your agents during ticket creation (within *Select Template*).

Actions Within actions, Zammad will provide all available ticket attributes. This allows you to create granular templates with the information you need.



Editing templates If your template no longer fits your need or contains errors, simply click on the template name to edit it.

Cloning templates Zammad allows you to clone existing templates and continue your new template from there.

To do so, use *Actions* of the desired template in your list and choose *Clone*. A new template modal will open with attributes prefilled.

Removing templates If you longer require a specific template, use *Actions* in the list and choose *Delete*.

> **Danger:** Removals are permanent.

CHAPTER

NINE

TAGS

With tags, tickets can be categorized and marked.

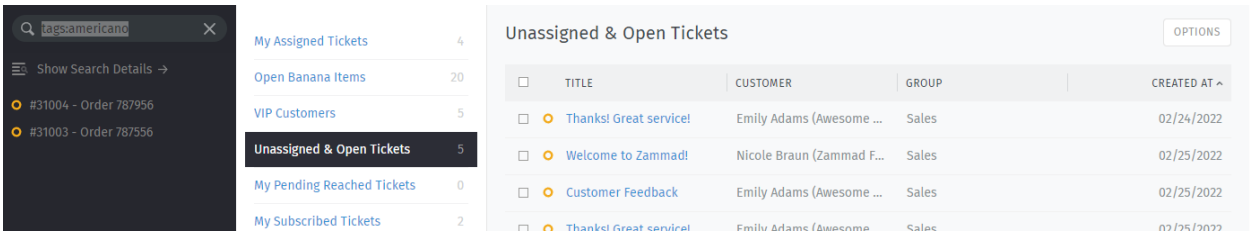
9.1 How to tag

All agents can tag tickets. You can find further information about tagging of tickets on our user documentation.

9.2 Tag-Overviews

You can list all the marked tickets by tag in different ways:

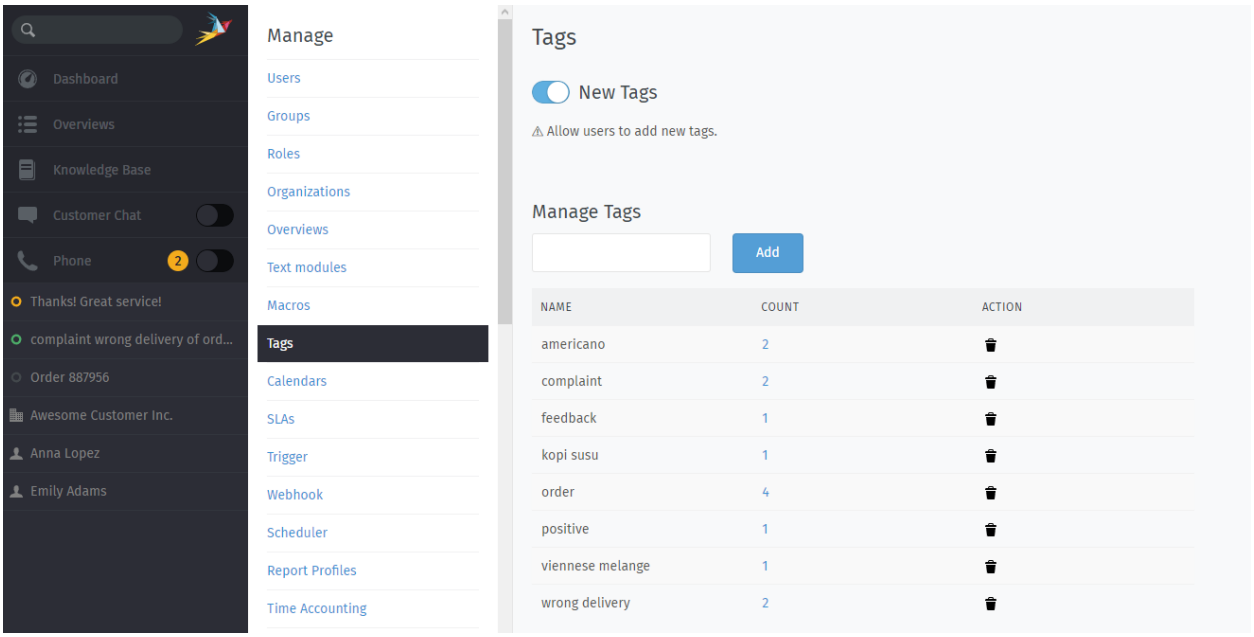
1. by providing an appropriate overview Create a new overview and select "Tag contains ..." as condition. You can find further information in *Overviews*
2. by a search-query Enter e. g.: `tags:americano` in the search mask and you will receive all tickets that have been tagged accordingly. For a collected listing, click on "Show Search Details" under the search box:



3. by clicking on the tag in the Tag Management Area In the Tag Management you can find an overview of existing tags used in the system (and a counter how often they are used):

   For more information about the individual tags, just click on the blue counter:

   Select one of the tickets that appears below the search box or get a complete overview of all tickets by clicking on "Show Search Details":



9.3 Tag Management Area

Here it can also (in addition to the overview - as described above) be set whether the agents are allowed to add tags themselves or not (in the left upper corner).

You can add more tags or delete them.

CHAPTER

TEN

CALENDARS

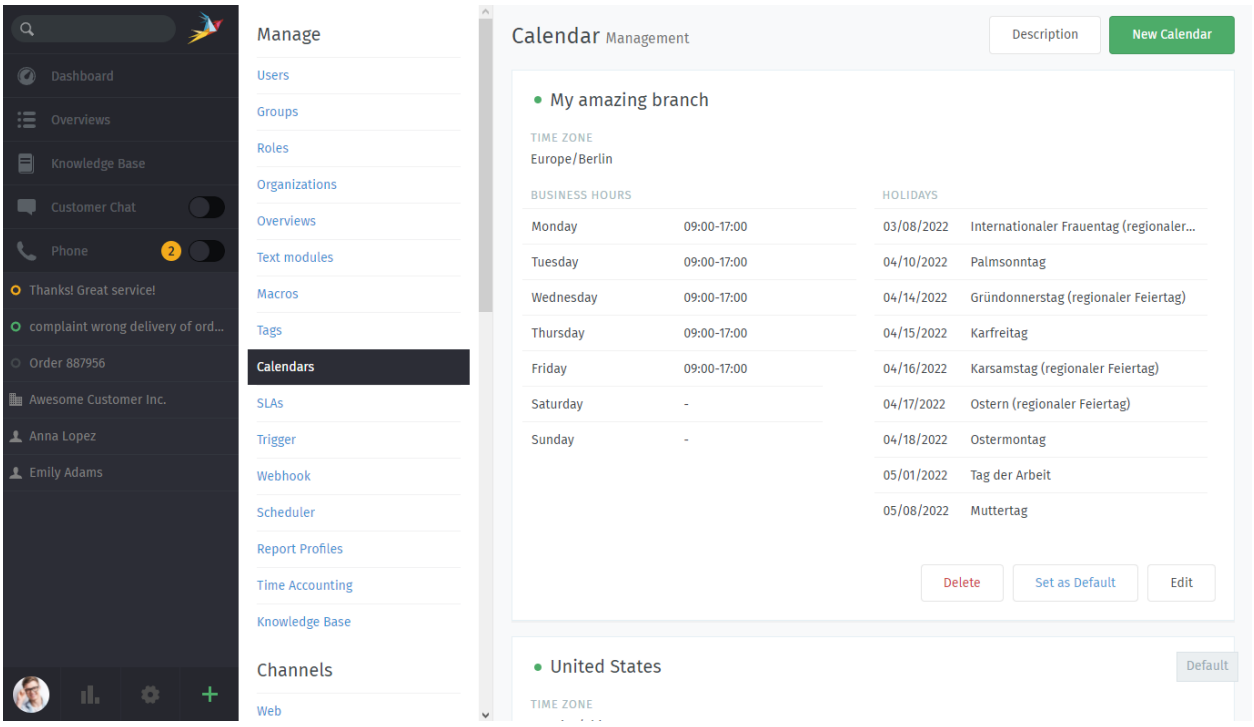
A calendar is required to:

- automate ticket escalations,
- generate reports that only capture activity during business hours, or
- set up time-sensitive triggers.

Define a “standard”-calendar which system-wide is valid. Only in the specified business hours, escalation notifications are sent to agents.

If you have customers for which you have to comply with different customer business hours, you can create several calendars. The allocation to the customer tickets can be adjusted via the SLAs.

And this is what it looks like:



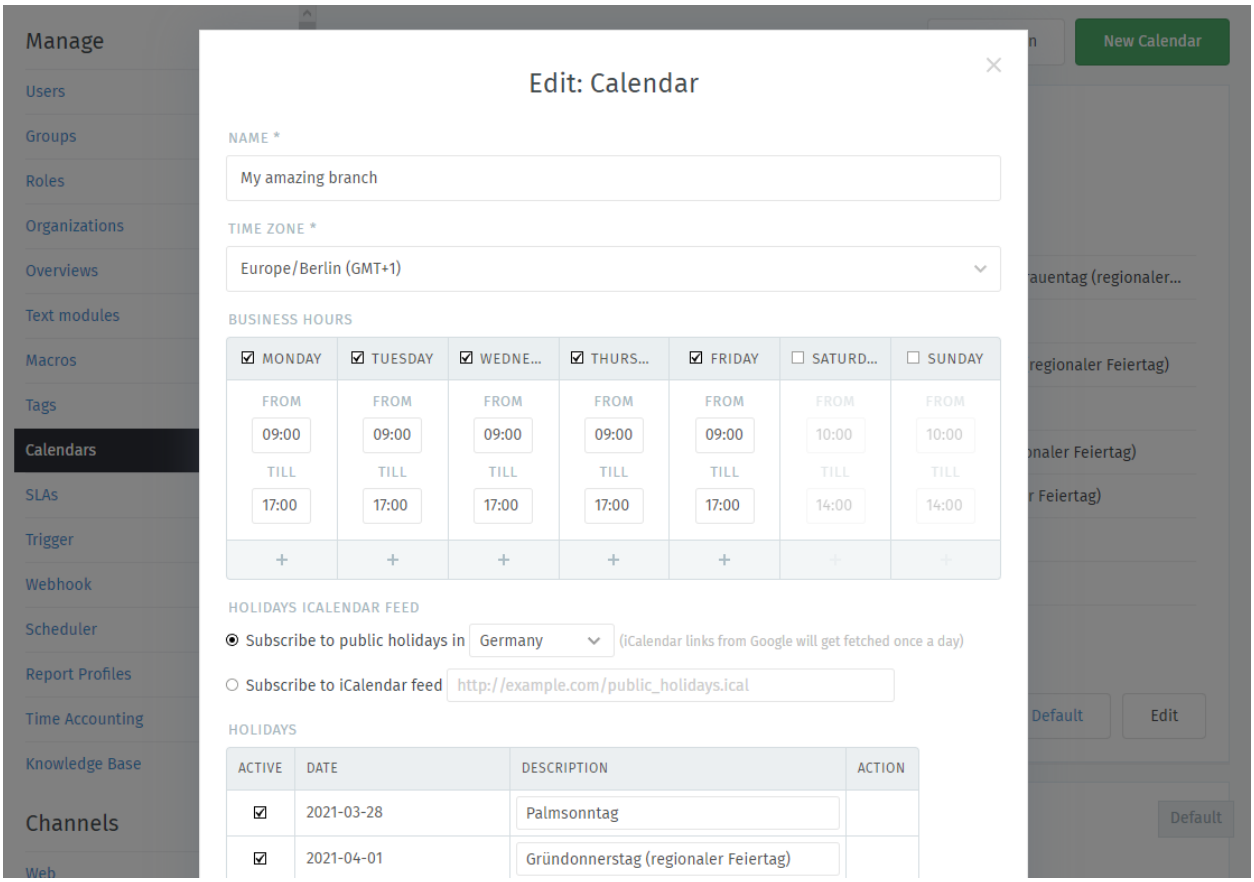
All created calendars are displayed in the overview.

New Calendar Here you can create a new calendar if agents or customers belong to another time zone.

Delete Just push the delete-button to delete this specific calendar - all SLAs assigned to this calendar are automatically assigned to the default calendar.

Set as Default Pressing this button sets this calendar as the default calendar for the entire system.

Edit Via this button you get to the edit-mask (same mask as in "New Calendar"):



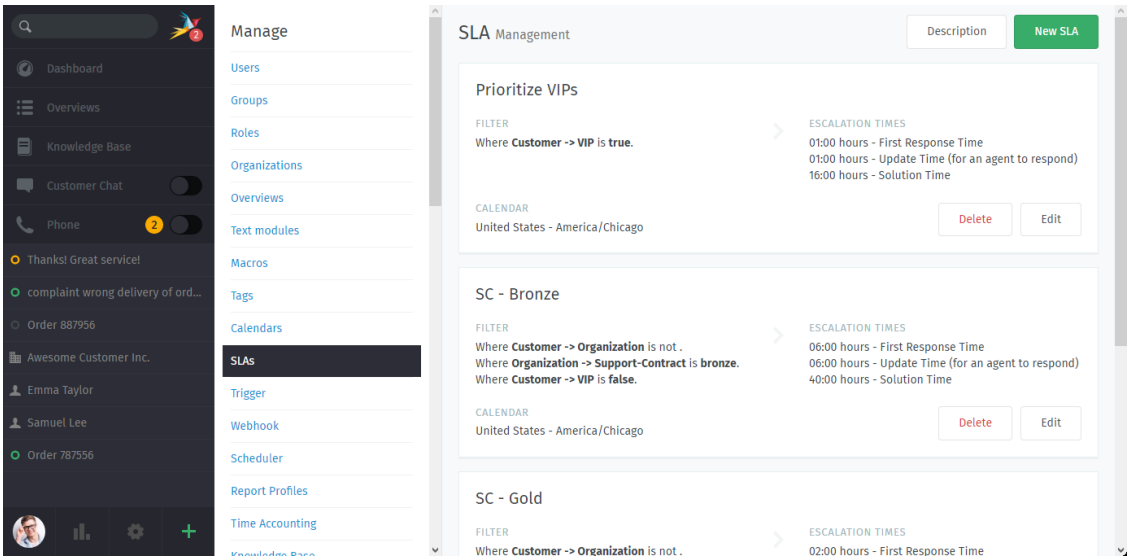
–> determine a name, a time-zone, the business hours to be used for this calendar and special holidays. In addition, you can subscribe to the iCalendar, which will automatically load all holidays from Google (updated once a day) … and you can add a note.

CHAPTER

ELEVEN

SLAS

Service Level Agreements, abbreviated SLAs, help you to meet specific response times for your customers' requests. This way you can define goals such as answering every inquiry within eight hours. If you are at risk of missing this target, Zammad will alert you.



Agents will be notified via several, optional ways. You can provide overviews for escalated or soon to escalate tickets to help you agents. Also, agents can check the current applying SLA timings of tickets at any time.

Online notifications Zammad will warn agents roughly 15 minute before escalation and when the ticket finally escalates. This depends on the notification configuration of your agent.

Email notifications Zammad will warn agents roughly 15 minute before escalation and when the ticket finally escalates. This depends on the notification configuration of your agent.

X

Overviews You can configure *Overviews* to allow your agents to filter for escalated tickets or those that are about to escalate. By default Zammad comes with an overview that will show all tickets that are either escalated or escalating within the next 10 minutes: `Escalated Tickets`.

Tip: If you have all/most of your tickets covered by SLAs you may consider changing the default sorting of existing overviews by escalation time, instead of creation time.

SLA timings of a ticket Ticket zooms provide a timestamp of the next escalation of a ticket. Agents can hover that timestamp and learn about all upcoming escalation stages.

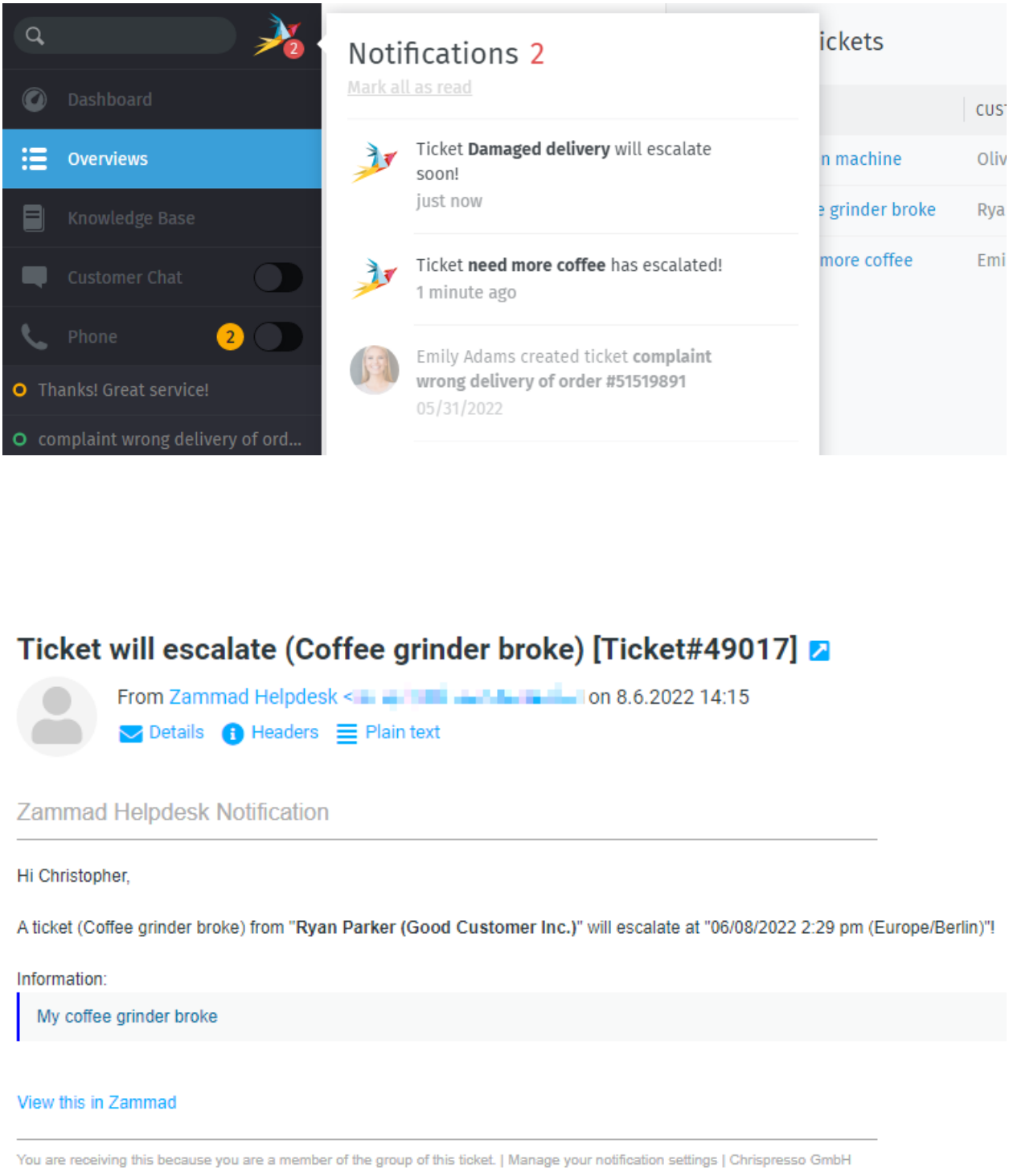


Fig. 1: Notification by mail that a ticket is going to escalate.

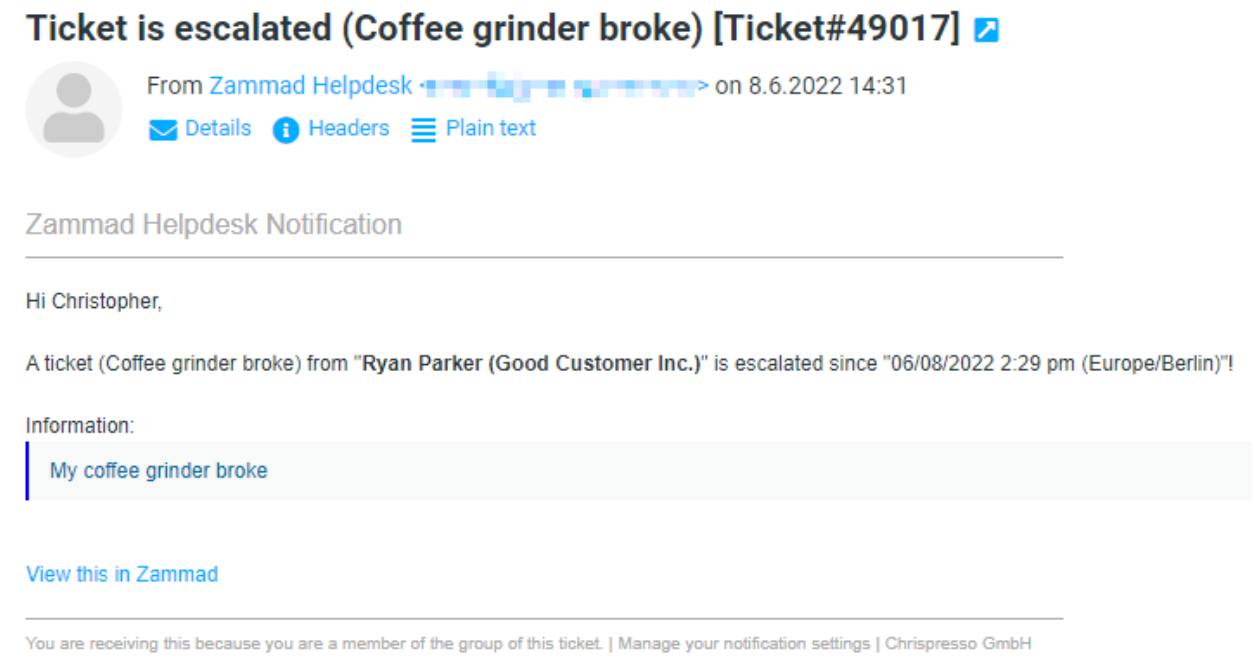


Fig. 2: Notification by mail that a ticket has escalated.

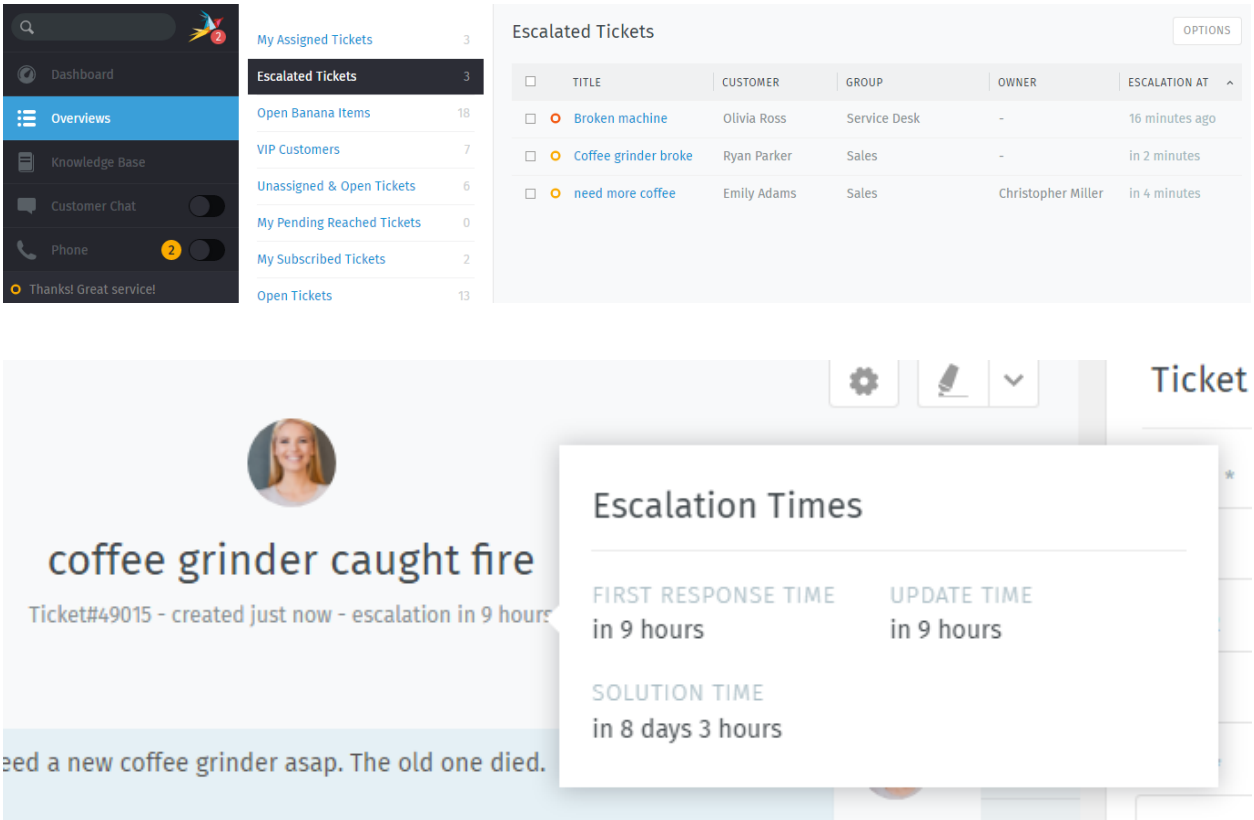


Fig. 3: A fresh ticket without any response by agents.

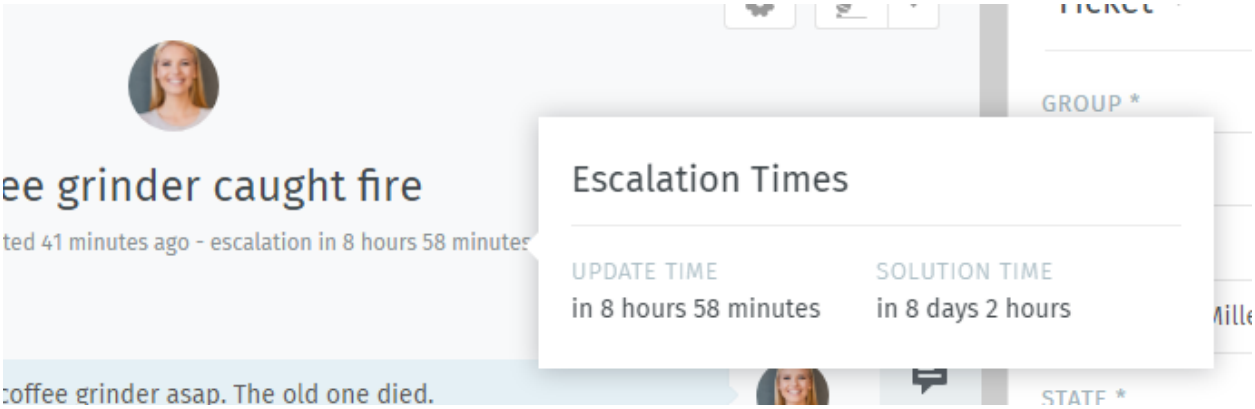


Fig. 4: A ticket after the agents initial response and a customer response.

X

11.1 Learn by example

This page contains some possible example configurations for a SLA we could think of. SLAs in general are very flexible and powerful, below doesn't have to be your standard!

Hint: If they don't make sense to you, don't worry—just skip ahead to *How do they work?* to learn about all the options in detail, then come back here to see them in action.

All following SLAs have the same base configurations. They may not use all of them in all samples.

- **Groups**
  - Sales
  - Support
  - 2nd Level
- **Attribtues**
  - User / VIP (default, Boolean)
  - Organization / Support-Contract (Single selection field, `None` as default value)
- **Calendar setting** The calendar is set to 8 hours per day, from Monday to Friday.

1. Support contract levels and VIP customers

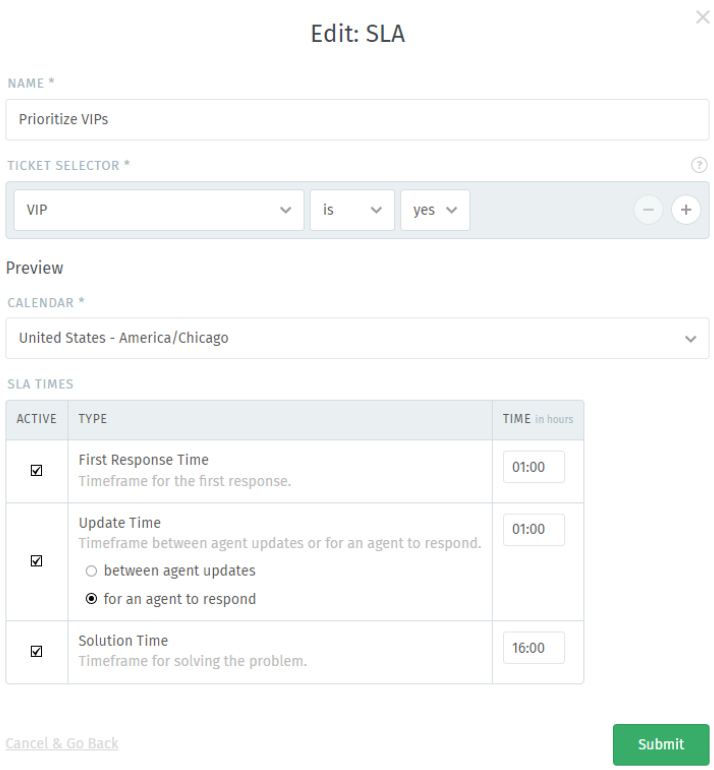
This approach uses a combination of contract levels (where any level except for none causes SLA escalations) and VIP customers that can have further priority.

The configuration of these SLAs ensures that either fitting support contract levels trigger or the VIP state is being used. They're not overlapping.

VIP customers

SLA configuration checking if the customer is a VIP customer.

Support contract levels



The settings of each support contract level SLA basically are the same. The only setting that differs is the support contract level of each configuration and the SLA timings.

For this reason we're showing the bronze level configuration as an example only.

The result This approach causes the following SLA timings for each level:

If the customer has no VIP state and the organization has a support level

Bronze

First response time: `6 hours`
Update time (for an agent to respond): `6 hours`
Solution time: `40 hours`

Silver

First response time: `4 hours`
Update time (for an agent to respond): `4 hours`
Solution time: `32 hours`

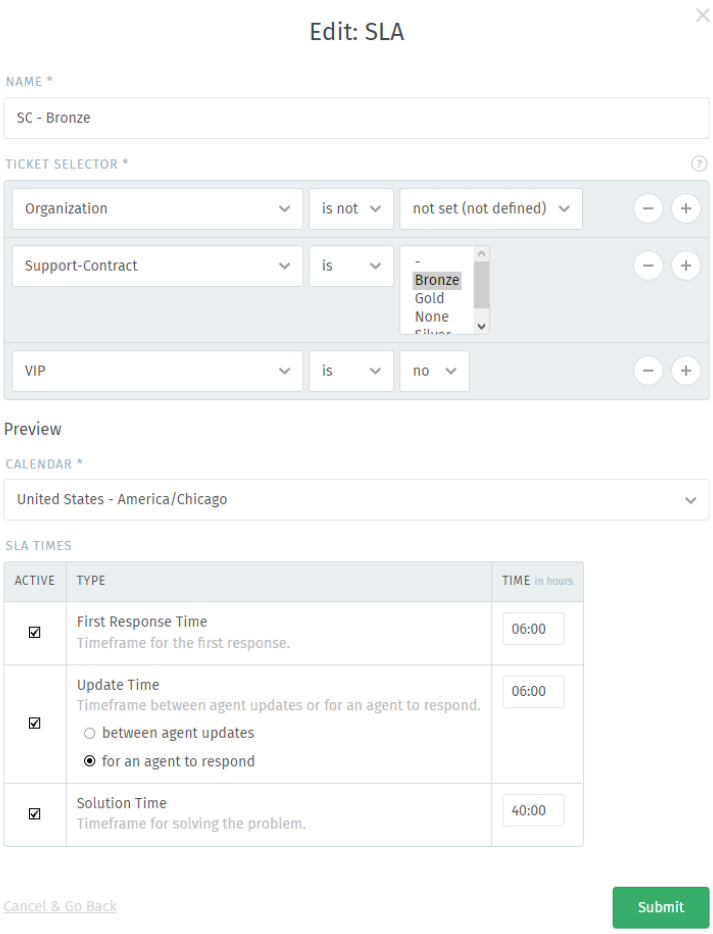
Gold

First response time: `2 hour`
Update time (for an agent to respond): `2 hour`
Solution time: `24 hours`

None This setting has no SLA configuration. Tickets will not escalate.

If the customer has the VIP state

First response time: `1 hour`

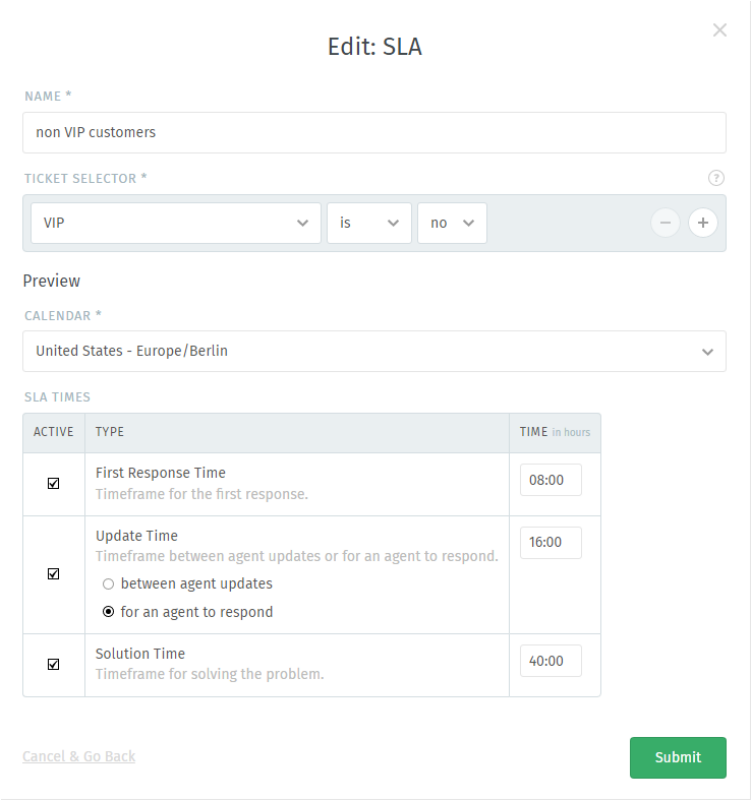

Edit: SLA
NAME *
SC - Bronze
TICKET SELECTOR *
Organization
is not
not set (not defined)
Support-Contract
is
-
Bronze
Gold
None
VIP
is
no
Preview
CALENDAR *
United States - America/Chicago
SLA TIMES
ACTIVE
TYPE
TIME in hours
First Response Time
Timeframe for the first response.
06:00
Update Time
Timeframe between agent updates or for an agent to respond.
between agent updates
for an agent to respond
06:00
Solution Time
Timeframe for solving the problem.
40:00
Cancel & Go Back
Submit

Update time (for an agent to respond): `1 hour`
Solution time: `16 hours`

2. **VIP customers**

   non VIP customers

   SLA configuration checking if the customer is not a VIP customer.



   VIP customers

   SLA configuration checking if the customer is a VIP customer.

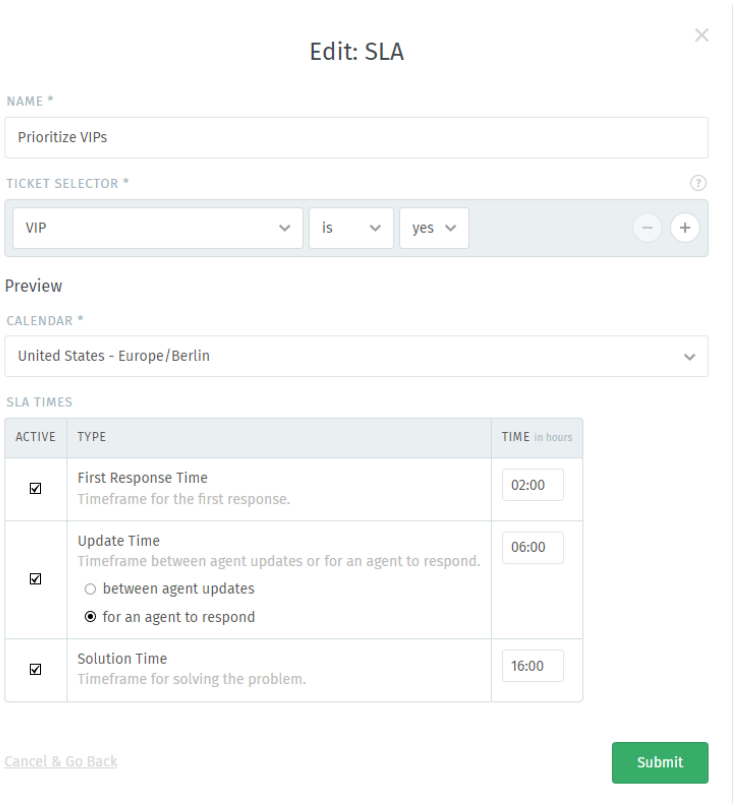
   The result

   If the customer has the VIP state

   First response time: `2 hour`
   Update time (for an agent to respond): `6 hours`
   Solution time: `16 hours`

   If the customer has no VIP state

   First response time: `8 hours`
   Update time (for an agent to respond): `16 hours`
   Solution time: `40 hours`



11.2 How do they work?

You can define several independent SLAs, however, ensure to have no overlapping SLAs as their calculations may not work as you intended them originally.

Warning: Updating existing SLA configurations *may* cause temporary performance issues depending on your instance size and affected tickets. You may want to perform those changes outside of your business hours.

Tip: Communication type articles are not enough for your SLA? You can also configure Zammad to allow **public** notes as fulfillment for SLA requirements, our console documentation tells you how.

Name Give your SLA a meaningful name to quickly see what or who is affected.

Ticket selector Specify the conditions on which tickets this SLA should apply to.

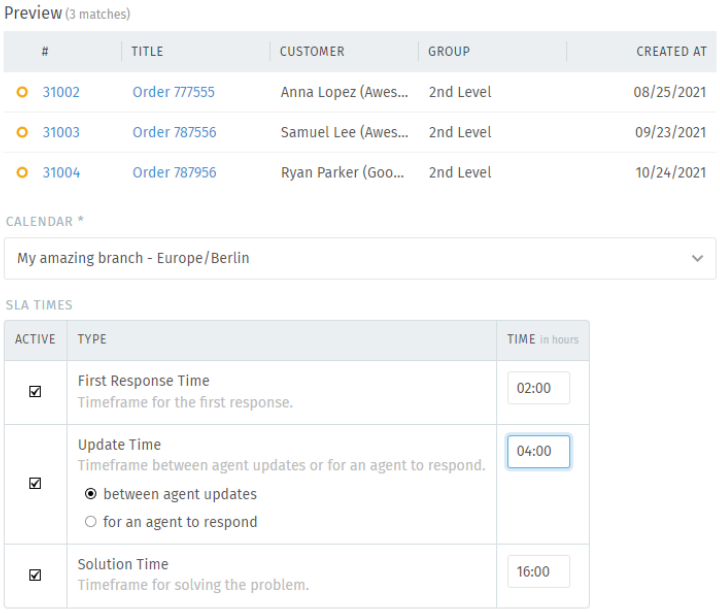
Preview In the preview you see the selection of the tickets and double check whether those are correct.

Calendar Zammad calculates ticket escalations based on your business hours. It makes no sense to escalate tickets when you're not in.

Choose the correct calendar here. If you haven't set your business hours yet, see *Calendars* to learn more.

SLA Times Define the escalation timings based on your working hours. Keep in mind that if you defined 8 hour business hours per day, a 16 hour escalation will take two business days.

First Response Time frame for the first response (external call, email). This time is calculated from the ticket creation.



Note: The start time *does not* reset if you're e.g. moving the ticket into new groups or different SLA calculations. That is because the creation time of the ticket does not change - keep this in mind.

Update Time Time frame for every following response (external call, email). This time either counts from ticket creation (if no first response is set) or *after* the first response was done.

between agent updates In this scenario your agent have to respond every n hours depending on your configuration. This causes Zammad to not care if your customer replied or not.

> **Warning:** This setting can be quiet stressful for your agents.

for an agent to respond From the moment your customer replied to the ticket, your agents have the configured time amount to respond until the ticket escalates.

Solution Time Time frame for solving the problem (setting the ticket to a closed state type).

> **Warning:** This escalation timing *does not* care about ticket responses. It's being calculated from the ticket creation.

It is up to you if you set one, two or all three times. When the SLA time is reached, the ticket escalates.

Hint: *First response* and *Solution time* can only apply once. Update time applies until the ticket has been solved or the SLA no longer matches your criteria.

The only way to stop escalations in default installations is to set the ticket to either a pending reminder or pending close. **However**: as soon as the ticket is being set to open (e.g. by a customer responding) the ticket may instantly escalate depending on its age and your configuration.

Note: Ticket escalations will notify all affected agents. This highly depends on their chosen notification settings (Profile). Escalated tickets can also be filtered for:

- by searching
- *Triggers*
- *Scheduler*
- *Overviews*

CHAPTER

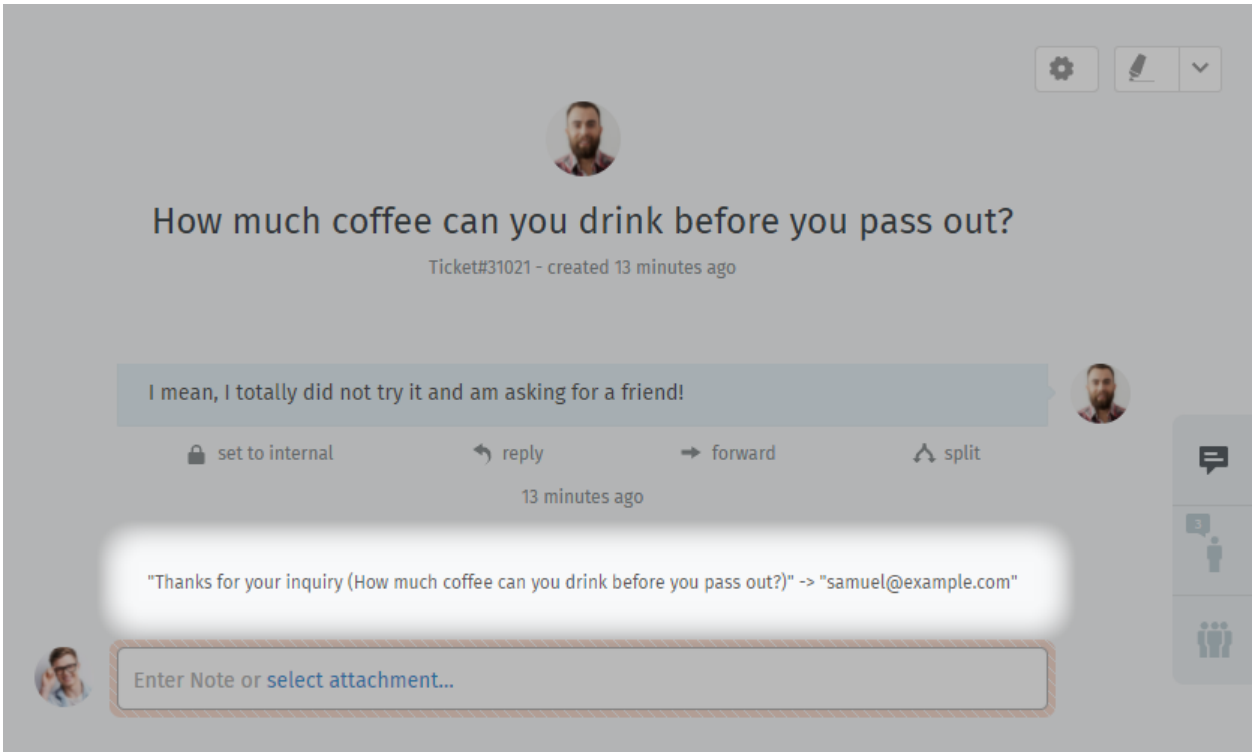
TWELVE

TRIGGERS

Use triggers to set up all kinds of **if-this-then-that** automation workflows.

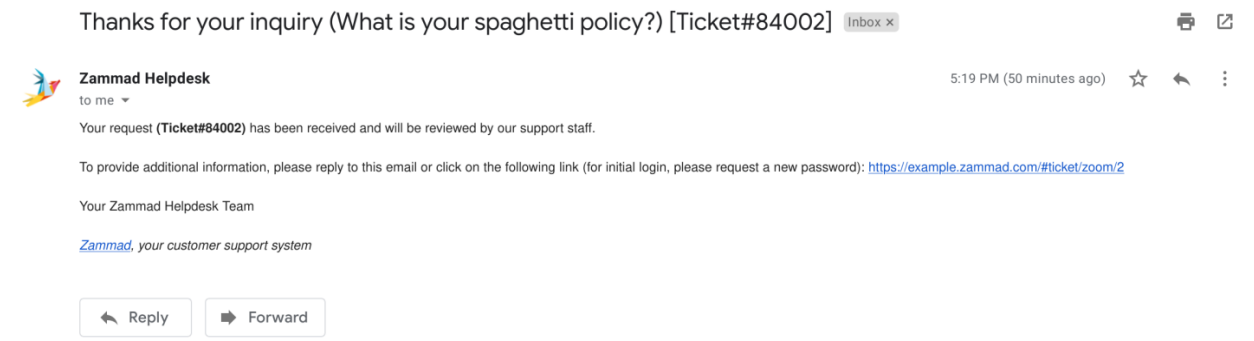
Hint: For **every-so-often** automation workflows, try *schedulers* instead.

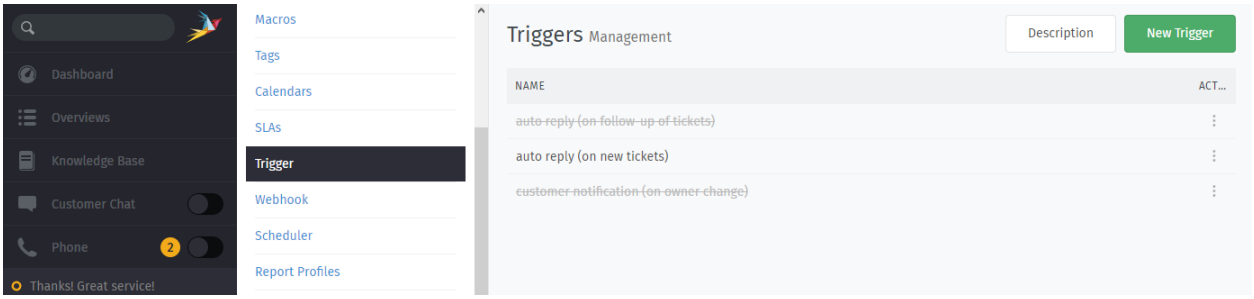
The first thing to know about triggers is that **you're already using them**. From the moment you set up Zammad, it starts sending auto-replies to all incoming emails. Recognize this line?



It means that the ticket is from an incoming customer email, and that the customer received this message in response:

This auto-reply message is a **trigger**. You can disable it, modify it, or create new ones for all sorts of automation tasks on the Trigger page of the admin panel:





12.1 Learn by example

To get you up and running quickly, here are some examples of the kinds of automation tasks you can set up with triggers.

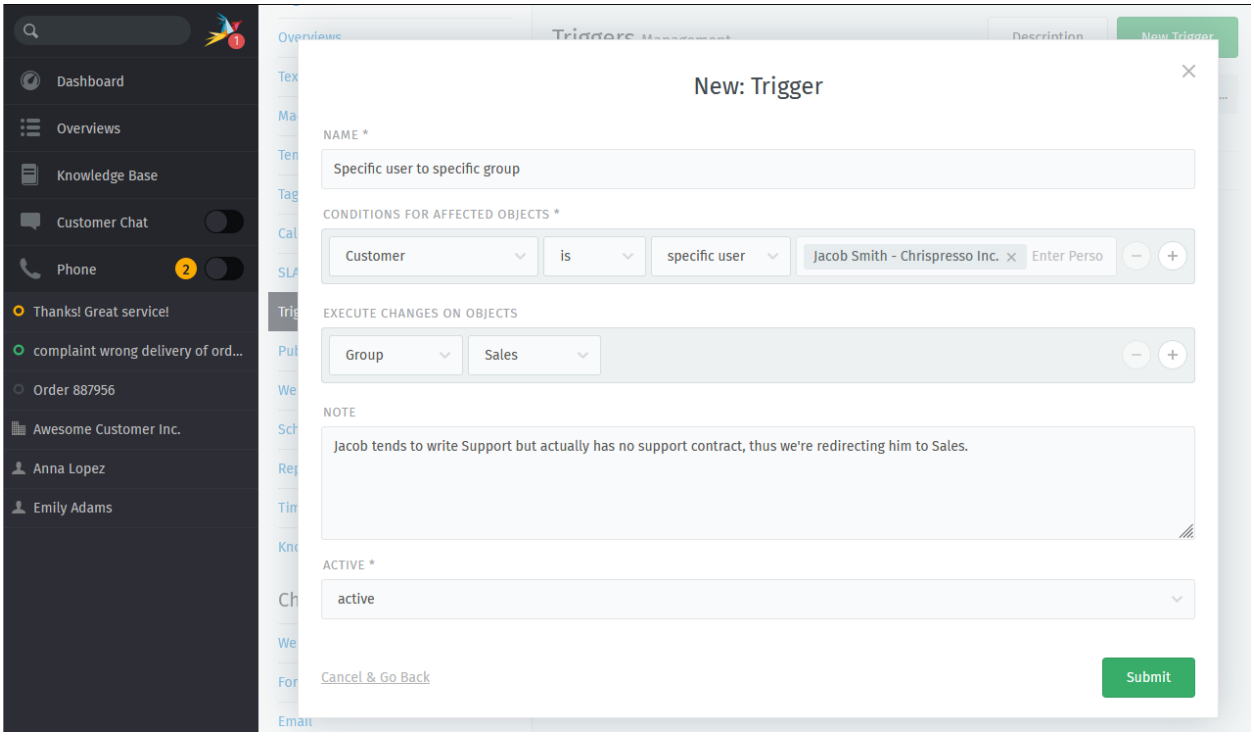
Hint: If they don't make sense to you, don't worry—just skip ahead to *How do they work?* to learn about all the options in detail, then come back here to see them in action.

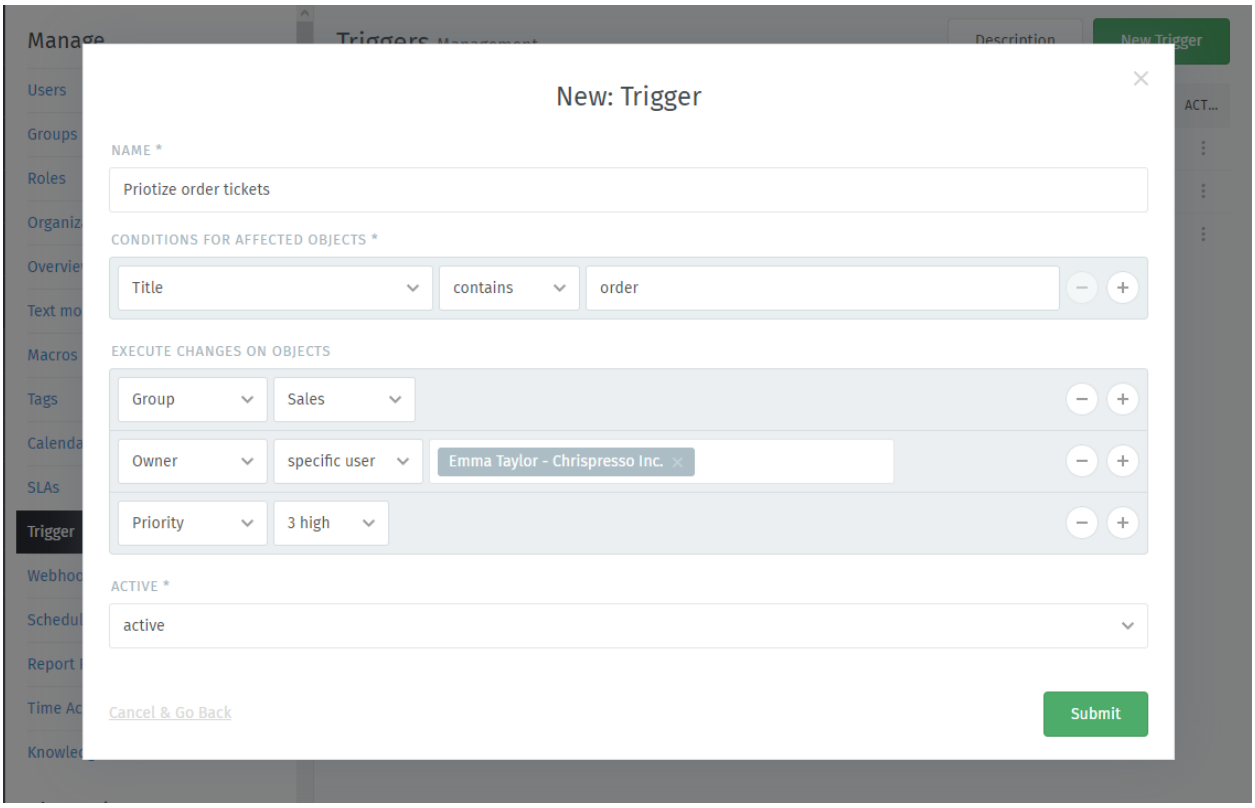
---

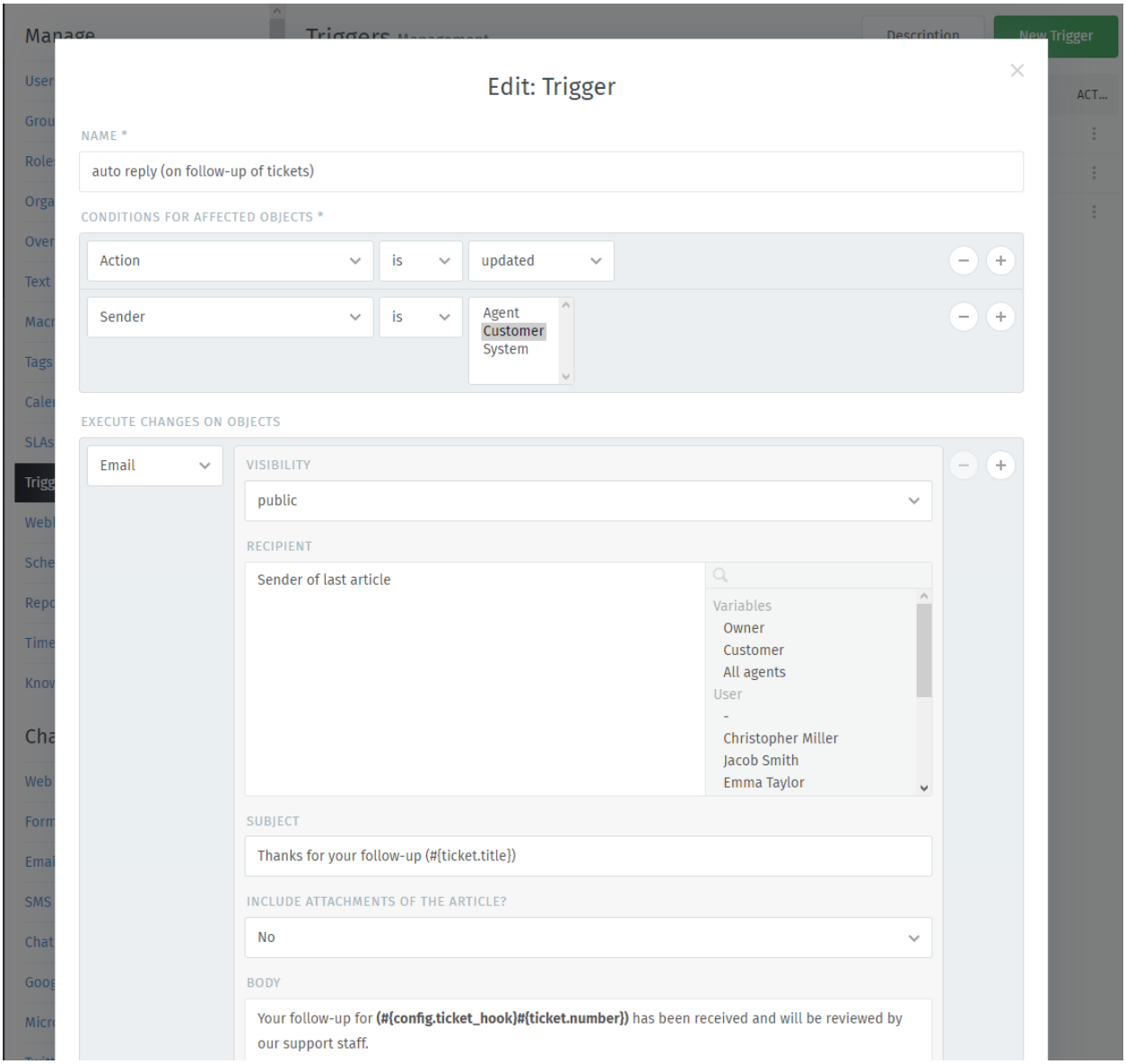
1. Any time Jacob Smith creates a ticket, assign it to the Sales group:
2. Emma Taylor is responsible for all sales internally, so if a new ticket has the word "order" in the subject, assign it to her and make sure it's set with a high priority:
3. Send an auto-reply email to *any customer* who responds to a ticket:

Note: Not all automated messages come from triggers!

For instance, when *agents* receive a system email about a newly created ticket, that's built into the system itself. If you need to customize those, you will have to *manually edit files on your server*.


Dashboard
Overviews
Knowledge Base
Customer Chat
Phone
Thanks! Great service!
complaint wrong delivery of ord...
Order 887956
Awesome Customer Inc.
Anna Lopez
Emily Adams
New: Trigger
NAME *
Specific user to specific group
CONDITIONS FOR AFFECTED OBJECTS *
Customer
is
specific user
Jacob Smith - Chrispresso Inc.
EXECUTE CHANGES ON OBJECTS
Group
Sales
NOTE
Jacob tends to write Support but actually has no support contract, thus we're redirecting him to Sales.
ACTIVE *
active
Cancel & Go Back
Submit


Manage
Users
Groups
Roles
Trigger
New: Trigger
NAME *
Priotize order tickets
CONDITIONS FOR AFFECTED OBJECTS *
Title
contains
order
EXECUTE CHANGES ON OBJECTS
Group
Sales
Owner
specific user
Emma Taylor - Chrispresso Inc.
Priority
3 high
ACTIVE *
active
Cancel & Go Back
Submit


Edit: Trigger
NAME *
auto reply (on follow-up of tickets)
CONDITIONS FOR AFFECTED OBJECTS *
Action
is
updated
Sender
is
Agent
Customer
System
EXECUTE CHANGES ON OBJECTS
Email
VISIBILITY
public
RECIPIENT
Sender of last article
Variables
Owner
Customer
All agents
User
-
Christopher Miller
Jacob Smith
Emma Taylor
SUBJECT
Thanks for your follow-up (#{ticket.title})
INCLUDE ATTACHMENTS OF THE ARTICLE?
No
BODY
Your follow-up for (#{config.ticket_hook}#{ticket.number}) has been received and will be reviewed by our support staff.

12.2 How do they work?

Triggers consist of two parts: **conditions** and **changes**. Conditions answer the question, "when should this trigger fire?" Changes answer the question, "what should happen when it does?"

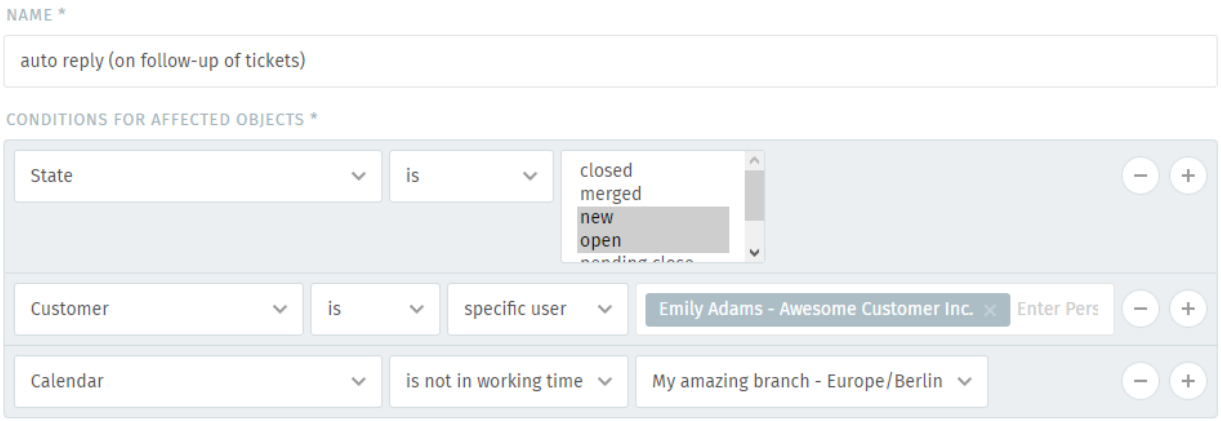
Triggers are evaluated in alphabetical order, by **name.** In some situations triggers might be the wrong choice, see *Limitations* for more information.

Hint: Email trigger behavior can be manipulated

Please have a look at *Email header manipulation* in case this is a relevant use case for you.

12.2.1 Conditions

When creating a trigger, define your conditions here:

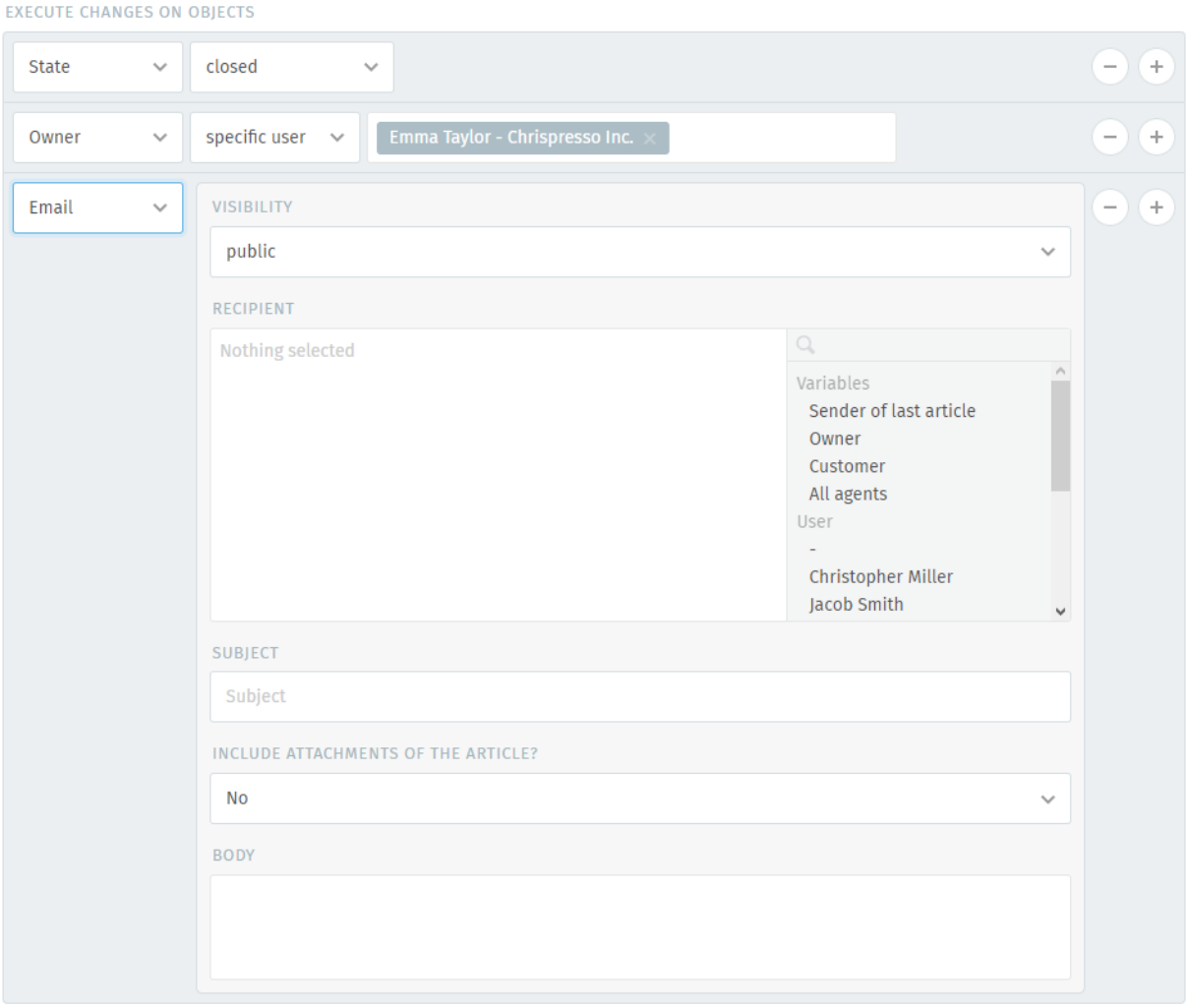


Trigger conditions are **and-selectors** and thus all conditions must apply as configured for the trigger to fire. You can configure triggers to fire based on the properties of:

- **The Ticket itself** *e.g.,* Was this ticket newly created? Is the ticket currently open? When was the last time we received contact from the customer on this ticket?
- **New Articles on the ticket** *e.g.,* Was this article added by email? by phone? Was it created by an agent, or a customer? Does the subject contain a certain set of words?
- **The Customer that created the ticket** *e.g.,* What is the customer's name? Is the customer a VIP? What department does the customer work in?
- **The Organizations that the ticket's customer belongs to** *e.g.,* What is the name of the customer's organization? Does it have a note attached to it containing a certain set of words?
- **The Execution time the trigger is being triggered** *e.g.,* Only send an auto-reply if the message was received outside of regular business hours. ("Regular business hours" can be defined on *Calendars* setting.)

12.2.2 Actions

When creating a trigger, define your changes here:



A trigger can do the following things once its conditions have been met:

- **Modify the ticket** *e.g.,* Escalate its priority, close it, reassign it, rename it, add tags, etc.

  Date & time attributes (like **Pending till**) can be specified in *absolute* or *relative* terms.

  Hint: You can also combine static text with placeholders for text fields. Remember that the placeholders' values have to be known during trigger runtime.

  Learn more about *Variables*.

- **Send an email or SMS** Either to the customer, the agent who owns the ticket, or every agent in the system.

  Tip: Sending emails allows you to include the attachments of the triggering article if required.

- ***Fire a webhook*** Connect Zammad to another web service or application to give it live updates about new tickets.
- **Add internal or public notes to the ticket** This allows you to help your agents with specific information if needed. (e.g. automated changes a trigger applied to the ticket)

Note: In order to send emails with Triggers, you need to configure an email address for the group the trigger is working in. If you don't, Zammad will skip the Trigger completely.

Hint: Certain actions (such as *email*, *SMS* and *notes*) support *Variables*, which can be used to build highly-customized message templates.

12.3 Limitations

It's important to understand when a trigger can be used and when it's better to use e.g. Zammad's *Scheduler* or *post-master filters*.

Triggers will fire during the following conditions:

- Creation of a ticket
- Updating a ticket

While the creation of tickets and triggering these actions is straight forward, updated of tickets are a bit trickier. In terms of triggers, a ticket is only updated if you press the update button on the lower right of a ticket. Adding tags to a ticket or switching articles visibility **is no ticket update**.

Also keep in mind that we're always only working on the last article. This means you can't trigger for past articles. Triggers always handle the current ticket attributes and the article (if applicable) that cause the trigger to fire.

If your use case doesn't fit in above possibilities, you might want to have a look at Zammad's *Scheduler*.

12.4 System Notifications

Note: System notifications can only be customized on **self-hosted installations**.

System notifications are automated emails sent by Zammad for critical system events, such as account changes or SLA violations.

Unlike the automated emails you can set up using *Triggers* or the *Scheduler*, these notifications are built into Zammad itself: if you need to customize them, you will have to modify some of the files on your server.

12.4.1 When are they sent?

All users are notified of:

- password change requests

Staff (admins & agents) are notified of:

- logins from a new device
- logins from a new country

Agents are notified of:

- new tickets
- ticket updates
- "ticket pending" reminders
- SLA violations (before and after the deadline)

Daily reminder emails are sent at midnight (UTC) for all unresolved "ticket pending" reminders and SLA violations.

12.4.2 How can I customize them?

Inside your Zammad directory (usually `/opt/zammad`), email templates for various events are stored inside the `/app/views/mailer` directory, named according to the language they're written in. Thus,

```
/opt/zammad/app/views/mailer/ticket_create/de.html.erb
```

is the German-language template used to notify agents whenever a new ticket is created. To modify this template, create another file with the same name and add a `.custom` suffix:

```
/opt/zammad/app/views/mailer/ticket_create/de.html.erb.custom
```

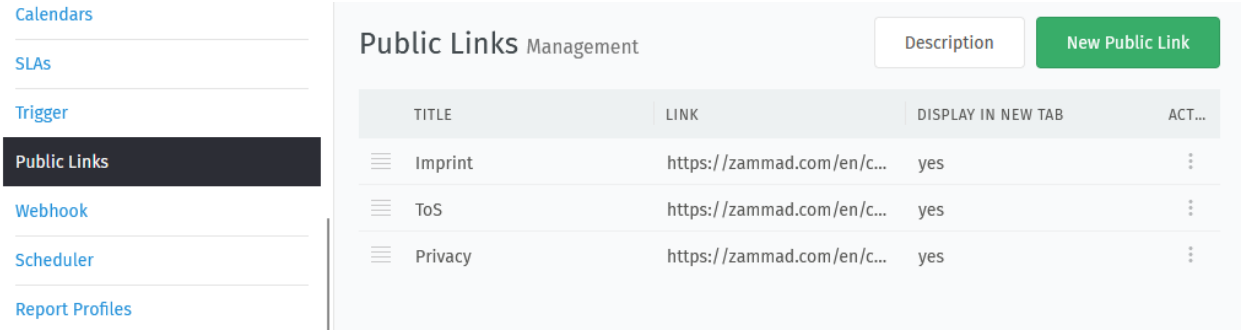
Now, this file will be used instead of the original when sending notification emails in German.

CHAPTER

THIRTEEN

PUBLIC LINKS

Public links allow you to provide important links at different places within the UI of Zammad. This allows you not just to provide legal information like e.g. data privacy or contact information. You can also provide further useful links to other services if needed.



Learn how to …

- *Adding new public links*
- *Manage existing public links*
  - *Re-arrange links*
  - *Actions for existing links* (Cloning & Removing)

13.1 Adding new public links

You can add new public links via the *New Public Link* button on the upper right. You'll see a new dialogue that consists of the following information.

Link This is the actual target page you're linking to.

Note: Zammad only allows URLs that start with either `http://` or `https://`.

Important: You may not use data privacy and terms of service URLs of `zammad.com` nor `zammad.org`.

Why? Every instance is special on its own. Every hosting is different, data retention and other things do not fit as every company using Zammad handles these matters differently.

This function was made to allow you to link to your *own resources*.

Title This is the regular text your user sees as the link text. You may want to keep the title as short as possible as it's displayed in the footer of the context you'll later select.

Description The description for URLs is an accessibility feature that helps users with screen readers to better understand the scope of the URL. It will also be shown by normal browsers when hovering over the link in question.

Use this to describe the link. This value is *optional*.

Context The context setting allows you to choose one or several places where this link should be displayed. Depending on your choice, Zammad will then show the links on the relevant pages.

You can currently select from:

- Forgot Password Screen

Note: This context *does not* affect the *Password Login* function.

- Login Screen
- Signup Screen

To help you understand scopes better, here's the different scopes as a screenshot. Note that we intentionally did set all links for all contexts.

Forgot Password Screen

Login Screen

Signup Screen

Display in new tab This setting allows you to determine if Zammad should tell the browser to either open the URL in a new tab or within the existing tab.

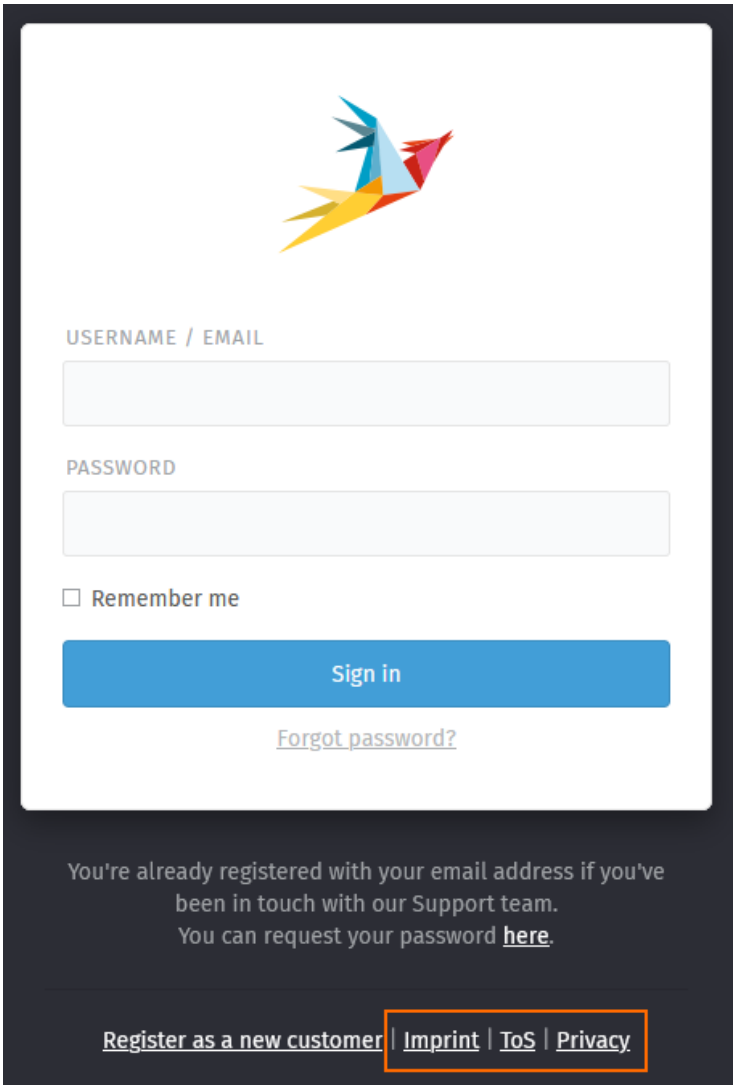
Opening URLs in the same tab may cause inconvenience to the user.
Default: `yes`

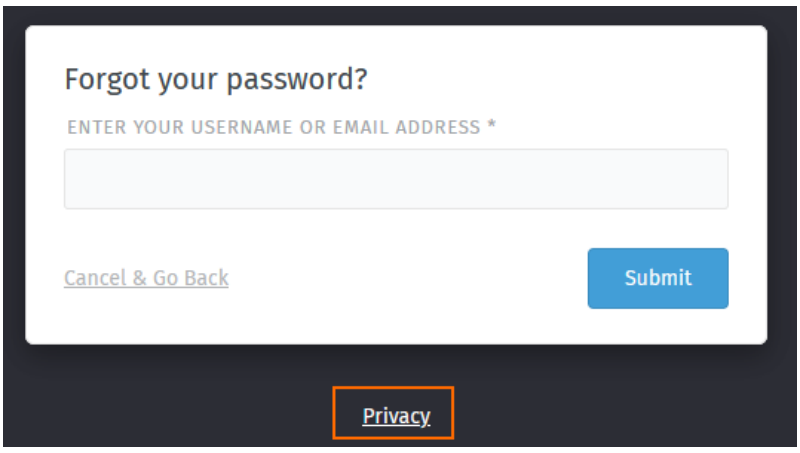
13.2 Manage existing public links

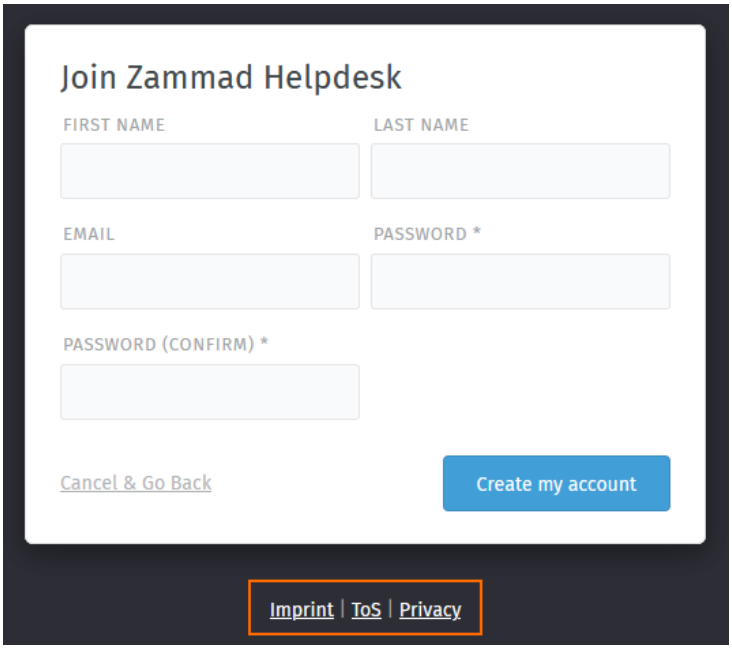
13.2.1 Re-arrange links

By default Zammad sorts your public links in the order you've created them. That may be a problem if you add a link later on. For this reason you can easily change the link order by dragging them to another position.

These changes are affective immediately.


USERNAME / EMAIL
PASSWORD
Remember me
Sign in
Forgot password?
You're already registered with your email address if you've been in touch with our Support team.
You can request your password here.
Register as a new customer | Imprint | ToS | Privacy


Forgot your password?
ENTER YOUR USERNAME OR EMAIL ADDRESS *
Cancel & Go Back
Submit
Privacy



13.2.2 Actions for existing links

Updating existing entries Got a typo in your URL or title? Want to change the context the link is being shown on? No problem! Just click on the affected link title and adjust the entry as needed.

You'll be given the same options as you have during the link creation.

Cloning If you want to add a new link that's very similar to an existing one, simply click on and select *Clone*.

Zammad will open a new public link dialogue with the existing settings filled in.

Removing If you no longer require a public link, you can remove it by using and selecting *Delete*. Zammad will ensure that you really want to remove the entry with a modal.

> **Danger:** Deleting is **final**. There's no way to bring back removed public links.

CHAPTER

FOURTEEN

WEBHOOK

Webhooks are a way to integrate Zammad with other web services or applications, allowing them to subscribe to live updates about tickets instead of having to poll the Zammad server every *n* minutes.

Note: Webhooks may not arrive immediately.

Webhooks are sent out with the same priority and order as email triggers. If webhook dispatch fails (*e.g.,* because the receiving server is misconfigured), Zammad will retry up to four times.

Hint: Webhooks are available for *Triggers* and *Scheduler*.

14.1 How does it work?

Under the hood, Zammad sends a POST request to a third-party URL ("API endpoint") you specify in the New Trigger dialog. The application server behind this URL/endpoint must be configured to receive messages from Zammad and handle the attached data accordingly.

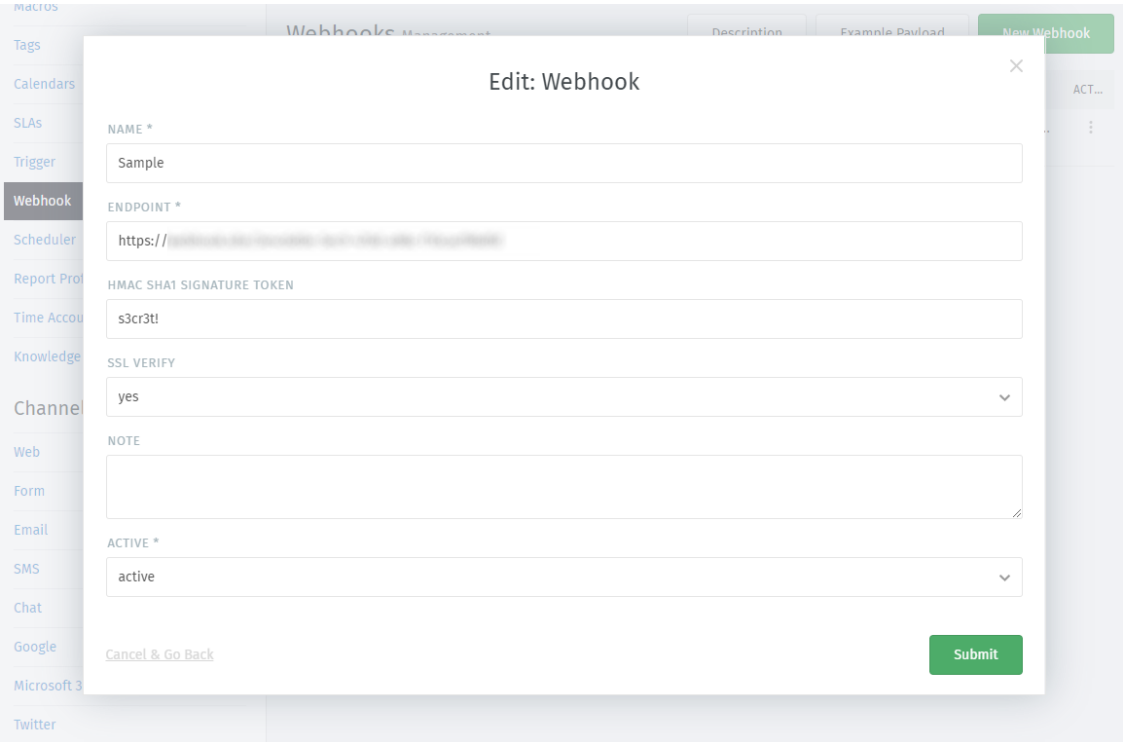
Webhook requests from Zammad contain the following JSON data about new/incoming tickets:

- ticket attributes/metadata
- *all* associated articles
- associated users (*e.g.,* article senders, owners, etc.)
- associated user roles
- associated user organizations (if applicable)
- associated groups

14.1.1 Adding webhooks

Webhooks are defined globally. This allows you to use one specific endpoint on several triggers or schedulers.

> **Warning: Zammad webhooks are specific**
>
> Keep in mind that the remote site has to be able to understand the webhook Zammad is sending! Simply throwing Zammad payloads at webhook endpoint may not have the desired result!



You can configure the following information to webhooks:

Name (mandatory) This name will be displayed within trigger and scheduler selections.

Endpoint (mandatory) Webhook endpoint Zammad sends its payload to.

> **Note:** Zammad ignores basic authentication parameters.

HMAC SHA1 Signature Token If set all sent webhooks contain a *x-hub-signature* header allowing the remote site to verify the request.

> **Note: Security note**
>
> This *does not* encrypt the payload. Use HTTPs connections to secure the communication. It contains a HMAC signature of the body of the webhook request
>
> Learn more about HUB-Signatures

SSL verify Defaults to `yes` - if you're using unsecure self signed certificates set this option to `no`.

Note If required you can leave useful information for other Zammad admins to understand the webhook in question better.

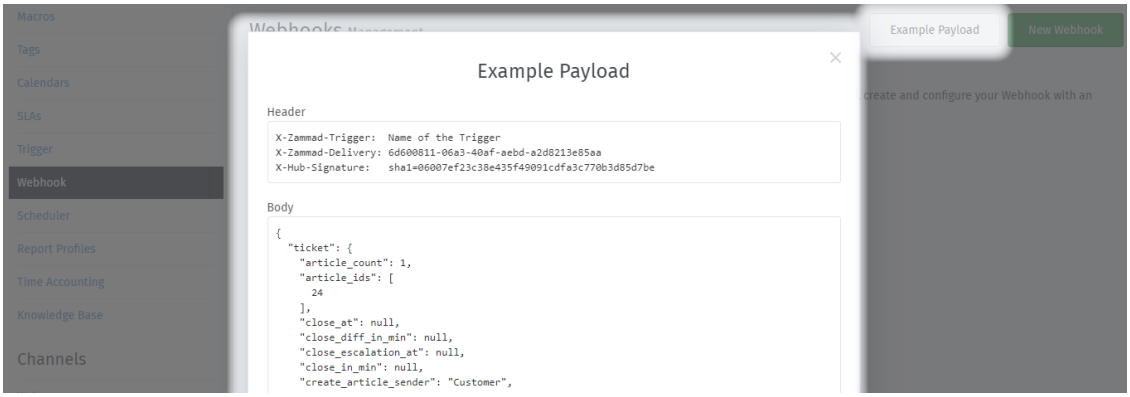
Active If set to `inactive` you can no longer select the webhook within trigger or scheduler actions.

> **Warning:** Setting webhooks to inactive that are used by triggers or schedulers will not run. If triggers or schedulers have other actions configured as well they'll still be executed.

14.1.2 Webhook Payload

Tip: A more personal payload…

Your Zammad instance also provides a payload as example. This payload *does fit your installation* and provides your custom objects!



Request headers

Zammad sends the following headers in each webhook POST request:

`User-Agent` `"Zammad User Agent"`

`X-Zammad-Trigger` The name of the originating trigger

`X-Zammad-Delivery` A unique, random ID string

`X-Hub-Signature` The SHA-1 hash of your HMAC-SHA1 signature token (assuming you provided one when creating your trigger)

JSON payload (example)


```
{
  "ticket": {
    "article_count": 1,
    "article_ids": [
      104
    ],
    "create_article_sender": "Customer",
    "create_article_sender_id": 2,
    "create_article_type": "phone",
    "create_article_type_id": 5,
    "created_at": "2020-11-13T14:34:35.282Z",
    "created_by": {
      "active": true,
      "created_at": "2020-11-13T12:57:47.679Z",
      "created_by": "-",
      "created_by_id": 1,
      "email": "chris@chrispresso.com",
      "firstname": "Christopher",
      "id": 3,
      "image": "7a6a0d1d94ad2037153cf3a6c1b49a53",
      "lastname": "Miller",
      "login": "chris@chrispresso.com",
      "organization": "Chrispresso Inc.",
      "organization_id": 2,
      "out_of_office": false,
      "role_ids": [
        1,
        2
      ],
      "roles": [
        "Admin",
        "Agent"
      ],
      "updated_at": "2020-11-13T13:00:03.064Z",
      "updated_by": "chris@chrispresso.com",
      "updated_by_id": 3,
      "verified": false,
      "vip": false,
    },
    "created_by_id": 3,
    "customer": {
      "active": true,
      "address": "Bennelong Point\nSydney NSW 2000",
      "created_at": "2020-11-13T12:57:48.779Z",
      "created_by": "-",
      "created_by_id": 1,
      "email": "emily@example.com",
      "firstname": "Emily",
      "id": 8,
      "image": "99ba64a89f7783c099c304c9b00ff9e8",
      "lastname": "Adams",
```


(continues on next page)

(continued from previous page)


```
    "login": "emily@example.com",
    "note": "did order café au lait, ask next time if the flavor was as expected",
    "organization": "Awesome Customer Inc.",
    "organization_id": 3,
    "out_of_office": false,
    "phone": "0061 2 1234 7777",
    "role_ids": [
      3
    ],
    "roles": [
      "Customer"
    ],
    "updated_at": "2020-11-13T14:34:37.366Z",
    "updated_by": "chris@chrispresso.com",
    "updated_by_id": 3,
    "verified": false,
    "vip": false,
  },
  "customer_id": 8,
  "group": {
    "active": true,
    "created_at": "2020-11-13T12:57:47.498Z",
    "created_by": "-",
    "created_by_id": 1,
    "follow_up_assignment": true,
    "follow_up_possible": "yes",
    "id": 3,
    "name": "Service Desk",
    "updated_at": "2020-11-13T12:57:48.044Z",
    "updated_by": "-",
    "updated_by_id": 1,
    "user_ids": [
      3,
      4,
      5
    ],
    "users": [
      "chris@chrispresso.com",
      "jacob@chrispresso.com",
      "emma@chrispresso.com"
    ]
  },
  "group_id": 3,
  "id": 81,
  "last_contact_at": "2020-11-13T14:34:35.318Z",
  "last_contact_customer_at": "2020-11-13T14:34:35.318Z",
  "number": "10081",
  "organization": {
    "active": true,
    "created_at": "2020-11-13T12:57:47.524Z",
    "created_by": "-",
    "created_by_id": 1,
```


(continues on next page)

(continued from previous page)


```
      "domain_assignment": false,
      "id": 3,
      "member_ids": [
        8,
        6,
        7
      ],
      "members": [
        "emily@example.com",
        "anna@example.com",
        "samuel@example.com"
      ],
      "name": "Awesome Customer Inc.",
      "note": "Global distributor of communication and security products, electrical and␣
↪electronic wire &amp;amp; cable.",
      "shared": true,
      "updated_at": "2020-11-13T14:34:35.346Z",
      "updated_by": "-",
      "updated_by_id": 1
    },
    "organization_id": 3,
    "owner": {
      "active": true,
      "created_at": "2020-11-13T12:57:48.036Z",
      "created_by": "-",
      "created_by_id": 1,
      "email": "emma@chrispresso.com",
      "firstname": "Emma",
      "id": 5,
      "image": "b64fef91c29105b4a08a2a69be08eda3",
      "lastname": "Taylor",
      "login": "emma@chrispresso.com",
      "organization": "Chrispresso Inc.",
      "organization_id": 2,
      "out_of_office": false,
      "role_ids": [
        2
      ],
      "roles": [
        "Agent"
      ],
      "updated_at": "2020-11-13T12:57:48.072Z",
      "updated_by": "-",
      "updated_by_id": 1,
      "verified": false,
      "vip": false,
    },
    "owner_id": 5,
    "priority": {
      "active": true,
      "created_at": "2020-11-13T12:54:02.238Z",
      "created_by": "-",
```


(continues on next page)

(continued from previous page)


```
      "created_by_id": 1,
      "default_create": true,
      "id": 2,
      "name": "2 normal",
      "updated_at": "2020-11-13T12:54:02.238Z",
      "updated_by": "-",
      "updated_by_id": 1
    },
    "priority_id": 2,
    "state": "open",
    "state_id": 2,
    "ticket_time_accounting": [],
    "ticket_time_accounting_ids": [],
    "title": "Webhook-Test",
    "updated_at": "2020-11-13T14:34:35.333Z",
    "updated_by": {
      "active": true,
      "created_at": "2020-11-13T12:57:47.679Z",
      "created_by": "-",
      "created_by_id": 1,
      "email": "chris@chrispresso.com",
      "firstname": "Christopher",
      "id": 3,
      "image": "7a6a0d1d94ad2037153cf3a6c1b49a53",
      "lastname": "Miller",
      "login": "chris@chrispresso.com",
      "organization": "Chrispresso Inc.",
      "organization_id": 2,
      "out_of_office": false,
      "role_ids": [
        1,
        2
      ],
      "roles": [
        "Admin",
        "Agent"
      ],
      "updated_at": "2020-11-13T13:00:03.064Z",
      "updated_by": "chris@chrispresso.com",
      "updated_by_id": 3,
      "verified": false,
      "vip": false,
    },
    "updated_by_id": 3
  },
  "article": {
    "attachments": [
      {
        "id": 174,
        "filename": "image1.jpeg",
        "size": "35574",
        "preferences": {
```


(continues on next page)

(continued from previous page)


```
          "Content-Type": "image/jpeg",
          "Mime-Type": "image/jpeg",
          "Content-ID": "81.969520479@zammad.example.com",
          "Content-Disposition": "inline",
          "resizable": true,
          "content_preview": true
        },
        "url": "https://zammad.example.com/api/v1/ticket_attachment/81/104/174"
      }
    ],
    "body": "This is a simple Webhook Test.<div><br></div><div>\n<img style=\"max-
↪width:100%;width: 849px;max-width: 100%;\" src=\"/api/v1/ticket_attachment/81/104/174?
↪view=inline\"><br>\n</div>",
    "content_type": "text/html",
    "created_at": "2020-11-13T14:34:35.318Z",
    "created_by": {
      "active": true,
      "created_at": "2020-11-13T12:57:47.679Z",
      "created_by": "-",
      "created_by_id": 1,
      "email": "chris@chrispresso.com",
      "firstname": "Christopher",
      "id": 3,
      "image": "7a6a0d1d94ad2037153cf3a6c1b49a53",
      "lastname": "Miller",
      "login": "chris@chrispresso.com",
      "organization": "Chrispresso Inc.",
      "organization_id": 2,
      "out_of_office": false,
      "role_ids": [
        1,
        2
      ],
      "roles": [
        "Admin",
        "Agent"
      ],
      "updated_at": "2020-11-13T13:00:03.064Z",
      "updated_by": "chris@chrispresso.com",
      "updated_by_id": 3,
      "verified": false,
      "vip": false,
    },
    "created_by_id": 3,
    "from": "Emily Adams <emily@example.com>",
    "id": 104,
    "internal": false,
    "origin_by": "emily@example.com",
    "origin_by_id": 8,
    "sender": "Customer",
    "sender_id": 2,
    "ticket_id": 81,
```


(continues on next page)

(continued from previous page)


```
    "to": "Service Desk",
    "type": "phone",
    "type_id": 5,
    "updated_at": "2020-11-13T14:34:35.318Z",
    "updated_by": {
      "active": true,
      "created_at": "2020-11-13T12:57:47.679Z",
      "created_by": "-",
      "created_by_id": 1,
      "email": "chris@chrispresso.com",
      "firstname": "Christopher",
      "id": 3,
      "image": "7a6a0d1d94ad2037153cf3a6c1b49a53",
      "karma_user_ids": [],
      "lastname": "Miller",
      "login": "chris@chrispresso.com",
      "organization": "Chrispresso Inc.",
      "organization_id": 2,
      "out_of_office": false,
      "role_ids": [
        1,
        2
      ],
      "roles": [
        "Admin",
        "Agent"
      ],
      "updated_at": "2020-11-13T13:00:03.064Z",
      "updated_by": "chris@chrispresso.com",
      "updated_by_id": 3,
      "verified": false,
      "vip": false,
    },
    "updated_by_id": 3,
    "accounted_time": 0
  }
}
```

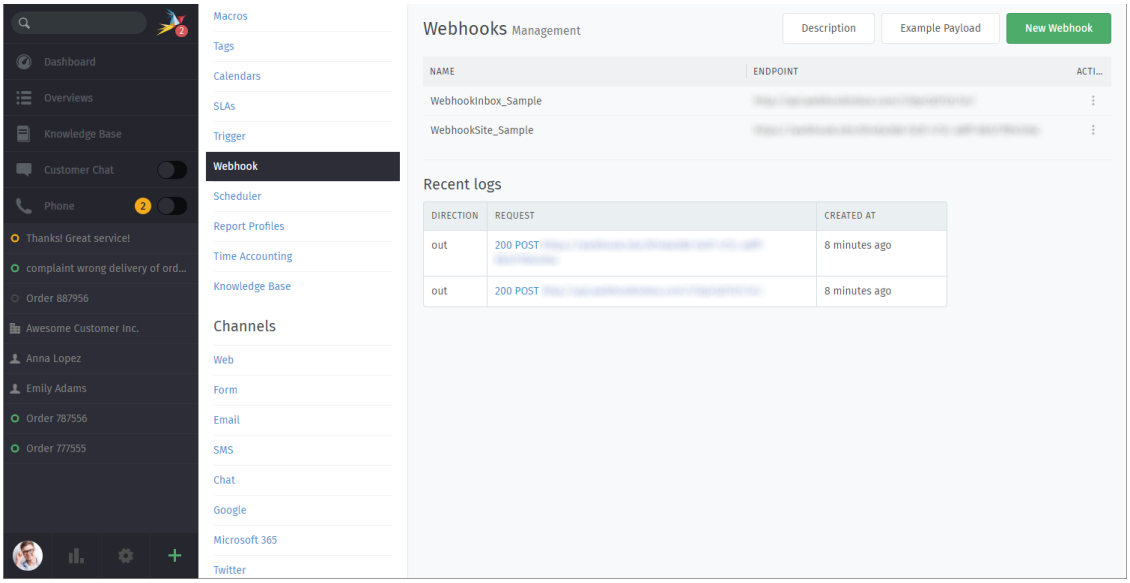

Note:

- For better readability, all empty and `null` values have been omitted from the sample payload above. That means the webhooks you receive *will include additional fields not shown here*.
- Webhooks will also include fields for any relevant *custom objects* defined in your system.
- Attachments are *not included*; links to attachments *are* (authentication required).
- None of the following **user attributes** are included:
  - `last_login`
  - `login_failed`
  - `password`
  - `preferences`

- `group_ids`
- `groups`
- `authorization_ids`
- `authorizations`

14.1.3 Webhook Logs

Zammad provides a history of your recent webhooks. You can find them below **Recent logs**.



If you need more details you can click on the request link in question. Zammad will provide a modal with the following information:

Direction Always `out`.

URL The URL Zammad sent the request to.

Method Always `POST`.

Status Contains the HTTP status code the remote server replied with. Should be `2xx` if successful.

Request Contains the request Zammad sent (HTTP header and payload)

Response Contains the remotes response header.

Created at Date and time the request was sent.

CHAPTER

FIFTEEN

SCHEDULER

The scheduler performs time-based automated actions. You can set up your own schedulers, configure at which points in time they should run, set up conditions to determine which tickets they should affect, and then configure the actions that you want to be executed on these tickets.

Note: Schedulers with *Action: Delete* are currently the only way in the Zammad front end to permanently delete tickets. This limitation is intentional as Zammad is designed to be revision-proof. A possible use case for such a scheduler is to delete spam tickets some time after creation (e.g. 30 days).

Warning: While it is possible to delegate scheduler permissions to normal agents with the `admin/scheduler` permission, it is inadvisable to do so. Malicious agents could use a scheduler to access tickets in restricted groups (by moving them to a non-restricted group) or to delete arbitrary tickets.

Hint:

Schedulers can be used to send periodic reminder emails. Use *Variables* to build highly-customized email templates.

Schedulers only perform `2000` tickets per run. This is a security function in case you accidentally miss configure the scheduler.

15.1 Add a new scheduler

Name Choose a name for the scheduler.

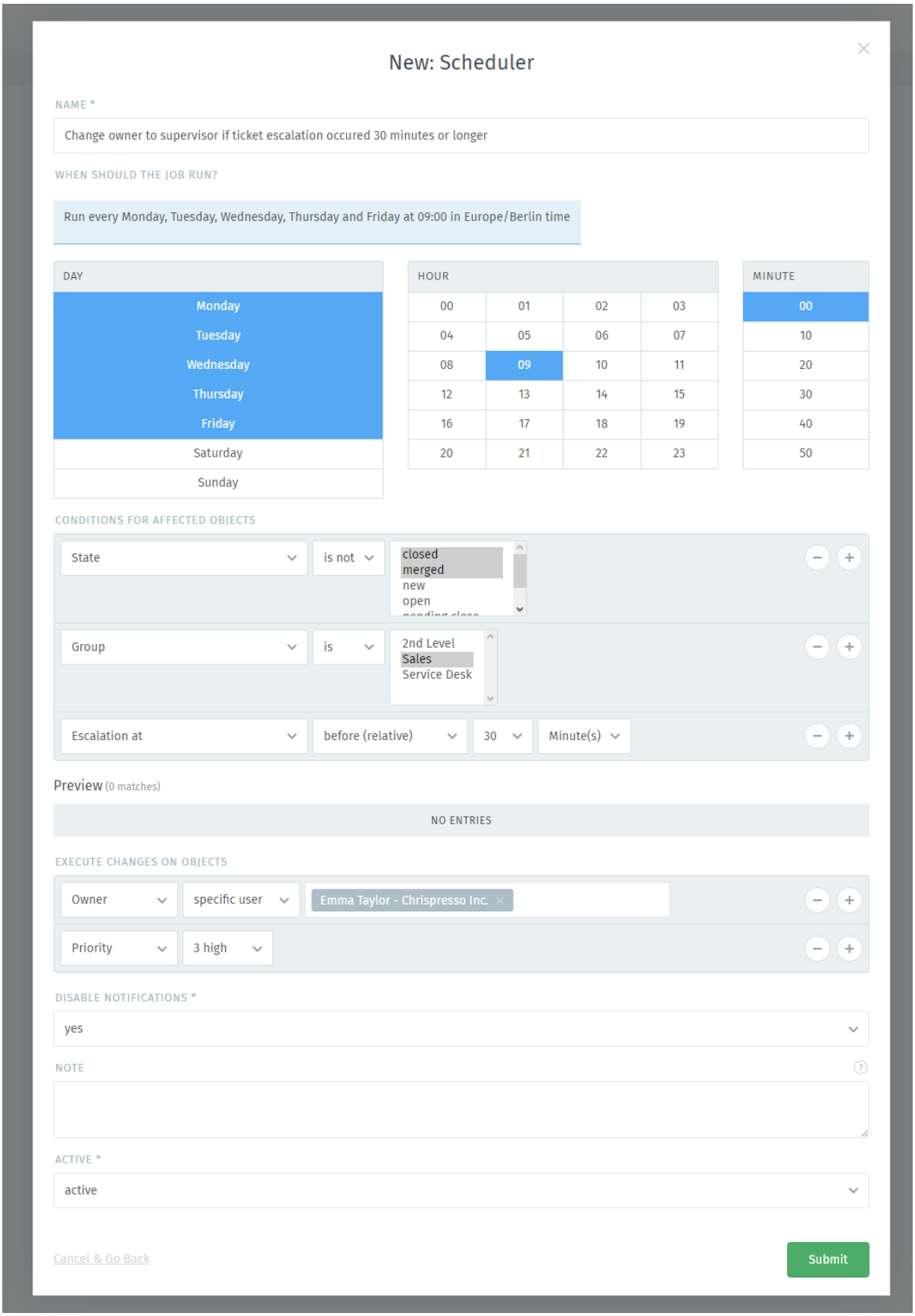
When should the job run? Choose the points in time using *Zammad's timezone* when the scheduler should run.

> **Note:** The scheduler tasks are not saving any timezone information. Thus: Scheduler tasks created prior Zammad 5.1 don't require any change.

Conditions for affected objects Determine the ticket attributes (conditions) to limit on which tickets the actions configured in step 5 are to be performed.

Preview This list previews some tickets that your conditions are matching and shows a total of how many tickets are being matched. Use this to double-check the entered conditions.

Execute changes on objects Determine the changes to be made to the ticket.


New: Scheduler
NAME *
Change owner to supervisor if ticket escalation occured 30 minutes or longer
WHEN SHOULD THE JOB RUN?
Run every Monday, Tuesday, Wednesday, Thursday and Friday at 09:00 in Europe/Berlin time
DAY
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
HOUR
00 01 02 03
04 05 06 07
08 09 10 11
12 13 14 15
16 17 18 19
20 21 22 23
MINUTE
00
10
20
30
40
50
CONDITIONS FOR AFFECTED OBJECTS
State
is not
closed
merged
new
open
Group
is
2nd Level
Sales
Service Desk
Escalation at
before (relative)
30
Minute(s)
Preview (0 matches)
NO ENTRIES
EXECUTE CHANGES ON OBJECTS
Owner
specific user
Emma Taylor - Chrispresso Inc.
Priority
3 high
DISABLE NOTIFICATIONS *
yes
NOTE
ACTIVE *
active
Cancel & Go Back
Submit

Disable notifications By default, actions triggered by schedulers won't send notifications. You can override this here by setting this to *no*.

Note You can use the note field to describe the purpose of the scheduler. This is only visible to other admins when they are editing the scheduler. It is *not* a way to add notes to tickets.

Active With this setting you can enable/disable the scheduler.

The scheduler shown in the screenshot would have the following effects:

Every workday (Monday to Friday) at 9:00 a.m. (*Europe / Berlin UTC+1*), all tickets which:

- are not closed or merged, and
- are assigned to the Sales group, and
- whose escalation was 30 minutes ago

will be:

- assigned to Emma, and
- have their priority changed to 3 high.

As a supervisor in the Sales group, this enables Emma to intercept and process escalated tickets.

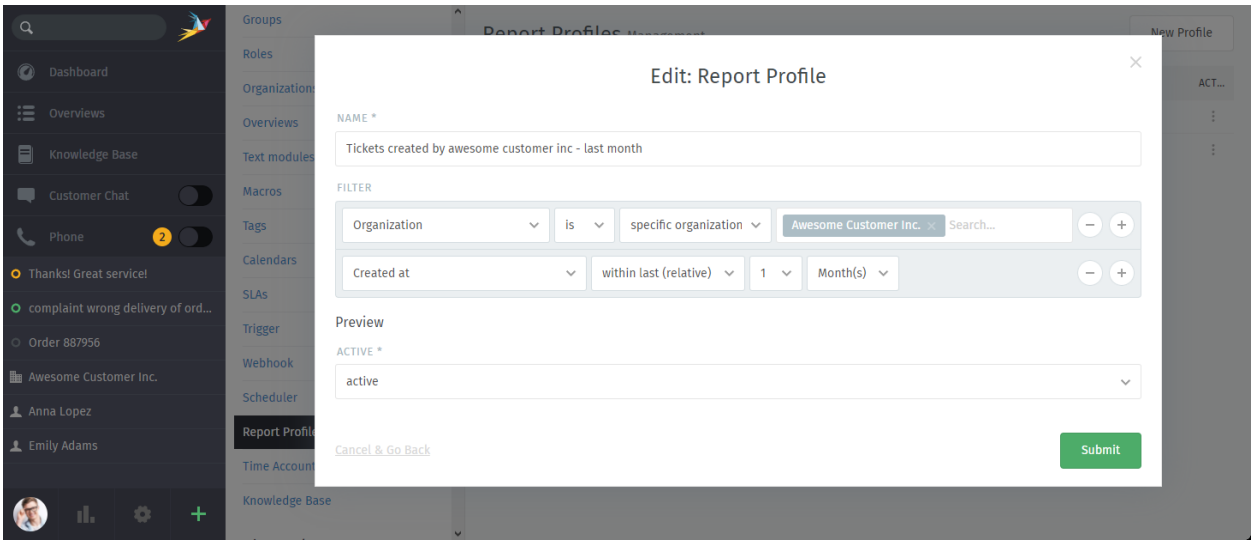
Emma will not receive notifications when the scheduler assigns her these tickets, and no note will be added to them.

CHAPTER

SIXTEEN

REPORT PROFILES

Report profiles are used to restrict / filter report-results. The idea of the profiles is to limit the number of tickets and determine the type of tickets you want to analyze. You can create any number of profiles in the Admin Interface in the "Report Profile" area. The edit-mask looks like this:



This example shows the statistics of all tickets of the organization "Awesome Customer" that were created in the last month.

The filters can be combined with each other as desired. The filters build on each other, which means that they are further restricted per additional attribute.

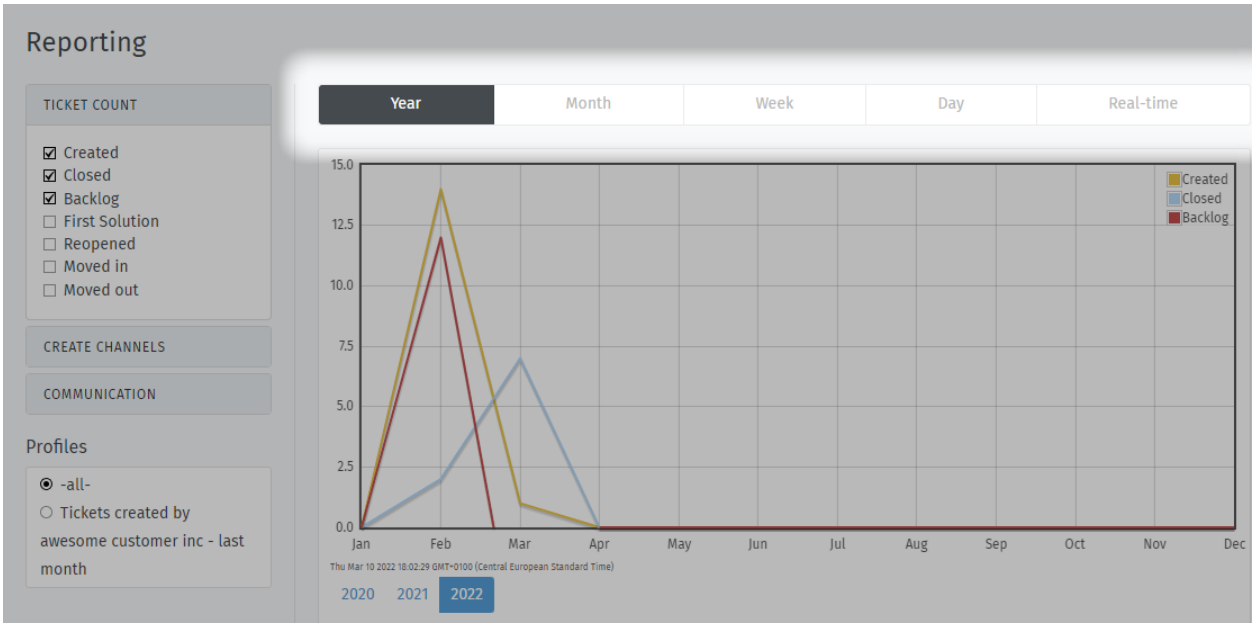
All configured filters are displayed in the statistics area and you can switch between them with one click:

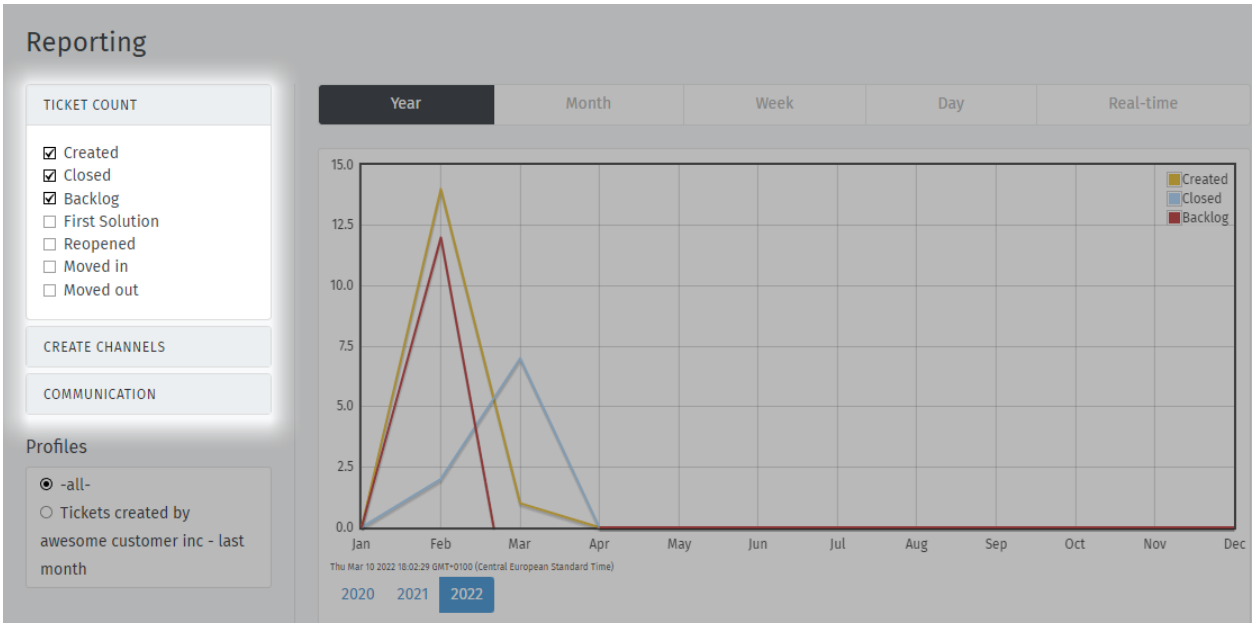
16.1 Further information about the reporting:

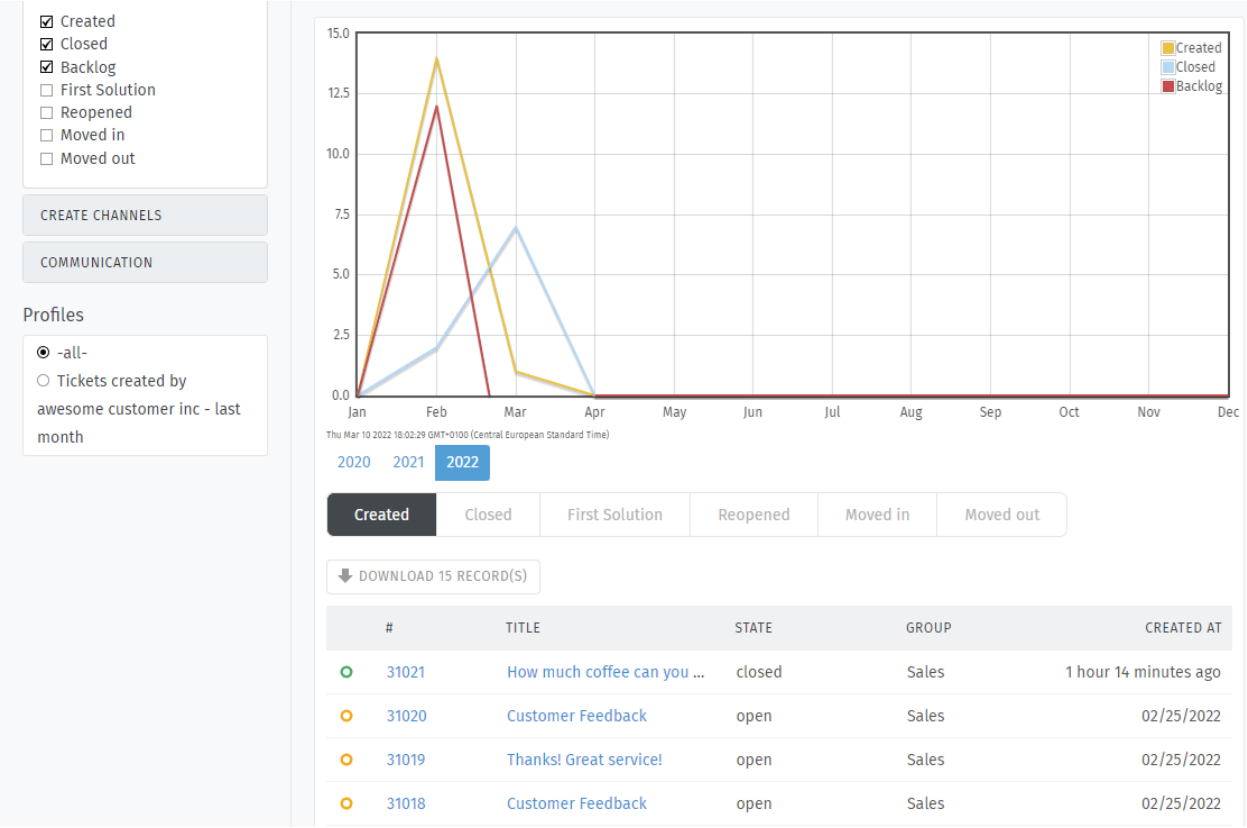
The time period and time interval can be changed with one click (the graphic adapts itself directly):

You can filter the Create Channels (Phone, email, Twitter,...) and Communication (Phone, email, Twitter,...) and select metrics (choose from the menu bar in the upper left corner - currently Ticket Count, Create Channels, Communication). Here are all tickets with a certain metric of a profile evaluated.

The list below the graphic shows the tickets that have been filtered out. This can be downloaded as CSV and processed in a statistics program:


Reporting
TICKET COUNT
Created
Closed
Backlog
First Solution
Reopened
Moved in
Moved out
CREATE CHANNELS
COMMUNICATION
Profiles
-all-
Tickets created by
awesome customer inc - last
month
Year
Month
Week
Day
Real-time
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Created
Closed
Backlog
Thu Mar 10 2022 18:02:29 GMT+0100 (Central European Standard Time)
2020
2021
2022


Reporting
TICKET COUNT
Created
Closed
Backlog
First Solution
Reopened
Moved in
Moved out
CREATE CHANNELS
COMMUNICATION
Profiles
-all-
Tickets created by
awesome customer inc - last
month
Year
Month
Week
Day
Real-time
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Created
Closed
Backlog
Thu Mar 10 2022 18:02:29 GMT+0100 (Central European Standard Time)
2020
2021
2022

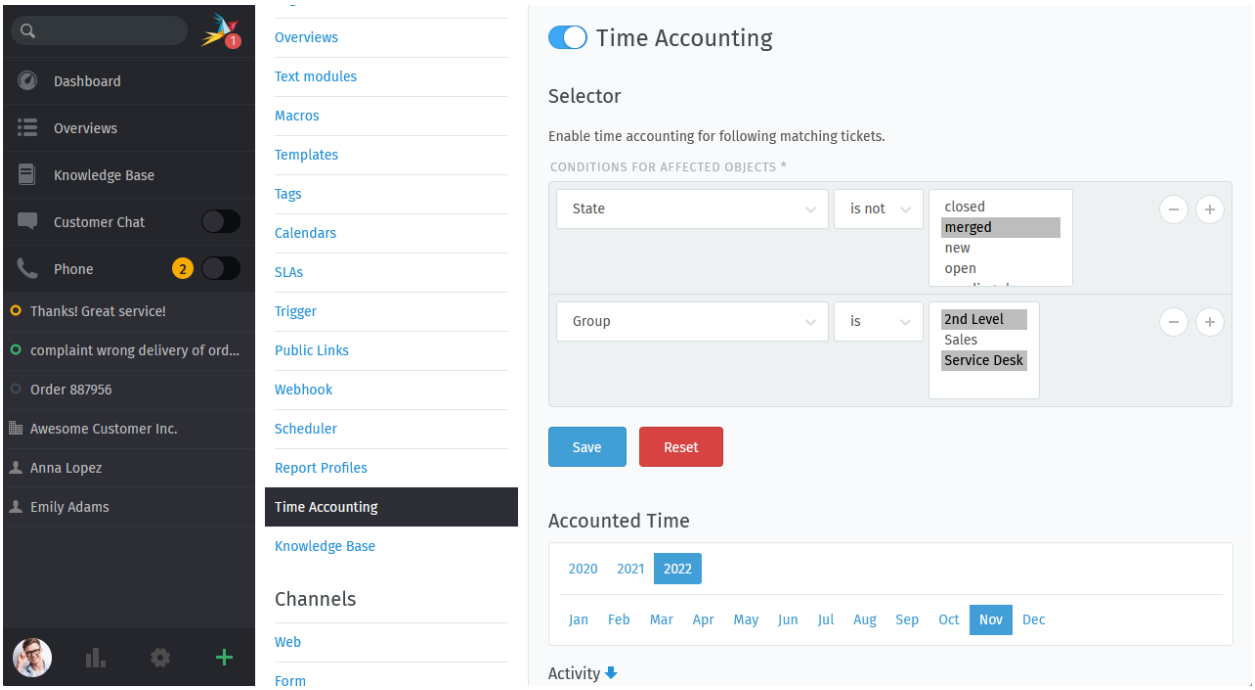

Created
Closed
Backlog
First Solution
Reopened
Moved in
Moved out
CREATE CHANNELS
COMMUNICATION
Profiles
-all-
Tickets created by awesome customer inc - last month
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Thu Mar 10 2022 18:02:29 GMT+0100 (Central European Standard Time)
2020
2021
2022
Created
Closed
First Solution
Reopened
Moved in
Moved out
DOWNLOAD 15 RECORD(S)
TITLE STATE GROUP CREATED AT
31021 How much coffee can you ... closed Sales 1 hour 14 minutes ago
31020 Customer Feedback open Sales 02/25/2022
31019 Thanks! Great service! open Sales 02/25/2022
31018 Customer Feedback open Sales 02/25/2022

CHAPTER

SEVENTEEN

TIME ACCOUNTING

If you want to know how much time you need for your each project or ticket, enable time accounting (turn on the switch on the top left side of the page).



17.1 How it works

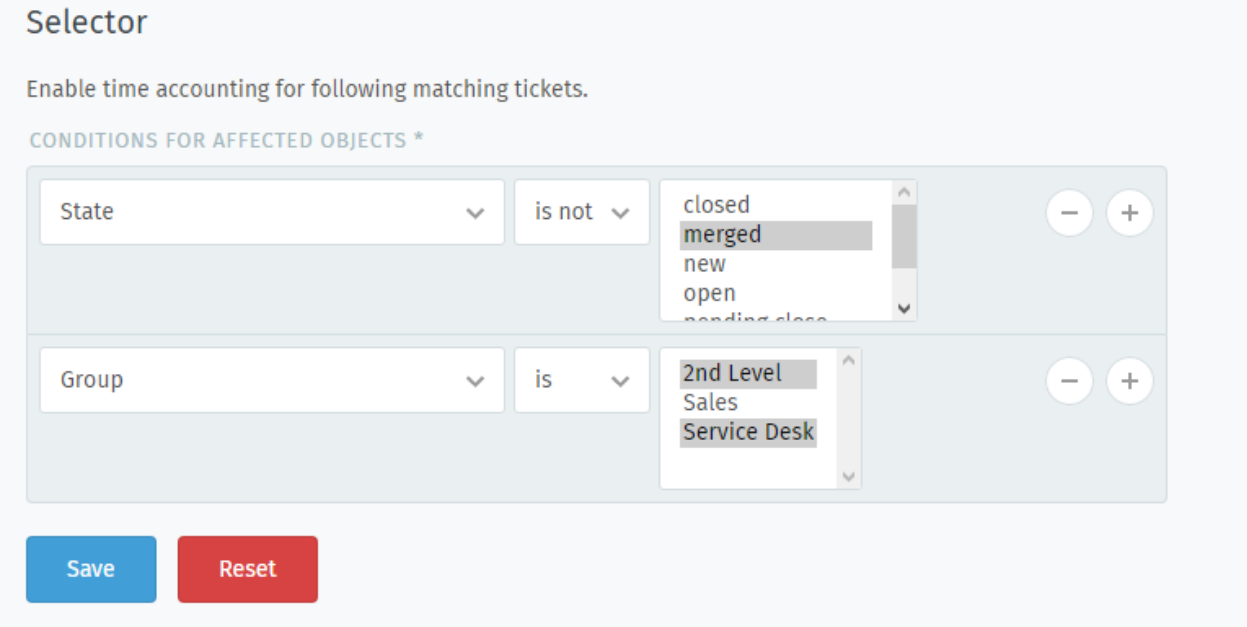
Zammad's time accounting uses ticket selectors (filters) to check if a ticket is applicable for time accounting or not. If a ticket applies for accounting, Zammad will request the agent to provide how much time was needed to process the current ticket step.

Note: In order for Zammad to bring up the time accounting dialogue to an agent, the agent has to update the ticket together with any article type. The article part is mandatory.

However, the time accounting dialogue is not mandatory and can be cancelled by your agents if needed. You cannot enforce time accounting.

If a ticket no longer applies for time accounting, the already accounted time is not lost.

Tip: The selector applies to the ticket state before any attribute changes have been saved. That means if your agent is e.g. going to close a ticket alongside writing an article, the ticket selector has to match the ticket state before closing for the time accounting dialog to appear.



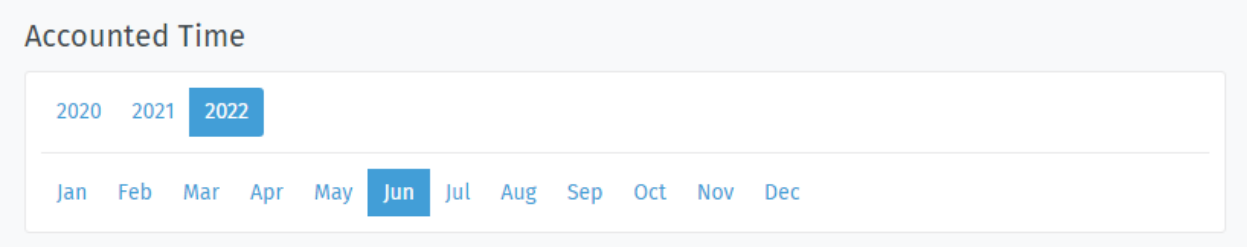
17.2 Reviewing accounted time

Below the selector configuration, Zammad provides an accounted time section which will contain all accounted times for your tickets. Accounted times are displayed per years and months.

Hint:

Having tickets that are overlapping several months?
No problem! Zammad provides *time units* and *time units total* to allow partial billing.

Select the right month Usually you want to bill accounted time of other months than the current one. Just select the relevant year and month to receive the accounted times and ticket information



Tickets and their accounted time Zammad allows you to receive the accounted information just like you need them. For this you currently have four options to view and also download the relevant data as CSV.

To download the CSV data, use the arrow down right next to each heading (e.g. “Ticket”).

Hint: Of course you can also automate this with API calls.

Activity This filter works similar to the ticket filter, with one exception: You’ll find each *individual* time accounting step of your agents. This is what you’d also see in the tickets history before Zammad 5.2.

In this list you’ll see the following ticket information:

- Number
- Title
- Customer
- Organization of customer (if applicable)
- Agent that accounted the time
- Time units accounted in the current month
- Created at

Ticket This filter contains all relevant tickets from the selected month.

In this list you’ll see the following ticket information:

- Number
- Title
- Customer
- Organization of customer (if applicable)
- Agent currently assigned (ticket owner)
- Time units accounted in the current month
- Time units total (time accounted during ticket life)
- Created at
- Closed at (if applicable)

Hint: The CSV file of this filter provides all ticket meta information.

Customer This provides you a per user filter on accounted time units. Each user has a total of time accounted in the current month (over all applicable tickets).

In this list you’ll see the following ticket information:

- Customer
- Organization of customer (if applicable)
- Time units accounted in the current month

Organization This provides a list of all organizations where customers have caused accounted time in that month.

Note: You can also see entries for customers that are part of an primary organization. Users without organization can be found in the Customer filter.

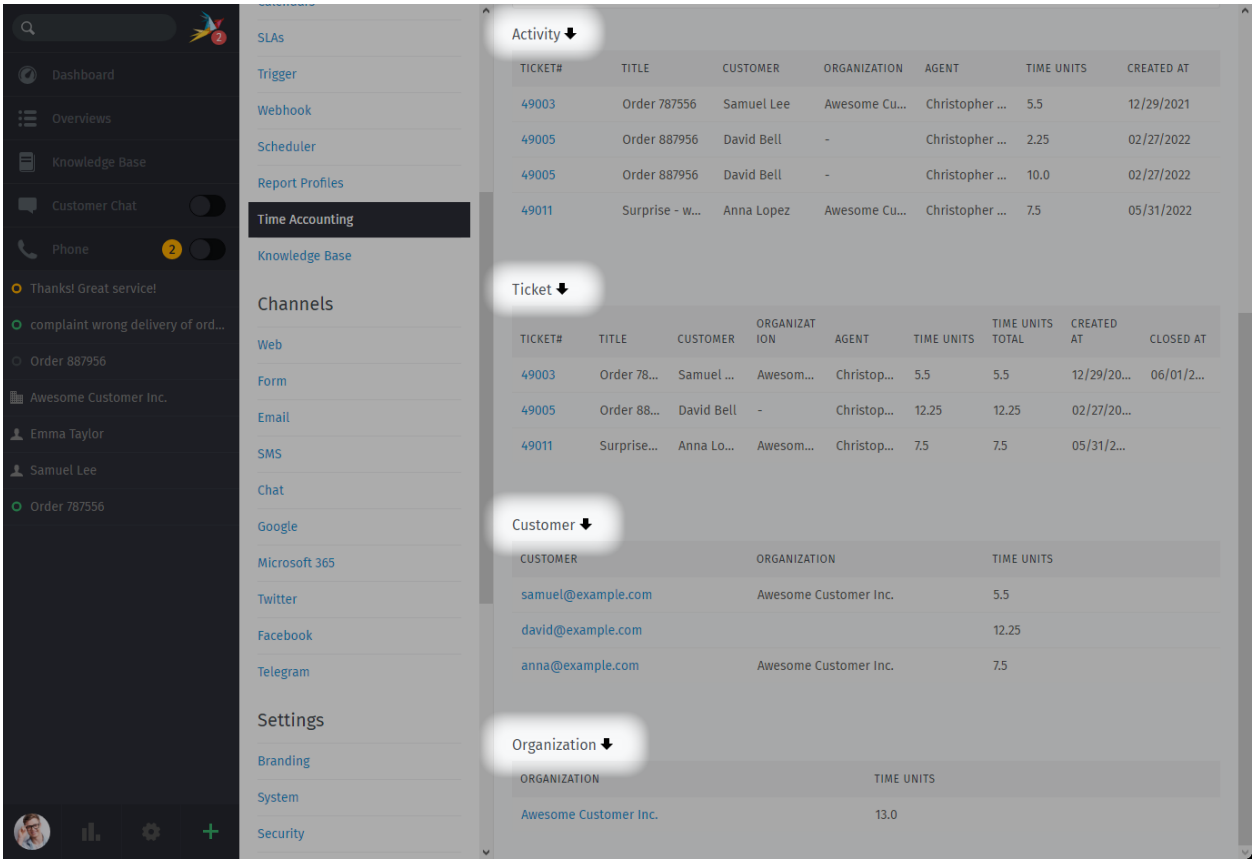


Fig. 1: Each heading allows you to download the CSV versions of the provided view via the downwards arrow.

CHAPTER

EIGHTEEN

KNOWLEDGE BASE

Publish your own library of FAQs, how-tos, internal SOPs and more with the **knowledge base**.

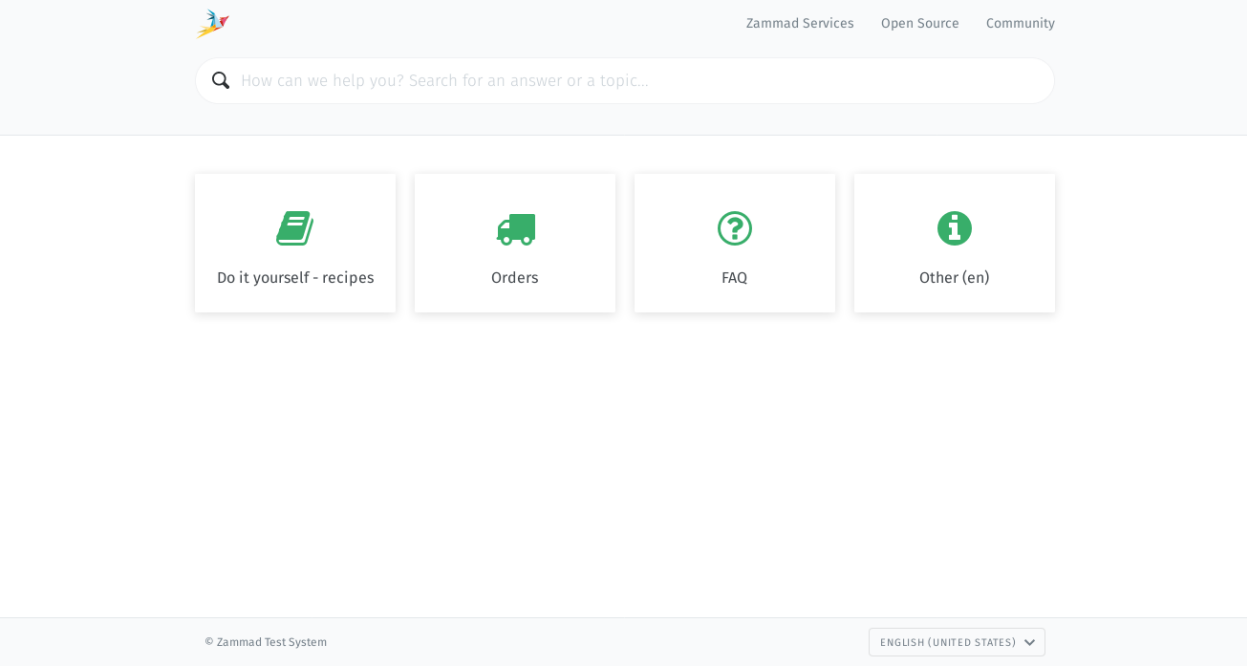


Fig. 1: See a live demo at https://support.zammad.com/help.

Hint: This document describes how to **configure** the knowledge base.

For details on how to **use and edit** it, please refer to the Zammad user documentation.

By default, only admin users are permitted to create, edit, and manage knowledge base articles. See *Roles* for details on how to grant write access to agents or other users.

Note: The knowledge base will not appear in the main menu until it has been enabled in the admin panel.

18.1 Features

- Multi-language support
- Visibility settings (**draft**, **staff-only**, or **public**)
- Full text search
- Scheduled publishing
- File attachments
- Wiki-style internal linking to both **KB answers** and **tickets**
- Rich text editor + embedded images

18.2 Setup

To enable the knowledge base, first select the languages/locales you wish to publish in:

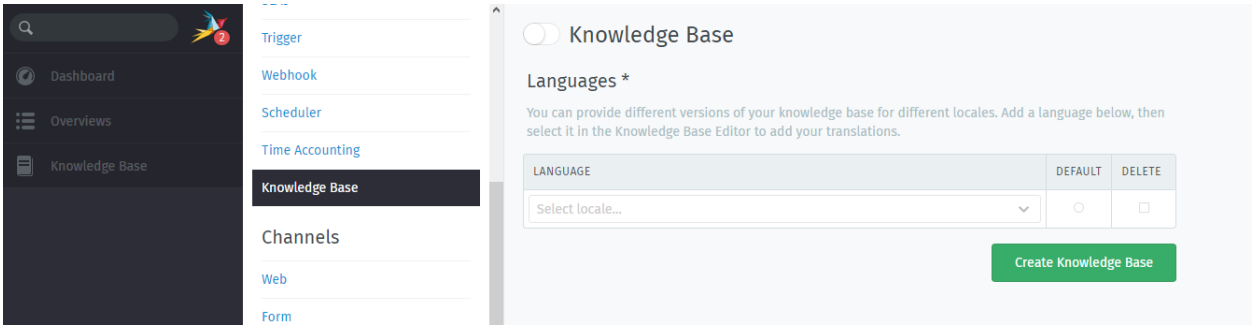


Fig. 2: You must choose at least one. (Don't worry, you can always change them later!)

Read on for details about each section of the knowledge base configuration.

18.2.1 Theme

Icon & Link Color Applies to all **category & article** entries in knowledge base menus, as well as hyperlinks in articles.

Header Color Applies to the area surrounding the search bar.

Header Link Color Defines the color of the header links to use. Make sure that this color has a proper contrast to *Header Color*.

Show Feed Icon You can enable Zammad to provide RSS feed URLs in both internal and public knowledge base. With this option being active, Zammad will provide you up to two RSS links:

- a general RSS feed of the whole knowledge base (top level)
- a category specific RSS feed of the category you're in (also applies to answers you're viewing)

This setting by default is set to `no`.

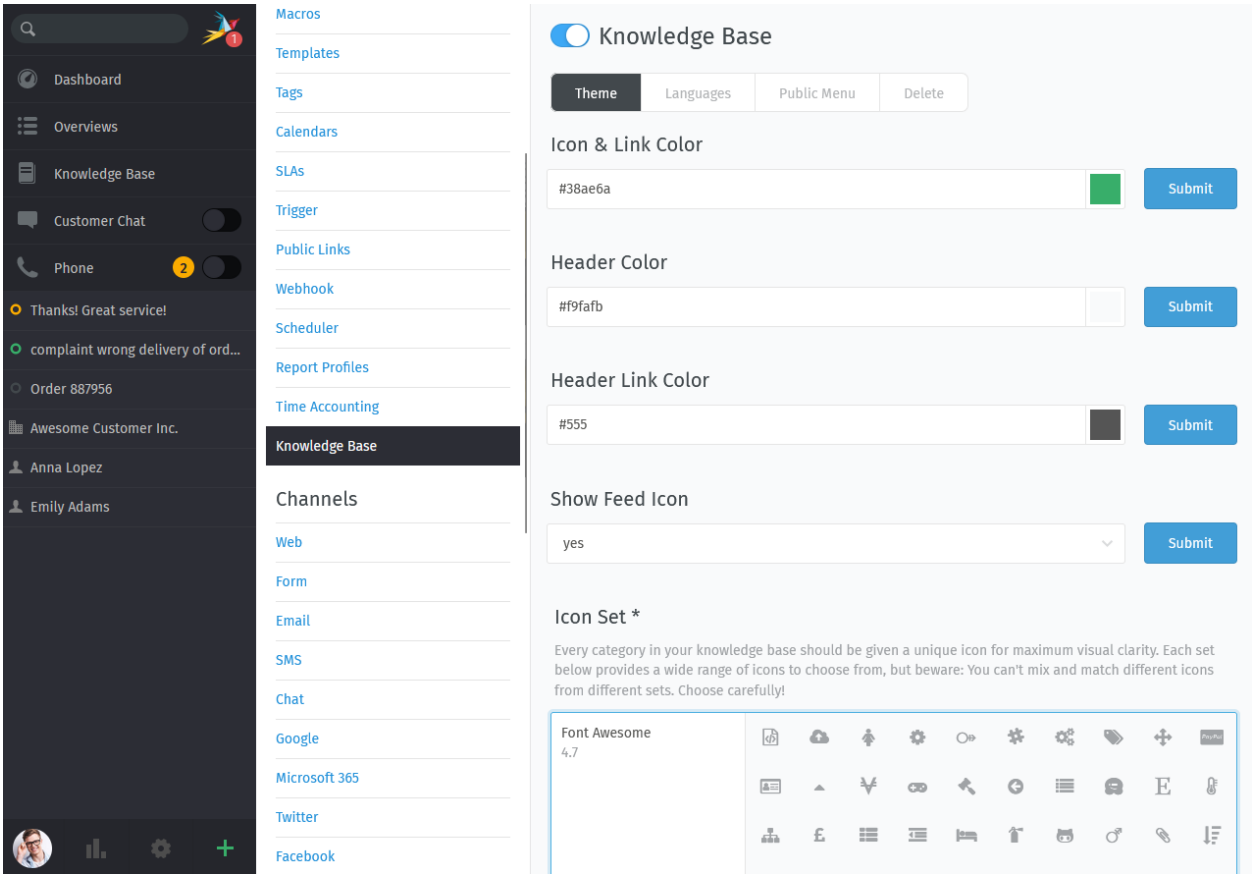
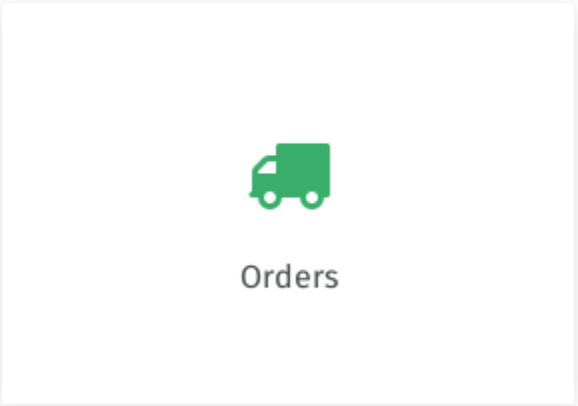


Fig. 3: Customize the appearance of the knowledge base.

Hint: Your agents will receive special RSS feed URLs with **access tokens**. Agents can always renew these. Keep in mind that sharing these URLs with third parties may provide access to **internal** answers!

Icon Set Defines the selection of icons that may be used when creating/editing categories.

Each category in the knowledge base *must* be given an icon. Icons appear prominently in the main menu, like so:



Warning: Re-assigning icons on all of your categories is tedious work. It's advisable to explore your options early to avoid having to change your mind down the road.

18.2.2 Languages

The knowledge base will automatically display the language matching each visitor's locale. Visitors may always manually switch to another language via a dropdown menu in the footer.

The default locale is displayed when the visitor's locale is not supported.

Articles that have not yet been translated into a given language will be hidden from that locale.

18.2.3 Public Menu

Use this section to unify the knowledge base with your own website's main navigation. Entries added here will appear in the knowledge base like so:

Zammad will provide a list of the current set links per knowledge base language. If you're missing a language, you'll have to add the language up front.

You can adjust every URL on language level.

Arranging URLs By clicking on "Edit", Zammad allows you to add, update, re-arrange or remove URLs from either your public header menu or public footer menu.

Title This is the URL title that's being displayed to your users.

URL The actual URL the user is going to open upon clicking.

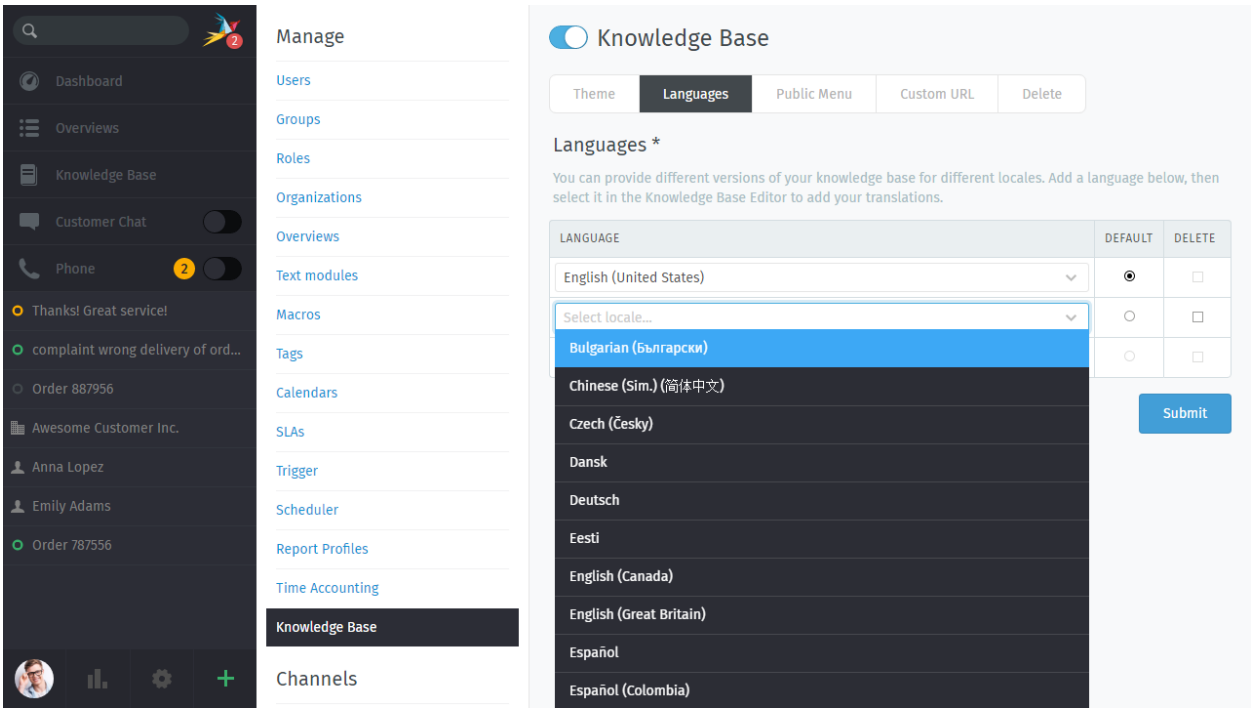
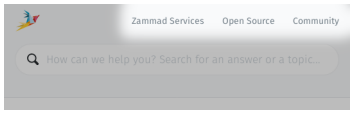


Fig. 4: Add or remove locales, or reassign the default.



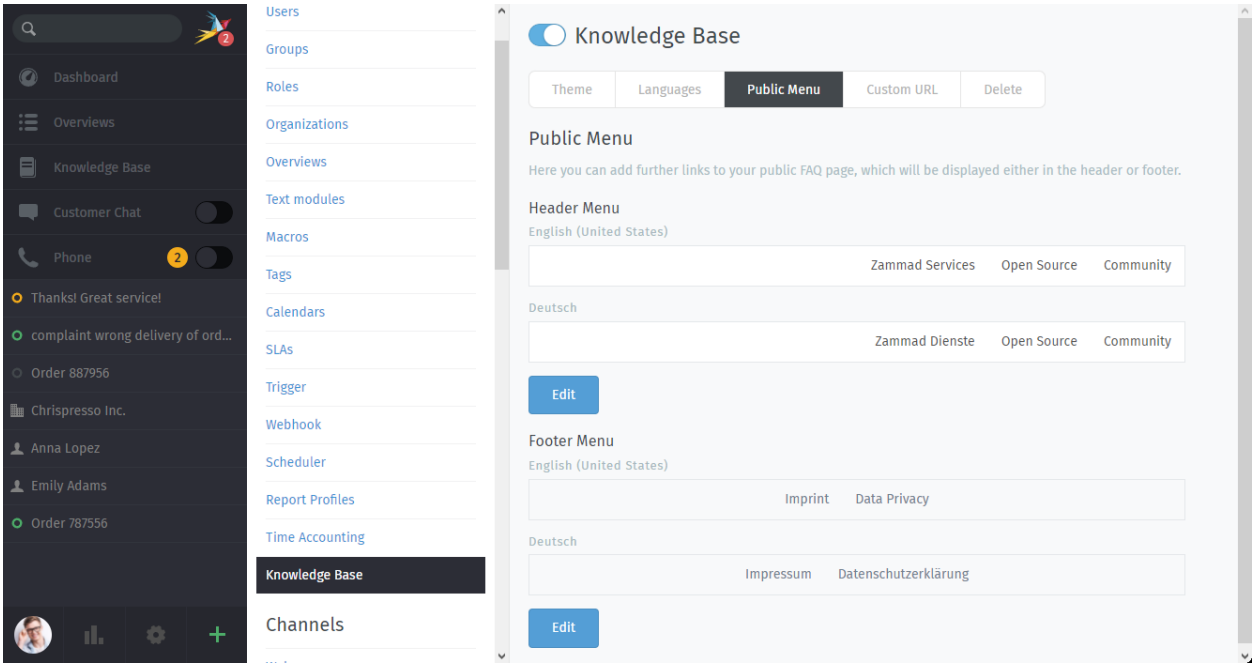


Fig. 5: Customize the nav menu that appears in the header of the knowledge base.

Target Allows you to tell your users browser to open the URL in a new tab. By default your user would leave the knowledge base page if not set.

Delete If you tick the delete field, the URL will be removed from the menu upon pressing on the "Submit" button.

Change URLs position Use to drag & drop the URLs in question to the new desired position. Your changes will be saved with pressing the "Submit" button.

Fig. 6: Re-arranging URL positions works for the header menu just as the footer menu

18.2.4 Custom URL

Note: This feature is only available on **self-hosted** instances.

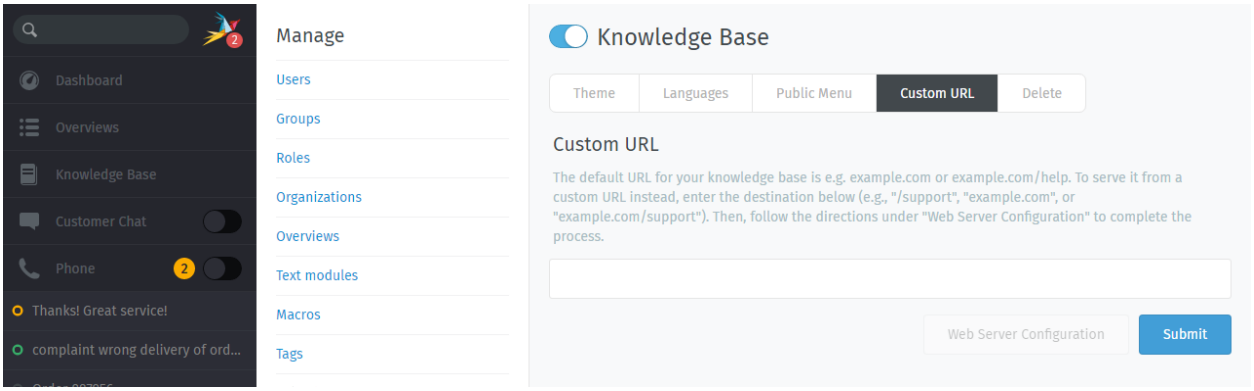


Fig. 7: Relocate the knowledge base to the URL of your choosing.

By default, the knowledge base will appear at the same domain as your Zammad instance, under `/help`.

If you wish to customize where it can be accessed, enter your desired URL here and configure your web server as described (instructions provided for Apache and NGINX only).

18.2.5 Delete

Use this panel to delete the knowledge base. If you wish to unpublish it without deleting all its content, simply disable it via the toggle button at the top of the window instead.

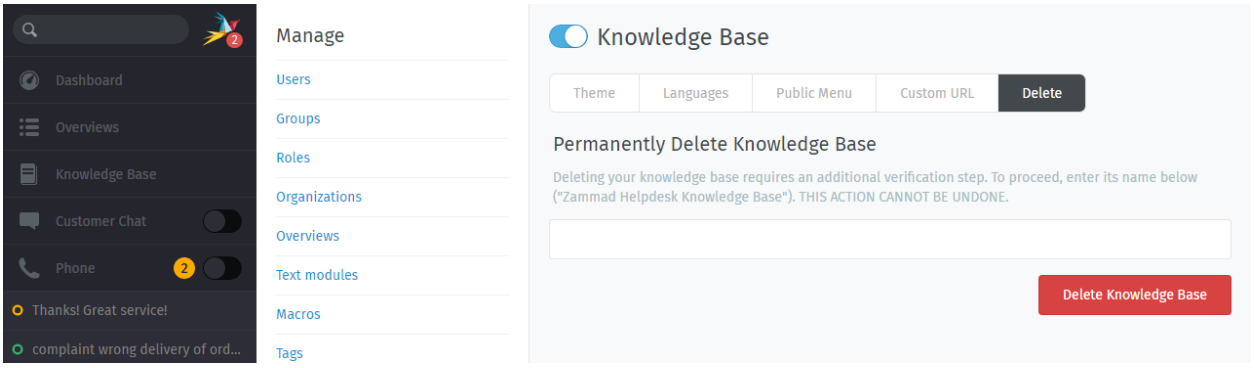


Fig. 8: Permanently delete the knowledge base and all the articles within it.

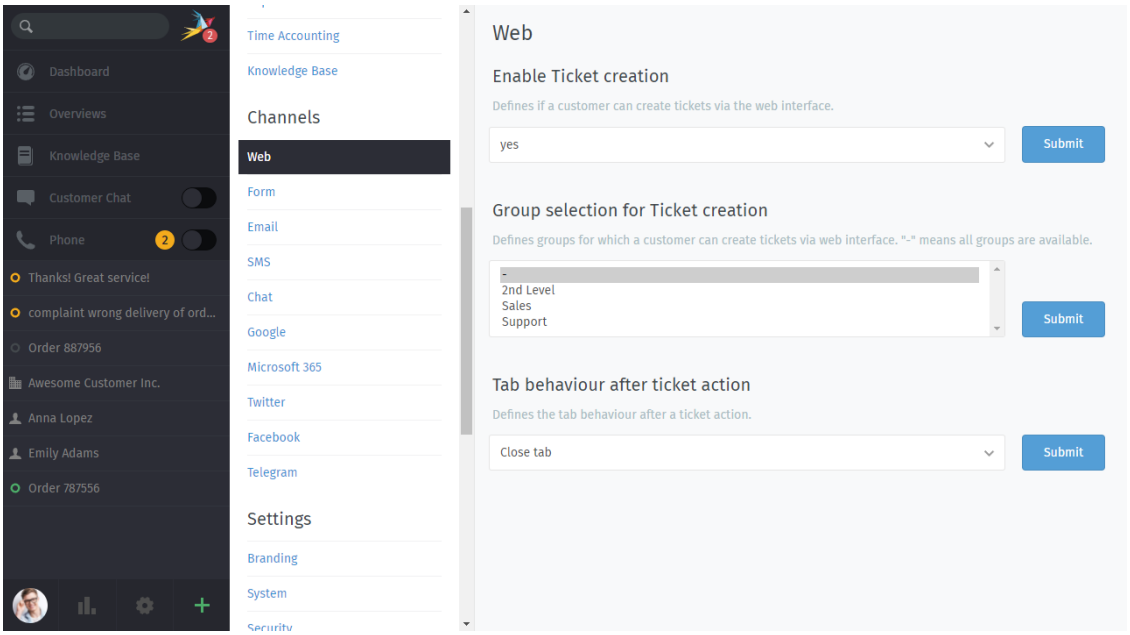
CHAPTER

NINETEEN

WEB

The web channel mainly affects your customers' using the web interface. These settings allow you to restrict your customers ticket creation.

Tip: *Core Workflows* allow even further restrictions and actions if you need them.



19.1 Settings affecting your customer

Enable Ticket creation You can forbid customers to create tickets via the web interface. This will remove the "" button on the lower left.

Default: `yes`

Note: This does not forbid updating existing tickets via UI.

Group selection for Ticket creation By default your customers may create tickets in all groups. As this may be an issue, especially when having several support levels or internal groups, you can always select a set of groups you want to be available.

Default: -

Note: This does not affect your agents

Agents are affected by *Group Access Levels*.

19.2 Global settings affecting *all* users

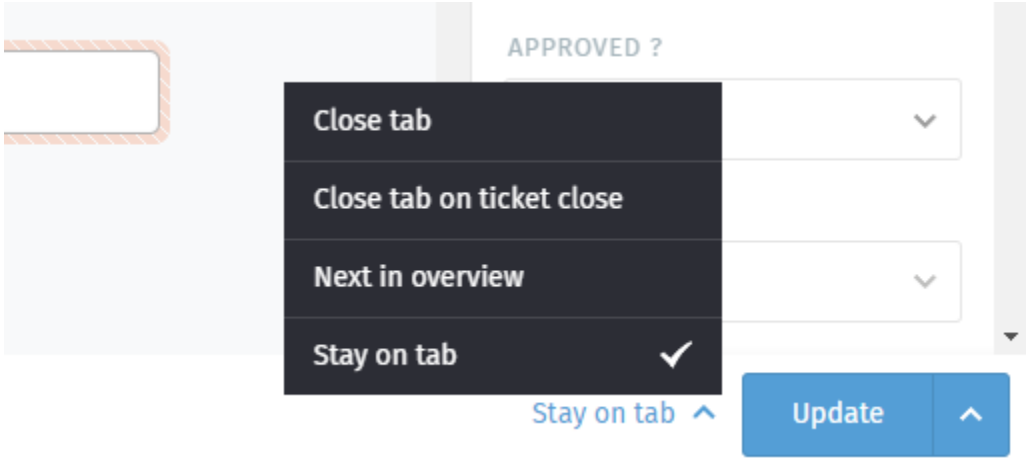
Tab behavior after ticket creation This setting allows administrators to provide a default behavior of Zammad's tab *after* ticket update.

Default: `Stay on tab`

Note: Users can always overrule

If your user decides to select a different tab behavior on any ticket, this action will be the new default behavior for that user.

Zammad remembers the decision of the user.

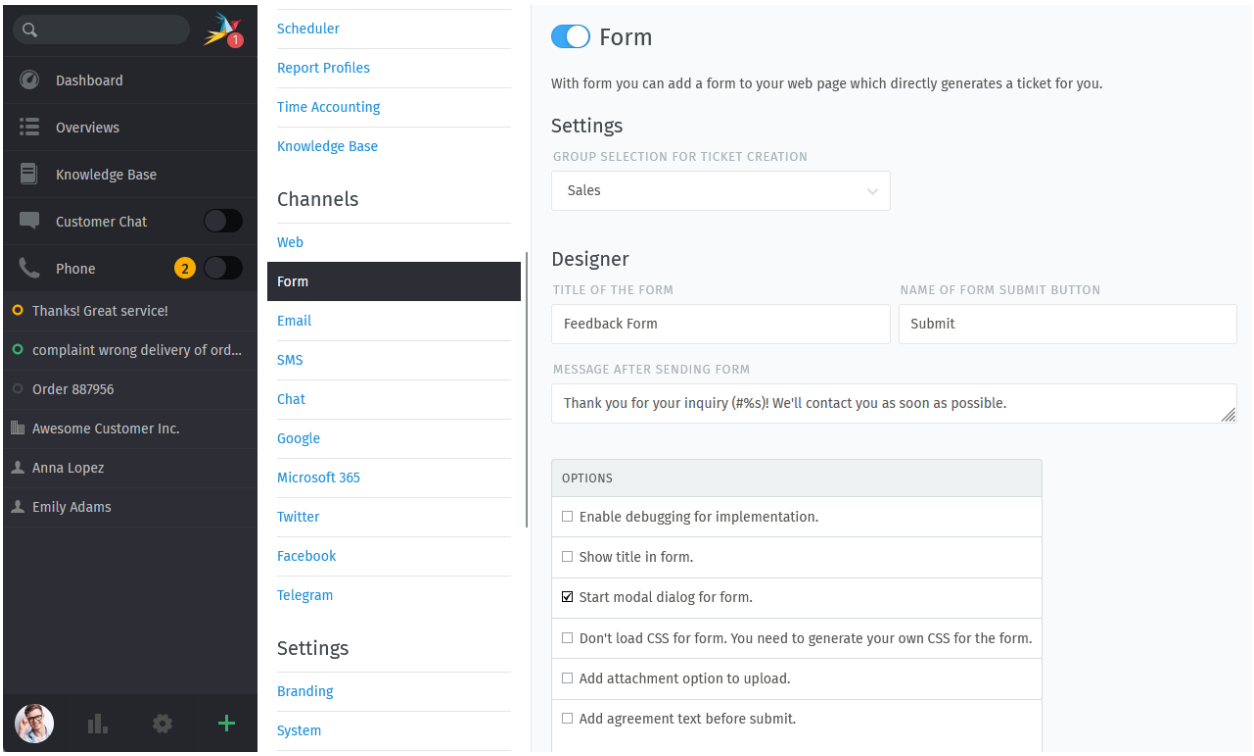


CHAPTER

TWENTY

FORMS

Feedback or contact forms are used on websites quite often. Usually they will generate an email which will be sent to somebody who forwards it and so on. With Zammad it's quite easy to integrate these forms into your website and directly generate tickets with them. In just 2 minutes.



20.1 Limitations

Please note the following limitations:

- The fields provided by the form are limited to the following:
  - Name
  - Email
  - Message
  - Attachment upload (optional)

  - Checkbox for custom agreement text (optional)
- As of now only one dedicated form per instance is possible

20.2 Settings

Zammad comes with certain settings for forms.

Active By default the form channel is *inactive*. Use the switch in front of "Form" to activate this channel.

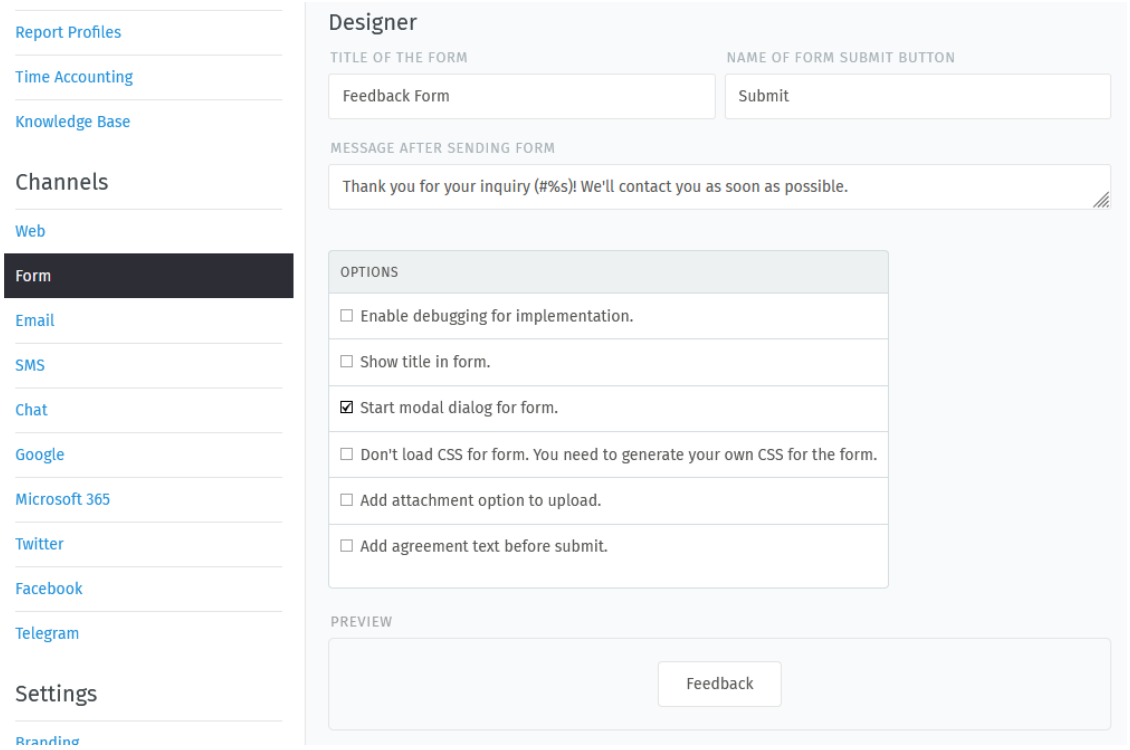
> **Note:** Forms will not be displayed on your website if the channel is not active. This does not affect the form preview on the channels setting page.

Group selection for ticket creation The group you set here defines where tickets should be created if they're supplied by Zammad's web form.

20.3 Designer

This section helps you to configure your form in the channel's scope. If you're happy with what you've chosen, you're provided with the code you need to copy to your website.

> **Warning:** The designer's changes *are not* stored anywhere in Zammad. This means that the provided source code needs to be copied every time you change settings here.



So let's talk about the options the designer provides.

Title of the form Choose how the heading of the form should be called. This setting is only relevant if you choose to display the form title in the form.

Default: `Feedback Form`

Name of form submit button If Zammad's default display name of the submit button does not fit, you can provide your own wording with this option. It will be used every time the form is shown.

Message after sending form After your user pressed the submit button, they will be provided with a message containing the ticket number of the newly created ticket.

Default *after* sending a form will look like so:

```
Thank you for your inquiry (#31015)!
We'll contact you as soon as possible.
```

Options Zammad provides the following additional configuration options for your form.

Enable debugging for implementation This option activates detailed debug information in your browser's developer tools console.

> **Warning:** This option *should not* be active on productive forms!

Show title in form This setting belongs to the setting `Title of the form` and will provide the form title within the form dialogue if selected.

Start modal dialog for form If selected, the form will be opened in a modal by clicking a button. Not selecting this option allows you to natively integrate the form within your website's body.

This option is set by default.

Note: No matter what you select here, the form *is always* loaded completely if your user browses the page containing the form.

Don't load CSS for the form You need to generate your own CSS for the form. By default Zammad's form comes with basic CSS. This may not fit your website's design or even interfere with it.

Selecting this option allows you to freely design the form without having to overwrite existing directives.

Add attachment option to upload Allows your user to upload **one attachment** to the form.

Note: Watch allowed attachment sizes here

This function *is not* limited technically. The only limitation that applies is your web servers upload limit.

Hint: SaaS only

If you're with Zammad hosted, attachments are limited by the package you've chosen.

Add agreement text before submit If enabled, this will allow you define a text that the form will display together with a checkbox. Thereby you can ensure your form conforms to legal require-

ments, e.g. by providing data privacy notes that the user has to accept before submitting the form.

Fig. 1: Zammad provides a free text form with limited capabilities. Use the *Weblink* button to add links to marked text passages.

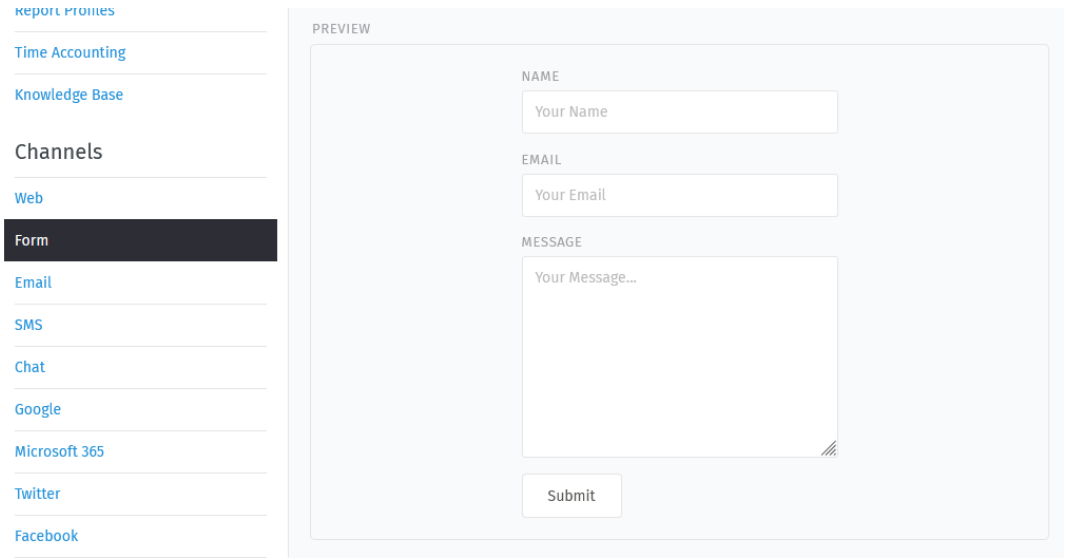
This allows you to link e.g. to your data privacy or ToS information.

20.3.1 Preview

Below the form options, Zammad provides a preview section to visualize the settings you've just chosen. By default you'll see a button named *Feedback*.

Clicking on the button will open the form modal.

Hint: If the form channel is set to active, you already are able to create tickets even from this preview mode.



20.4 Requirements

The requirement section provides you with everything you need in order to apply Zammad's web form to your website. It basically consists of two parts.

Header section The first code block provides you with Zammad's current jQuery dependency. This script section usually belongs to your website's header section.

Note: The channel form suggests the following script tag which loads the javascript libary required **from an external site**. This may not suite your local requirements. You can use locally hosted jQuery version, **however** the version is fixed.

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

> **Warning:** Do not mix jQuery versions - it's likely to break something.

Body section The second code block is the actual code required for your form to run.

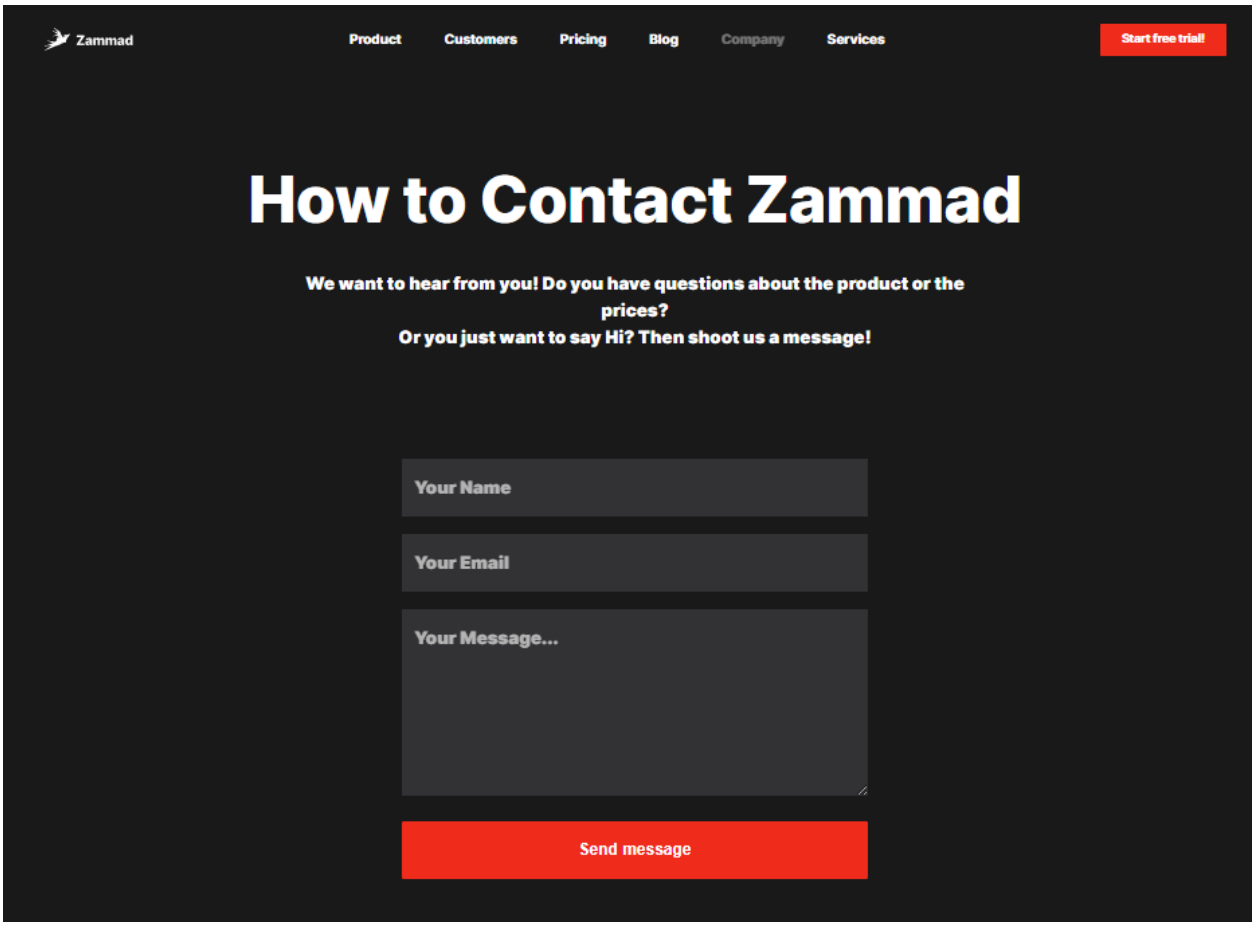
The first line containing `id="zammad-feedback-form"` highly depends on the settings selected above. This part belongs into the place you actually want the form or form button to appear.

The rest can either be placed at the same level or somewhere else on the same page.

> **Hint:** This code block is updated automatically when changing settings in the designer above.

> **Note:** These statements *highly* depend on your website. As you're responsible for your own website, you're on your own figuring out where exactly to place what.

Take the Zammad website as an example, the embedded form version can look like the following if you apply custom CSS to it.



20.5 Potential Spam Issue

The Form function could be abused by sending a high amount of messages / tickets to your Zammad instance. If you do not use Zammad Forms: **disable** them.

But don't worry! Zammad does limit the maximum count of created tickets based on different criteria. It also ensures that the email address being used is valid (with MX-Check on the email's domain).

Table 1: Form settings to limit ticket creation

Option	Default value	Description
`form_ticket_create_by_ip_per_hour`	`20`	Limits ticket creation per IP and hour to 20 tickets
`form_ticket_create_by_ip_per_day`	`240`	Limits ticket creation per IP and day to 240 tickets
`form_ticket_create_per_day`	`5000`	Limits ticket creation for forms to 5000 tickets per day

Hint:

You can change these default values. Please keep in mind, that setting those values higher might lead to problems in term of created tickets.

Please note that the following changes require console access to your Zammad host.

Change the ticket creation per IP and hour to 10:

```
rails> Setting.set('form_ticket_create_by_ip_per_hour','10')
```

Change the ticket creation per IP and hour to 50:

```
rails> Setting.set('form_ticket_create_by_ip_per_day','50')
```

Change the ticket creation per day to 500:

```
rails> Setting.set('form_ticket_create_per_day','500')
```

20.5.1 Further options to make it harder

Another way to make it harder for bots to automate against your Zammad instance is to change the location of `form.js` (`/opt/zammad/assets/form/form.js`).

Please keep in mind if you change the location of `form.js` (by e.g. copying) that you need to ensure that your `form.js` stays up to date if you update Zammad.

It's not recommended to delete form.js from it's location in that case, but to forbid access to it via your web server configuration.

CHAPTER

TWENTYONE

EMAIL

21.1 Accounts

21.1.1 Account Setup

Setting up a new email account? Here's what all the settings do.

But first, a word of warning! The import process does things you might not expect:

> **Danger: Zammad will delete all emails in your inbox during the import process.**
>
> Use the *Experts* dialog to *disable this behavior*.

> **Warning: Zammad will send an auto-reply message to every email it imports.** (Including the old ones!)
>
> Use the *Experts* dialog to *change this behavior*.

Note:

Gmail / G Suite users: Google is in the process of upgrading its security policies. To stay up-to-date, do **not** add your account as an email channel — instead, create a *Google channel*.

Microsoft 365 users: Microsoft is in the process of upgrading its security policies. To stay up-to-date, do **not** add your account as an email channel — instead, create a *Microsoft 365 channel*.

Basic

In most cases, Zammad is smart enough to figure out your email provider's configuration based on your email address alone.

Organization & Department Name The display name used for outgoing email.

If you add *multiple addresses* to a single account, you can define a separate Organization & Department Name for each one.

Email display names value can be *further customized in the Settings tab*.

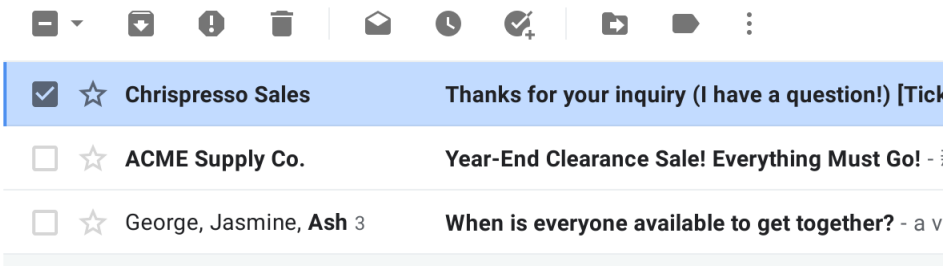


Fig. 1: A customer's inbox with an auto-reply from **Chrispresso Sales**.

Email Your email address.

If your account login/username is different from your email address, use the *Experts* dialog (see below).

If your inbox receives mail for more than one email address, be sure to *add your alternate addresses* after account setup.

Password Your account password.

Destination Group The *group* that incoming mail will be assigned to.

Use *filters* for more fine-grained sorting of incoming email.

Experts

If Zammad can't figure out how to connect your account (or if you just want to access advanced settings), use the *Experts* dialog.

Email Inbound

Type Choose from **IMAP** and **POP3**.

In most cases, you want IMAP. (With POP3, you won't be able to *keep messages on the server* or *specify which folder to fetch from*.)

Host Your email server's hostname or IP address (*e.g.,* `imap.gmail.com`).

Contact your email provider or system administrator if you don't know.

User This field is being pre-filled with your email address in case you've provided one before opening the expert settings.

Adjust this setting in case your username and email address differ.

Password Your account password.

SSL / STARTTLS Enable encryption when fetching messages.

You can choose from the following options:

- No SSL

> **Warning:** Retrieving Emails, just like sending your username and password without any encryption *is not secure*.
>
> You should never use this configuration on internet machines!

- SSL
- STARTTLS

Port Your email server's port (usu. `993` for IMAP, or `995` for POP3).

Contact your email provider or system administrator if you don't know.

Folder Specify which folder to fetch from, or leave empty to fetch from `INBOX`.

If specifying a nested folder, be sure to use the full path. (Some systems use different **path separators**; *e.g.,* `Inquiries/Tech-Support` vs. `Inquiries.Tech-Support`. Contact your email provider or system administrator if you don't know.)

Note: Additional Steps Required

In the last step of the account setup process, Zammad sends you an email from your own account, then waits for it to appear in the folder specified here. **Account verification will not complete until this test message has been received.**

If this folder does not receive incoming messages automatically, you may have to manually check your inbox during the verification step and move Zammad's test message there when it arrives.

Keep messages on server Specify what happens to your emails after Zammad imports them:

- `no` Zammad deletes all imported messages
- `yes` Zammad marks imported messages as read

  (With this option, Zammad will only import unread messages. This means Zammad may miss messages if the mailbox is externally modified.)

Note: Why does Zammad delete messages by default?

If you never clean out your inbox, it'll eventually reach its storage limit, and your mail server will start **rejecting incoming messages**. Most Zammad users never even look at their inbox once it's set up, so they rely on Zammad to keep it clean for them.

If you choose **yes** here, remember that it's your responsibility to clean out your inbox from time to time to keep it below its storage limit.

Import as

During the import process, Zammad treats **all messages** (including ones you've already read from months or years ago) as if they had been sent today: senders will receive auto-replies saying "your message has been received and we'll get back to you within 24 hours," and tickets created for each message will be marked as "new".

Use this option to disable this behavior for messages more than two weeks old.

Note: This option may not be shown if:

- all messages in your inbox are less than two weeks old
- you selected **Keep messages on server: Yes**

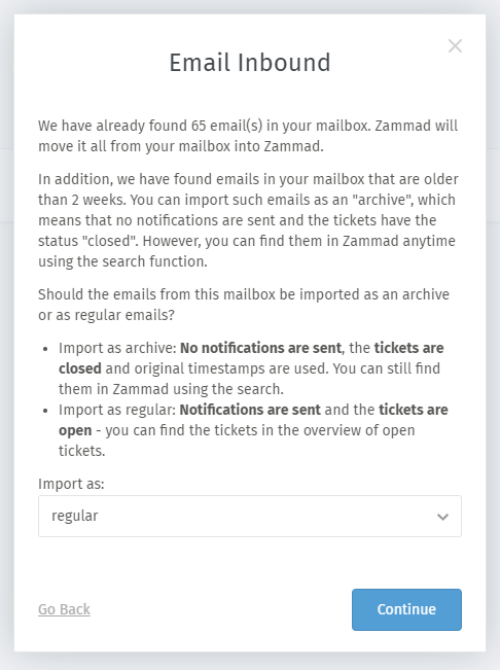


Fig. 2: How should old emails be imported?

- you selected **Type: POP3**

For more fine-grained control, manually disable this and other *triggers* before adding an email account, then turn them back on once all your messages have been imported.

Email Outbound

Send mails via Choose from **SMTP** and **local MTA** (*e.g.,* Sendmail).

Local MTA (mail transfer agent) configuration is only available on self-hosted installations.

Host Your email server's hostname or IP address (*e.g.,* `smtp.gmail.com`).

User Your account login/username.

Leave blank to use the same value from incoming account setup.

Password Your account password.

Leave blank to use the same value from incoming account setup.

Port Your email server's port (usu. `587` or `465`).

Zammad will detect and enable SSL/STARTTLS support automatically.

Verification

As a final step, Zammad sends a test email from your own account, to your own account, and to `verify-external-smtp-sending@discard.zammad.org` which discards the test mail right away.

We've created a landing page for discard.zammad.org which describes the backgrounds as well.

This this Zammad ensures that your email account is capable of sending internal and external - once this is verified the setup process is complete!

Hint: Don't forget to set the outgoing email address

In Zammad each group decides about the email address that's used for outgoing emails. The incoming group technically has no effect on it.

For this reason make sure to also adjust each affected *groups setting*.

Troubleshooting

- *Is a custom incoming mail folder to blame?*

21.1.2 Secondary Addresses

Secondary addresses (also known as *aliases*) allow you to send emails with a different "From:" address from the one on the account.

Fig. 3: Once you add a secondary address, you can configure a group to start sending emails with it.

Warning: Do not abuse this feature.

If you use secondary addresses to impersonate other parties, your IP is liable to get added to a spam blacklist.

Your email provider may also be set up to *receive incoming messages* for many addresses in the same mailbox. If this is the case, be sure to add your alternate inbox addresses here.

Display Name The display name used for outgoing email.

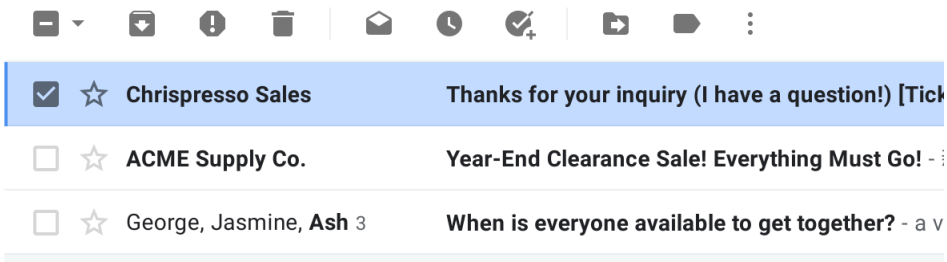


Fig. 4: A customer's inbox with an auto-reply from **Chrispresso Sales**.

Email display names value can be *further customized in the Settings tab*.

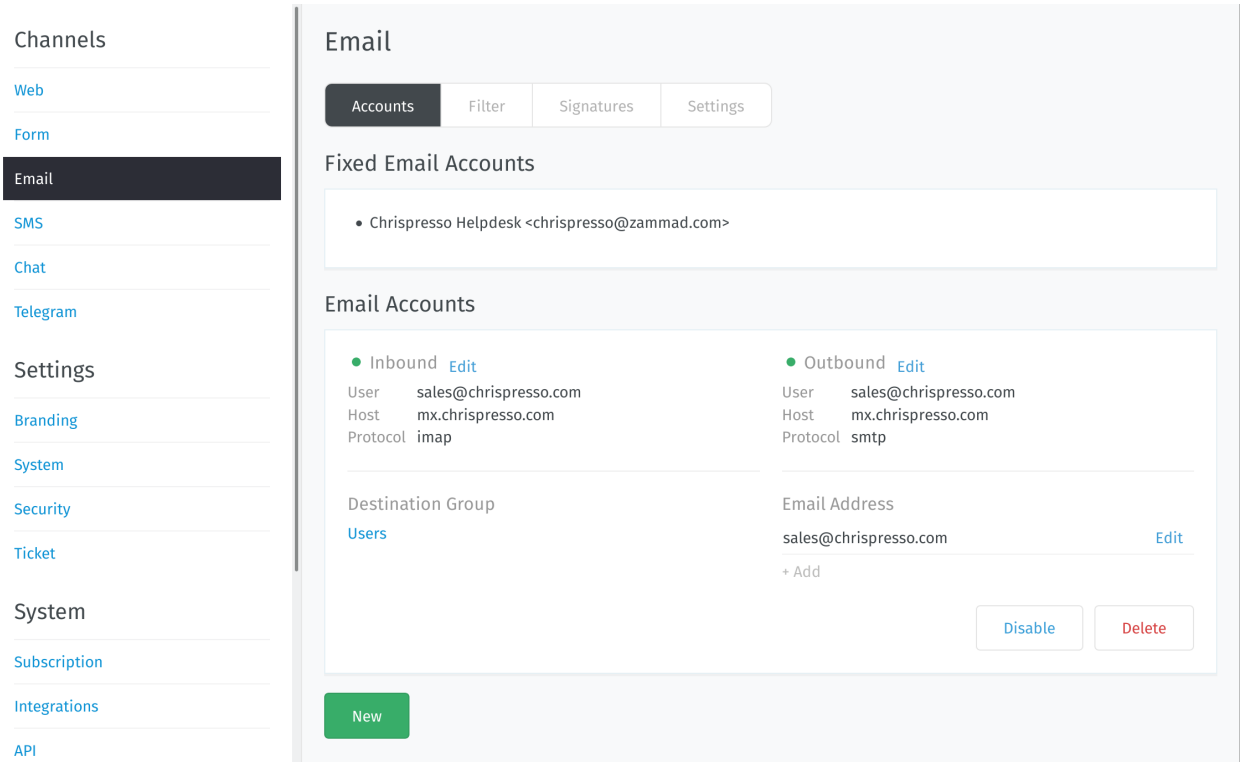
Email The alias address to send outgoing messages as.

Channel The email account to be used when sending outgoing messages from this alias.

Note Optional. Only visible from this dialog, the REST API, and the Rails console.

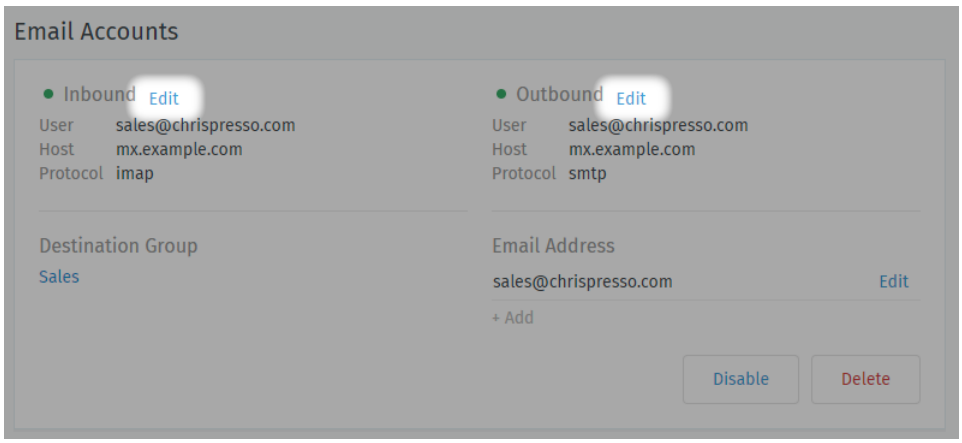
21.1.3 Managing Accounts

Once an account has been added, use the Accounts panel to edit its configuration.



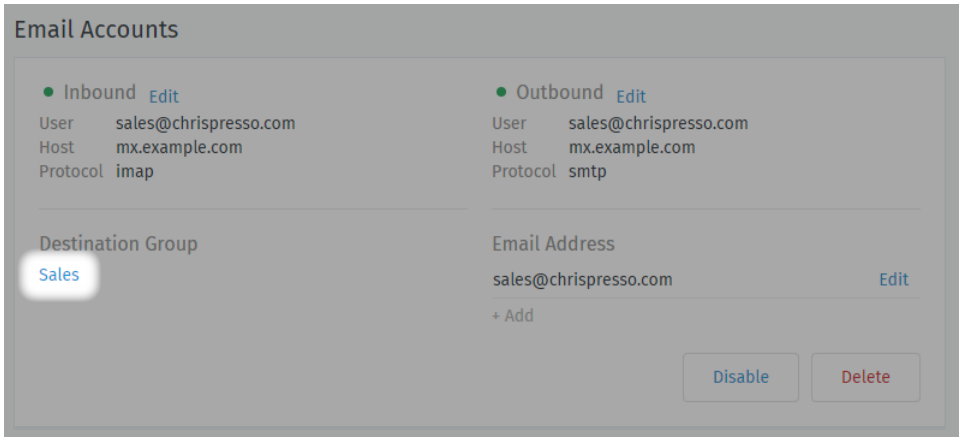
Mail Server Settings Click **Edit** on inbound/outbound account details to change your server configuration.

See *New Account Settings* for a detailed description of each option.



Note: In some browsers, you may have to manually re-enter your password.

Destination Group Click on the group name to reassign the account.



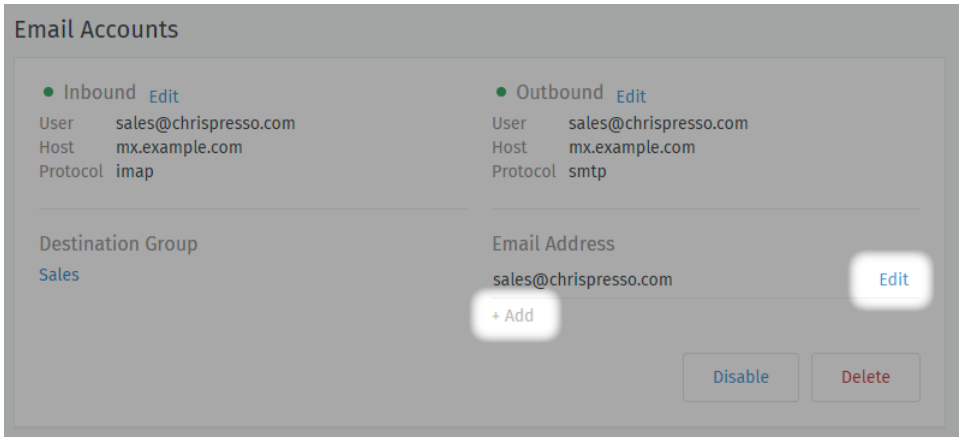
Only **active** groups will be displayed.

Changing this setting will not reassign existing tickets to the new group.

Hint: **Still can't send outgoing email tickets?** Check your *group settings*.

Email Address Use the **+ Add** or **Edit** buttons to set up secondary addresses on this account.

See *Secondary Addresses* for a detailed description of each option.



Enabled / Disabled Disabling an account temporarily prevents Zammad from importing its messages.

This may be necessary during scheduled maintenance or when migrating your installation to a new host.

Note: Disabling an account disables *outgoing* messages for it, as well.

Delete Deleting an account removes its configuration from Zammad entirely.

Note: **Additional Steps Required**

When an email account is deleted, its *email aliases* **remain in the system**. Be sure to reassign or delete them manually.

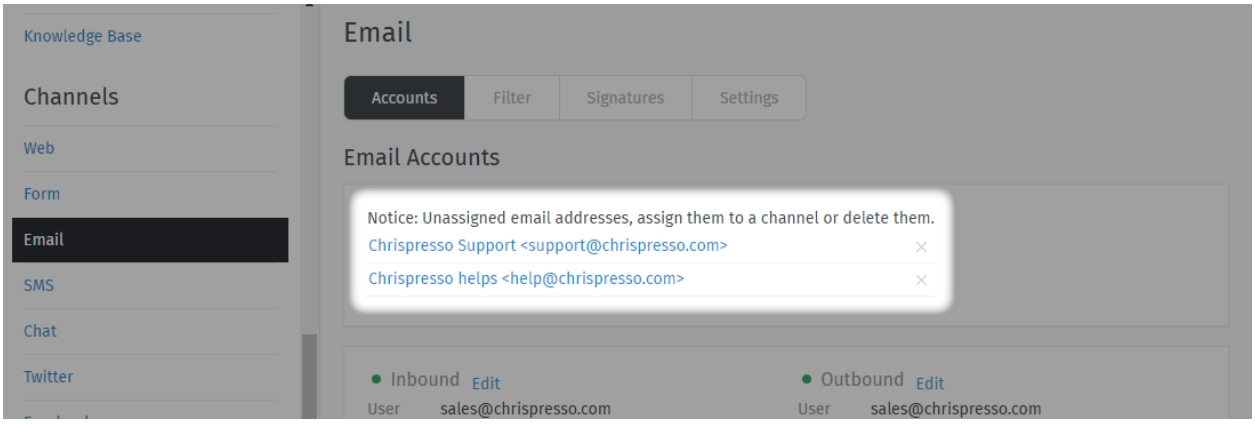


Fig. 5: Click on the address to assign it to another account, or click × to delete it.

Groups need an assigned an address to send outgoing emails. If you delete a group's assigned address, agents belonging to that group won't be able to send emails until you assign it a new one.

21.1.4 Email Notification

Note: The notification channel can only be configured on **self-hosted installations**.

For more information, see *System Notifications*.

System notifications are automated emails sent by Zammad for critical system events, such as account changes or SLA violations.

Use the **Email Notification** panel to configure how Zammad dispatches these notifications.

Send mails via Choose from **SMTP** and **local MTA** (*e.g.,* Sendmail).

Host Your email server's hostname or IP address (*e.g.,* `smtp.gmail.com`).

User Your account login/username.

Hint: The "From:" address on system notifications can be configured under *Channels* > *Email* > *Settings* > *Notification Sender*.

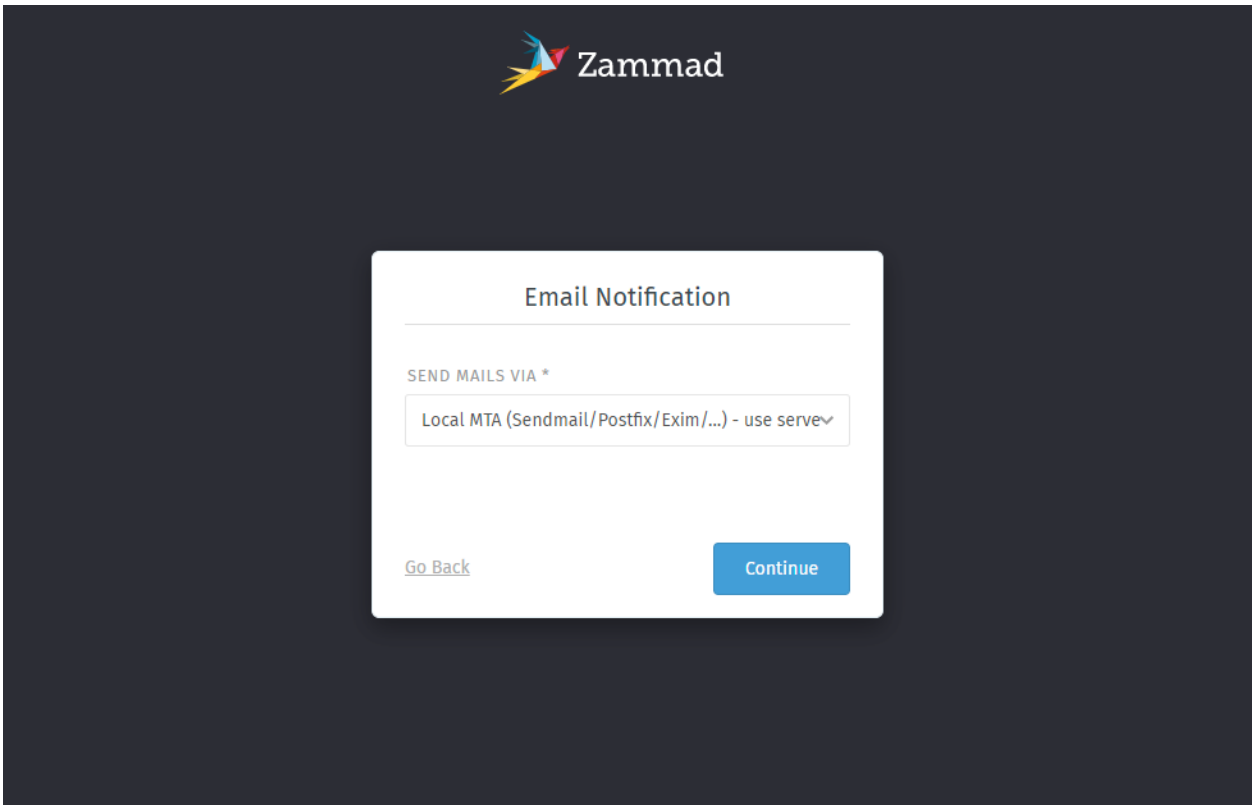
Password Your account password.

Port Your email server's port (usu. `587` or `465`).

Zammad will detect and enable SSL/STARTTLS support automatically.

Note: This looks familiar… Where have I seen it before?

This configuration step was part of the Getting Started wizard:



Account Setup Use the **New Email Account** dialog to connect your account.

Secondary Addresses Send and receive email at **additional email addresses**, all through the same mailbox/account.

Managing Accounts Edit the configuration of existing accounts in the **Accounts Panel**.

Email Notification Configure the outgoing mail provider for **system notifications**. (Self-hosted installations only.)

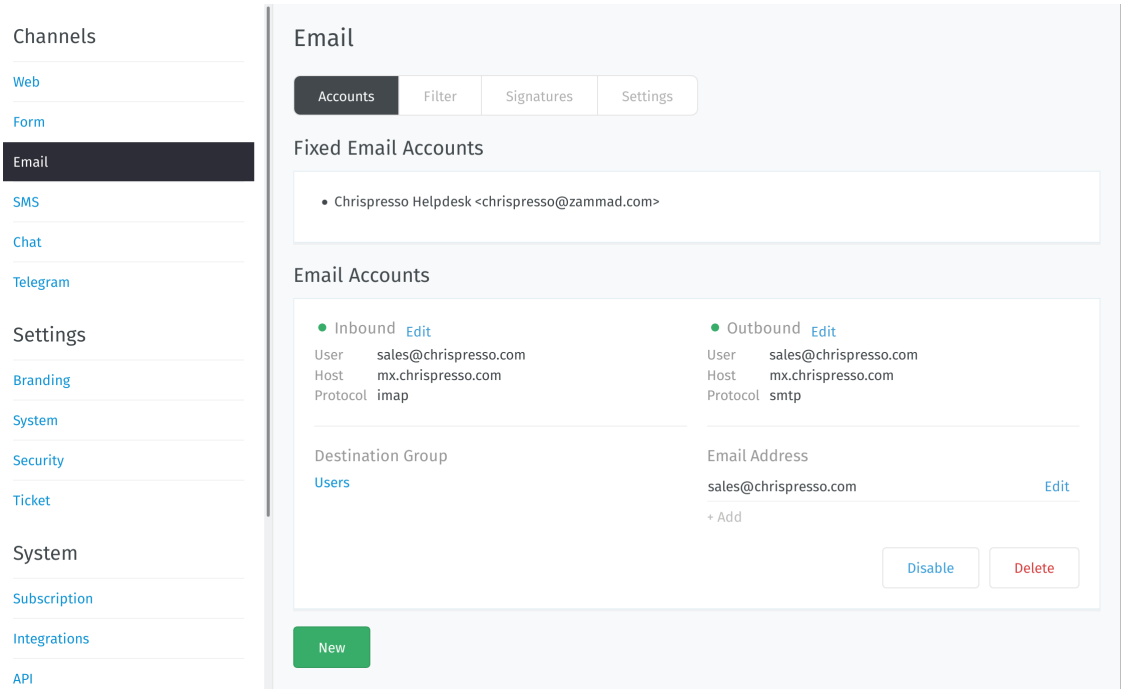
21.2 Filters

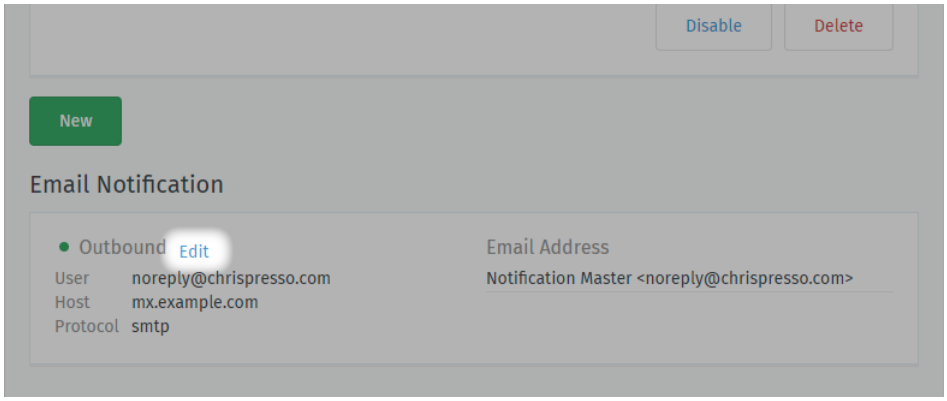
Postmaster filters allow you to match email headers (e.g. `From`, `To`, `Subject`, `X-Spam-Flag` etc.) and execute a set of actions whenever Zammad's email parser encounters a matching email. The actions will be applied to the ticket that is created or updated by this email. Here are some examples of what is possible with filters:

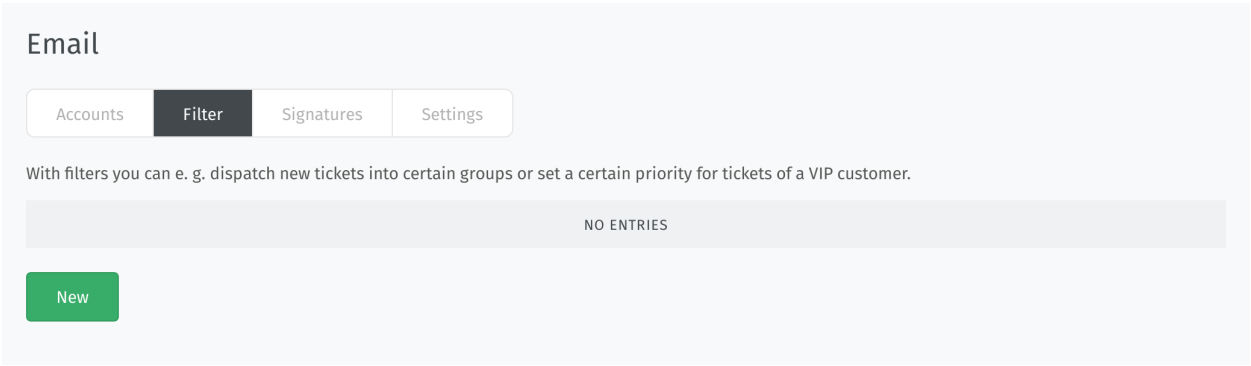
Automatically dispatch tickets into certain groups: For example, tickets from `amazon.com` could automatically be dispatched to the Purchasing group.

From: contains: `regex:(\.|@)amazon\.com`
Group: Purchasing

Note: Note that the Group action only has an effect when the matching email results in a new ticket. Zammad


Channels
Web
Form
Email
SMS
Chat
Telegram
Settings
Branding
System
Security
Ticket
System
Subscription
Integrations
API
Email
Accounts
Filter
Signatures
Settings
Fixed Email Accounts
Chrispresso Helpdesk <chrispresso@zammad.com>
Email Accounts
Inbound Edit
User sales@chrispresso.com
Host mx.chrispresso.com
Protocol imap
Outbound Edit
User sales@chrispresso.com
Host mx.chrispresso.com
Protocol smtp
Destination Group
Users
Email Address
sales@chrispresso.com Edit
+ Add
Disable
Delete
New


Disable
Delete
New
Email Notification
Outbound Edit
User noreply@chrispresso.com
Host mx.example.com
Protocol smtp
Email Address
Notification Master <noreply@chrispresso.com>


Email
Accounts
Filter
Signatures
Settings
With filters you can e. g. dispatch new tickets into certain groups or set a certain priority for tickets of a VIP customer.
NO ENTRIES
New

will not change the group of existing tickets.

Automatically increase the priority of tickets from a VIP customer:

From: contains: `ourvipcustomer@example.com`
Priority: 3 high

Note: Note that the Priority action only has an effect when the matching email results in a new ticket. Zammad will not change the priority of existing tickets.

Automatically tag and close spam tickets that have been marked as spam by anexternal spam filter (e.g. SpamAssassin):

X-Spam-Flag: contains: `YES`
Tag: add: `spam`
State: closed

Note: Note that the State action only has an effect when the matching email results in a new ticket. Zammad will not change the state of existing tickets. It will add the tag though if it missing, even if the mail is an update to an existing ticket.

The following actions are only effective when creating tickets:

- Group
- State
- Priority
- Owner

Different attributes of a filter can be combined with each other. Likewise, the following actions can be combined. The supported matches are "contains" and "contains not"; for advanced matching, you can use regular expressions by prefixing the string with `regex:`.

Note that Zammad matches against the full header, e.g. for a mail with `From: Display Name <display.name@example.com>`, the `From` condition will test against `Display Name <display.name@example.com>`. This is especially important when using anchored regular expressions; `regex:^display\.name@example.com$` would not match this mail!

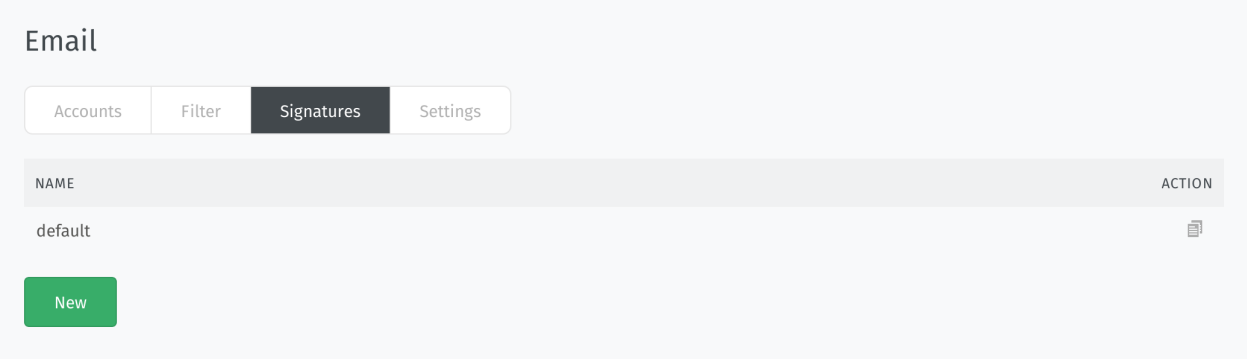
It should be borne in mind that the combined attributes build on each other. If a filter is no longer needed, it can either be temporarily set inactive or deleted directly.

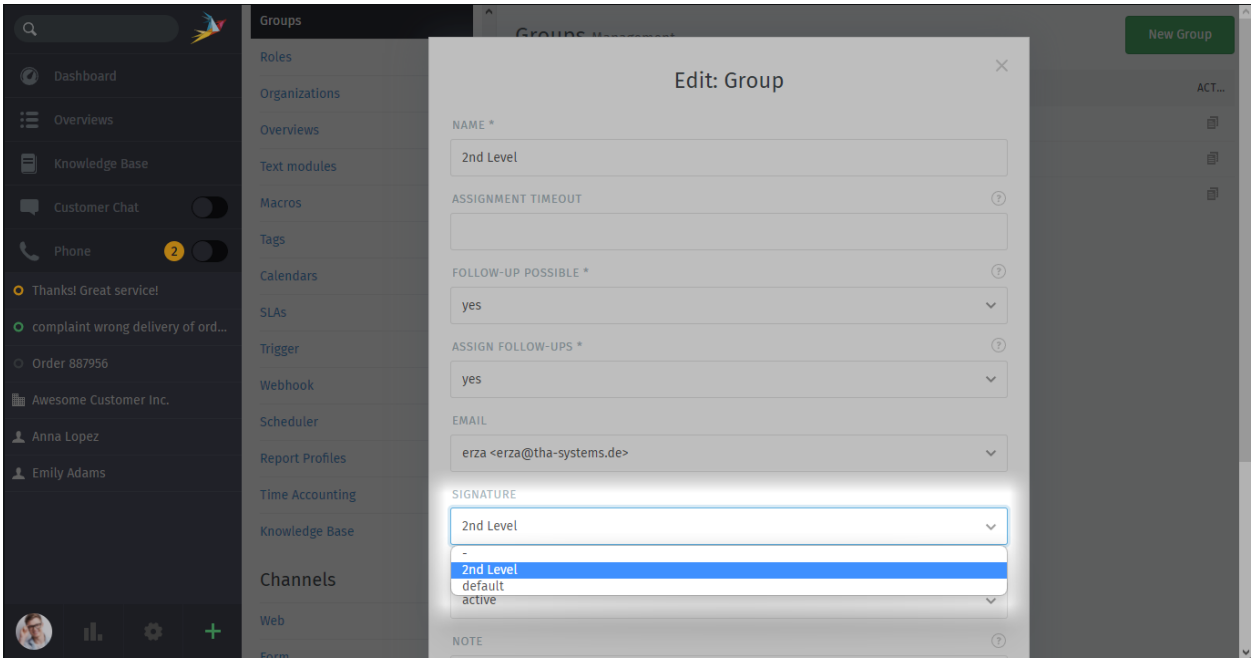
21.3 Signatures

You can create a separate signature for each group in Zammad. The individual signatures can be created and edited here.

Afterwards, the existing (and active) signatures are available in the group editing mask:

Each group can be assigned its own signature, but they can also all use the same signature.


Email
Accounts
Filter
Signatures
Settings
NAME
ACTION
default
New

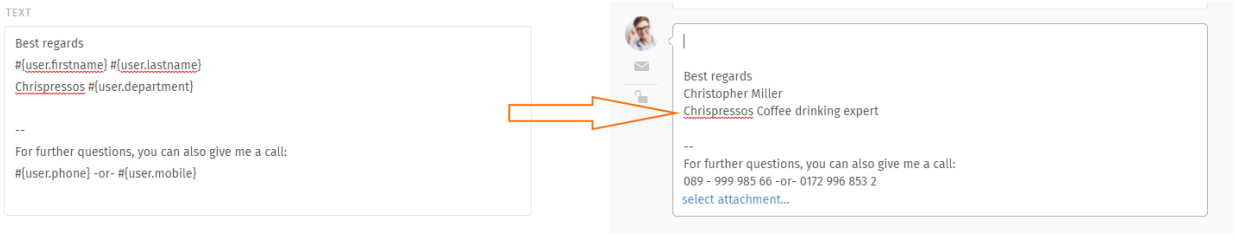

Groups
Roles
Organizations
Overviews
Text modules
Macros
Tags
Calendars
SLAs
Trigger
Webhook
Scheduler
Report Profiles
Time Accounting
Knowledge Base
Channels
Web
Dashboard
Overviews
Knowledge Base
Customer Chat
Phone
Thanks! Great service!
complaint wrong delivery of ord...
Order 887956
Awesome Customer Inc.
Anna Lopez
Emily Adams
New Group
Edit: Group
NAME *
2nd Level
ASSIGNMENT TIMEOUT
FOLLOW-UP POSSIBLE *
yes
ASSIGN FOLLOW-UPS *
yes
EMAIL
erza <erza@tha-systems.de>
SIGNATURE
2nd Level
-
2nd Level
default
active
NOTE

21.3.1 Dynamic Signatures

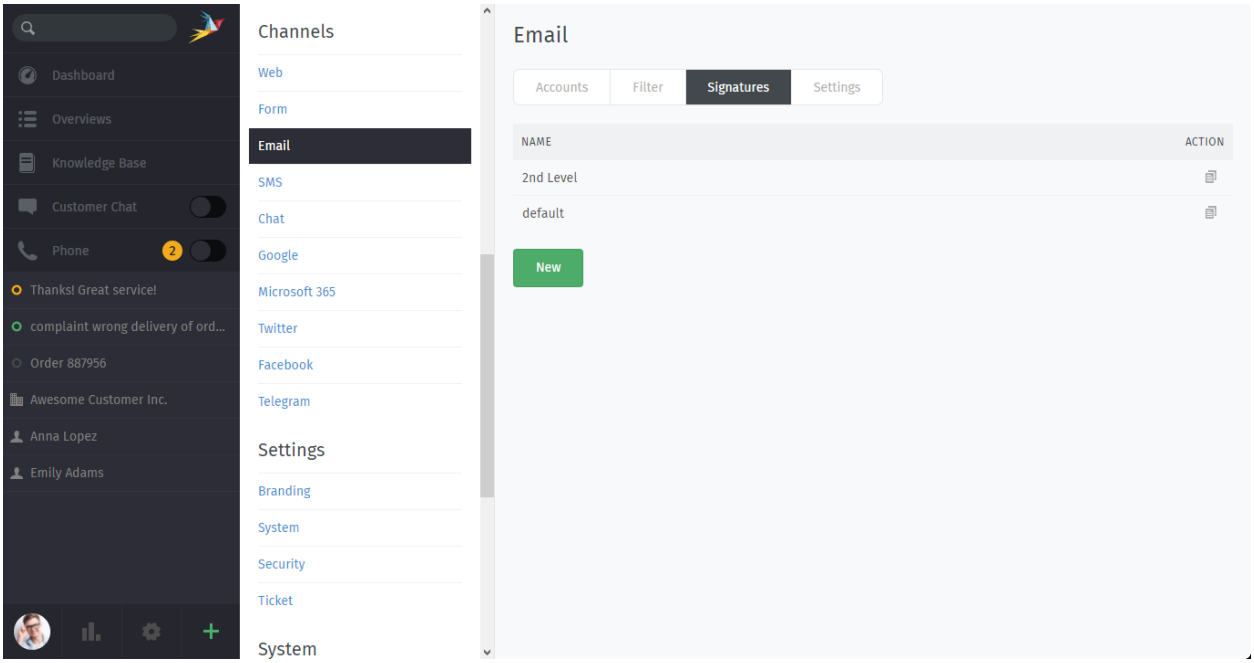
To individualize the signatures, it is possible to automatically load specific information into a signature via *Variables*. All information stored on the ticket, assigned customers or agents can be inserted. This makes it possible to design the signature individually. To load a list of available variables, enter two colons (`::`) into the *Text* box of the signature editor.

Hint: Please keep in mind that specific information might not be available during ticket creation. The best example here is the ticket number / id. Specific information are created with submitting the ticket and thus are not available before submitting.

Here is an example of a signature with variables and the result when you write a mail:



21.4 Settings



Below you can find the currently available email-related settings. Most of these settings have default values which can be found in this list as well.

Note: Some email-related settings are ticket-based settings, which is why they can be found in the *Composer Settings*.

21.4.1 List of Settings

Notification Sender: Default value `Notification Master <noreply@#{config.fqdn}>` This is the default sender address for Zammad that affects all mails but those generated because of replies (like triggers or agent-based mails). Your customers normally will not see this address. This email address does not need to receive and can't be assigned to a group.

Note: This address is relevant for agent notifications and password reset mails (also affects customers).

Additional follow-up detection In some situations the normal follow-up detection is not enough. This might be due to missing references in the subject (the ticket hook and number). These options can help to recognize follow-ups to existing tickets.

Note: Please note that searching in attachment and body might lead to false follow-up detection.

Maximum Email Size: Default value `10 MB` This one is pretty obvious: It defines the maximum allowed size of an email Zammad will fetch. Zammad will not fetch Mails that are bigger than this option.

Hint: This technically also affects attachments for articles.

Send postmaster mail if mail too large: Default value `yes (enabled)`

Note: Upgraded installations will, by default, have the value set to `no (disabled)`.

Option set to `yes` This setting will cause Zammad to automatically reply to mails that exceed the above mail size limit with a postmaster style mail. This will help your user to understand that his mail did not arrive and won't be reviewed by you.

Note: Zammad will still download and remove (if enabled) the mail from the mailbox. Instead of importing it to the database, it will save the affected mail to `/opt/zammad/tmp/oversized_mail/`.

Option set to `no` If the option is set to no, Zammad will not reply to mails that are too big. Your customer will **not notice** that the mail was too large! Instead, Zammad will use the monitoring endpoint to alert its administrators that it can't fetch a too large mail.

Learn more about *Monitoring*.

Sender based on Reply-To header: Default value `not set (-)` This setting decides how Zammad should recognize its customers from emails that contain a `Reply-To` header. This comes in useful if you're working with contact forms that need to use reply to headers.

Option set to `Take reply-to header as sender/from of email.` This setting will overwrite the initial `FROM` to the value used in `Reply-To` completely.

Option set to - or `Take Reply-To header as sender/from of email and use the real name of origin from.` This setting will partially overwrite the initial `FROM`. It uses the mail address from the `Reply-To` header and uses the given name of the `FROM` header, if given.

Customer selection based on sender and receiver list: Default value `yes` This option decides how Zammad should react if an agent sends a email to it.

Option set to `yes` The first user / email address from the recipient list will be used as the ticket customer.

Option set to `no` The agent will be set as ticket customer.

Note: Currently agents can't be customers within the UI. While Email communication works, agents can't see their own tickets (as a customer) if they don't have access to the group.

Block Notifications With the regex that can be defined here, you can ensure not to send any notifications to specific systems. By default this especially affects typical system addresses which can't receive emails anyway.

The default value is: `(mailer-daemon|postmaster|abuse|root|noreply|noreply.+?|no-reply|no-reply.+?)@.+?`

Sender Format: Default value `Agent Name + FromSeparator + System Address Display Name` This configures the display name used in the `FROM` header of mails Zammad sends.

Note: This does not affect Notification mails (to agents) and password reset mails. Emails that are not sent by agents (e.g. trigger-based notifications) will always fallback to `System Address Display Name` if needed.

Option set to `Agent Name + FromSeparator + System Address Display Name` This will cause Zammad to set the `FROM` header to agent name and the channel's display name, divided by a separator (configured below).

Example: `Christopher Miller via Chrispresso Inc.`.

Option set to `System Address Display Name` This will cause Zammad to always use the display name of the used channel in the `FROM` header.

Example: `Chrispresso Inc.`

Option set to `Agent Name` Zammad will use the agent's name which is very personal.

Tip:

Usually you'd also want to remove the ticket slug from the subject in those cases.
Learn more in *Settings → Ticket*.

Sender Format Separator: Default value `via` This is a can be a string you can freely choose. It divides the agent's name and the display name of the channel whenever needed.

Ticket Subject Forward: Default value `FWD` The above string will be used on the subject if you forward an email from Zammad.

Note: `:` will be automatically appended to the above string.

Ticket Subject Reply: Default value `RE` The above string will be used on the subject if you reply to a mail from Zammad.

Note: : will be automatically appended to the above string.

Ticket Subject Size: Default value `110` This setting enforces a maximum length for subjects when replying. If the subject you're using for your reply is too long, Zammad will automatically truncate the length and insert `[...]` to show it has shortened the subject.

Example: `RE: Test somew[...] [Ticket#123456]`

Note: This does **not** limit ticket titles within the UI, just the subjects when replying to an email.

21.4.2 Enhanced settings

Some less relevant settings can be changed via rails console if needed. As an example, Zammad allows you to send all outgoing communication to a BCC address for archiving reasons if needed. You can find the needed commands within the advanced customization settings.

21.5 Email header manipulation

Email header manipulation allows you to re-route or adjust tickets apart from filters or triggers. Like an API call, but with emails.

Header checks are case insensitive.

Warning: Trusted channels required

Below options are a potential risk with external communication and thus require channels being set to trusted explicitly.

Tip: Below headers are examples and –in our opinion– the most relevant ones. However: You can adjust mostly any article or ticket attribute (yes, custom ones as well) if you know the attribute's exact name.

The name column within *object's management* provides easy access to objects attribute names.

21.5.1 Trigger auto responses

Normally Zammad runs internal checks to see if an email is an automatic response. In these cases Zammad will not send trigger based responses.

There may be use cases where this behavior may be in your way, below options allow you to overcome this issue.

Note: In some cases combining below headers is crucial. This is intentional but may be confusing.

`x-zammad-send-auto-response` Set to `false` to disable trigger based responses. If set to `true` Zammad will send a response.

Hint: This option *does not* work if e.g. `precedence: list` is set unless you use below auto response header as well.

x-zammad-is-auto-response Providing this header allows you to tell Zammad that the mail in question is an auto generated response (`true`). This will cause email based triggers to be skipped.

Set this header to `false` if you want to generate auto responses.

Tip: This header allows you to overwrite auto detects for e.g. `precedence: list`.

21.5.2 Ticket attributes

Zammad allows you to use headers to manipulate ticket creations or follow ups. The manipulation can be used instead of triggers. Triggers are considered *after* header settings and thus can still overrule.

Note: Zammad differentiates between ticket creation and follow up

For creations use: `X-Zammad-Ticket-{Attribute Name}`
For follow ups use: `X-Zammad-Ticket-FollowUp-{Attribute Name}`

This allows you to ensure the changes are only applied in the required situation.

Warning: About values

- While headers are not case sensitive, values like e.g. priority names are case censitive: `1 low` will work, but `1 lOw` will not!
- When using attributes that require date / time values, ensure to use Time Zoned Times. e.g. for 28th September 2021 on 8 am CEST, either use:
  - `2021-09-28T08:00:00+0200`
  - `2021-09-28T08:00:00+02:00`
  - `2021-09-28T06:00:00.000Z`

X-Zammad-Ticket-Priority & X-Zammad-Ticket-FollowUp-Priority

Allows you to adjust a ticket's priority.
Example: `X-Zammad-Ticket-Priority: 1 low`

X-Zammad-Ticket-Group & X-Zammad-Ticket-FollowUp-Group

Allows you interfere with regular channel routing of the ticket.
Example: `X-Zammad-Ticket-Group: Sales`

X-Zammad-Ticket-Owner & X-Zammad-Ticket-FollowUp-Owner

Directly assign or change the ticket owner. Valid values are either `login` or `Email`
Example: `X-Zammad-Ticket-Owner: jdoe`

X-Zammad-Ticket-State & X-Zammad-Ticket-FollowUp-State

Set a specific ticket state.
Example: `X-Zammad-Ticket-State: closed`

> **Note:**
>
> Pending states always require the `pending_time` attribute on top.
> Like so: `X-Zammad-Ticket-Pending_Time: 2021-09-26T08:00:00+0200`

X-Zammad-Customer-Email

Manipulate the ticket customer - this can be a different user than the actual sender. Replying to the original sender is still possible.
Example: `X-Zammad-Customer-Email: jdoe@example.com`

> **Note:** This header is not available for follow ups.

X-Zammad-Customer-Login

Manipulate the ticket customer - this can be a different user than the actual sender. Replying to the original sender is still possible.
Example: `X-Zammad-Customer-Login: jdoe`

> **Note:** This header is not available for follow ups.

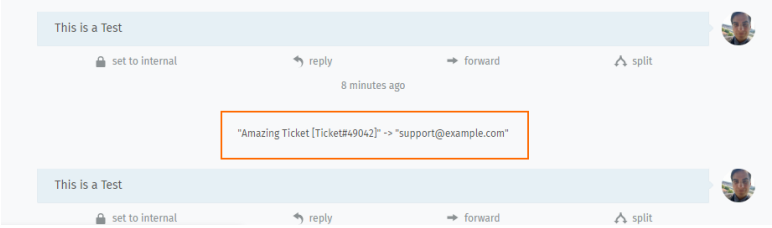
21.5.3 Article attributes

If needed Zammad allows you to manipulate attributes or states of fetched email articles.

X-Zammad-Article-Sender

Manipulate the sender type (Agent, Customer, or System)
Example: `X-Zammad-Article-Sender: System`

> **Warning:** System Emails are indicated in a similar way as trigger-response like entries Users can't see them natively.



X-Zammad-Article-Type

Change the article type of your incoming mail. This requires you to know which article types are available in your system.
Example: `X-Zammad-Article-Type: phone`

> **Warning:** This header can cause *serious issues* in your instance and may lead to unexpected behavior. Only use with absolute care!

`X-Zammad-Article-Internal`

Manipulate the default article visibility.
Example: `X-Zammad-Article-Internal: true`

`X-Zammad-Ignore`

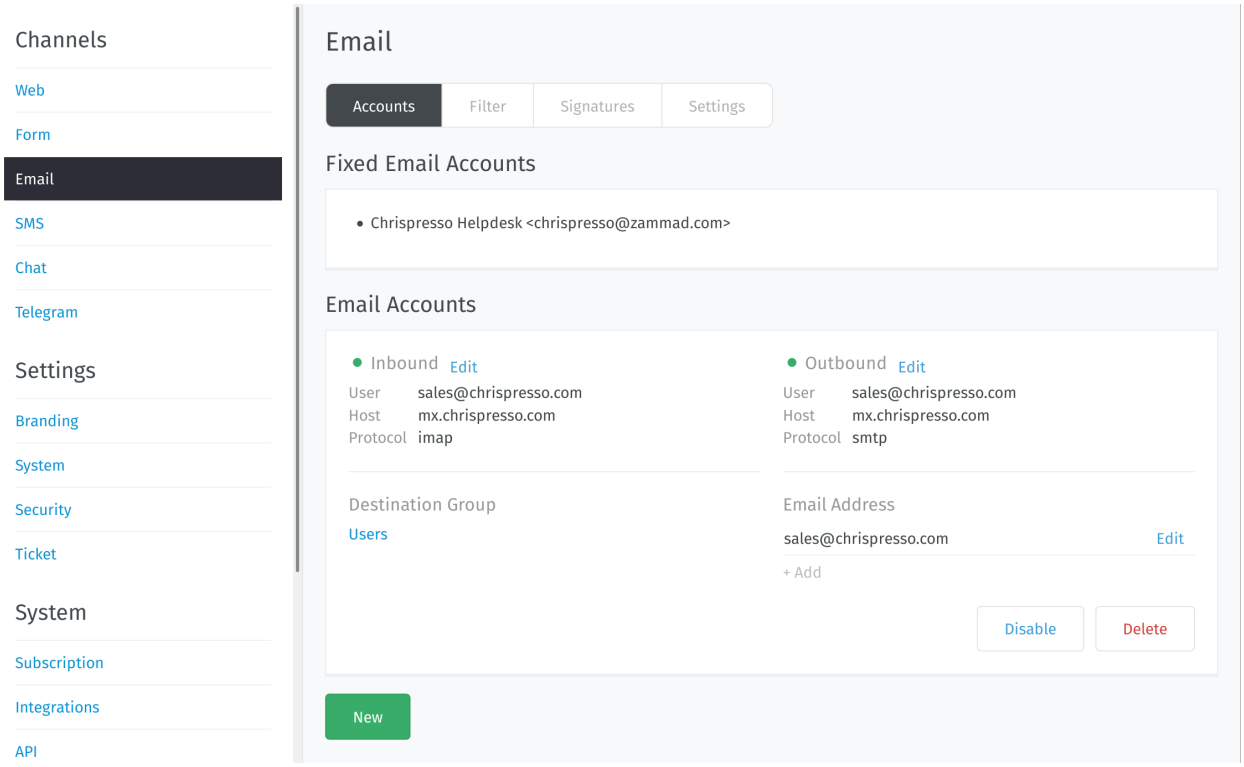
Tell Zammad to silently drop the Email.
Example: `X-Zammad-Ignore: true`

Control how Zammad **sends and receives email**.

Hint:

Using Gmail / G Suite? Set up a *Google channel* instead.

Using Microsoft 365? Set up a *Microsoft 365 channel* instead.



Migrate existing email channel to "XOAUTH" channel At this moment Zammad supports XOAUTH for the following providers:

- *Google*
- *Microsoft 365 (formerly Office 365)*

Use above links to use the migration option instead of removing and re-adding the channels. This will save precious time for something else!

Accounts **Connect Zammad to your email provider** so that it can watch your inbox, send auto-replies, and more.

(Self-hosted users may have already completed this step during new system setup.)

Filters **Make sure new tickets show up in the right place** with automated, if-this-then-that rules for all incoming email.

Signatures Customize signatures for all outgoing email.

Settings Manage options like:

- set the "From:" address on system notifications
- raise the limit on attachment sizes
- modify subject-line prefixes (*e.g.,* use "AW:" instead of "RE:")

Hint: Want to **manually edit email subjects** or **always copy parent messages into your replies**?

Check the *Composer Settings*.

Header manipulation Manipulate auto response behavior or incoming routing.

> **Warning:** This is a very advanced topic.

21.6 Extra Options for Self-Hosted Users

If you're too cool for POP3/IMAP/SMTP...

21.6.1 Watch your inbox with Fetchmail

Maybe you want to add emails via Fetchmail or Procmail to Zammad.

To get this to work you need to pipe your emails to rails.

Note: If you installed Zammad through a package manager (rather than from source), replace `rails r` with `zammad run rails r` below. To learn more, see Administration via Console.

Command line:

```
su - zammad
cd /opt/zammad
cat test/fixtures/mail1.box | rails r 'Channel::Driver::MailStdin.new(trusted: true)'
```

Fetchmail

Create .fetchmailrc:

```
su - zammad
cd ~
touch .fetchmailrc
chmod 0600 .fetchmailrc
```

vi .fetchmailrc:

```
#
# zammad fetchmail config
#
poll your.mail.server protocol POP3 user USERNAME pass PASSWORD mda "rails r
↪'Channel::Driver::MailStdin.new(trusted: true)'"
```

That's it. Emails now will be directly piped into Zammad.

Using Procmail for advanced features like presorting

If you want to do some more with your emails, like presorting to a Zammad group or filtering spam, you can use Procmail.

Fetchmail config looks slightly different.

vi .fetchmailrc:

```
#
# zammad fetchmail config
#
poll your.mail.server protocol POP3 user USERNAME pass PASSWORD mda /usr/bin/procmail is␣
↪zammad here
```

Create .procmailrc:

```
su - zammad
cd ~
touch .procmailrc
```

vi .procmailrc:

```
# --
# Pipe all emails into Zammad
# --
PATH=/opt/zammad/bin:/opt/zammad/vendor/bundle/bin:/sbin:/bin:/usr/sbin:/usr/bin:
SYS_HOME="/home/zammad"
RAILS_ENV=production
GEM_PATH=/opt/zammad/vendor/bundle/ruby/2.4.1/
LOGFILE="$SYS_HOME/procmail.log"
#VERBOSE="on"

:0 :
| rails r 'Channel::Driver::MailStdin.new(trusted: true)'
```

21.6.2 Dispatch messages with Sendmail

> **Warning:** For the initial setup of this you need administrative rights on the Zammad machine (console).

If you try to configure **only** an outgoing email account (as in, you do not wish to set up an incoming IMAP/POP3 account at all), you will find that it's simply not possible via the email channel setup wizard. Instead, you will have to create it via the CLI.

(The wizard is designed to provide an idiot-proof email configuration process for the average, non-technical user, so certain advanced options and use cases have been deliberately omitted.)

To configure Zammad to use sendmail, run the following command (you can use rails r [...] if you installed Zammad from source):

```
zammad run rails r "Channel.create(area: 'Email::Account', options: { inbound: {␣
↪adapter: 'null', options: {} }, outbound: { adapter: 'sendmail' } }, active: true,␣
↪preferences: { editable: false }, updated_by_id: 1, created_by_id: 1)"
```

Now, you should see a new **Email Account** entry in the admin settings panel:

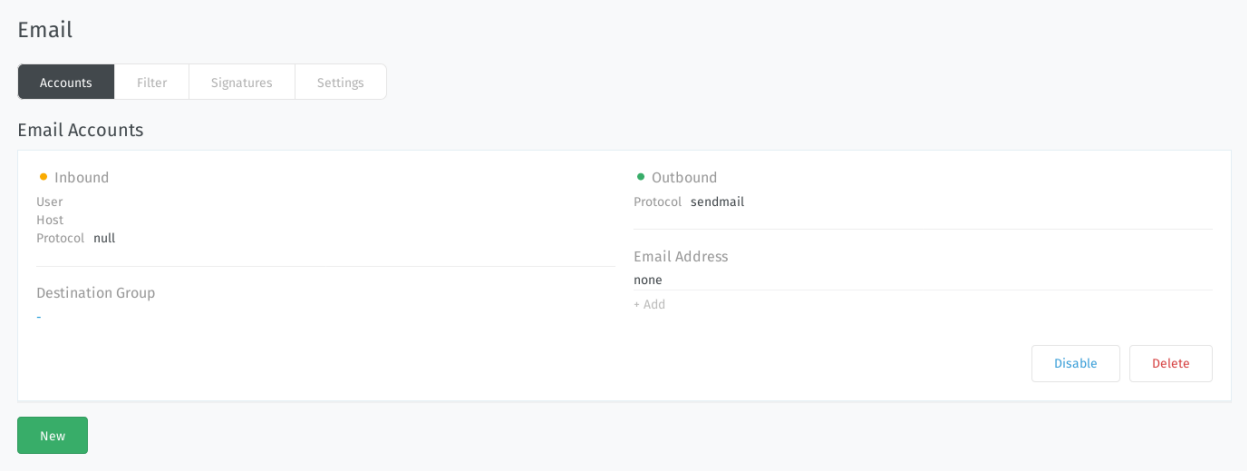


Fig. 6: Use the **Add** button under the **Email Address** heading to add new email addresses to send from.

CHAPTER

TWENTYTWO

CHAT

Chat has gotten very important for company and overall customer support. If used properly, support via chat can be a real efficiency booster. A downside of chats is when nobody responds or a bot responding to the customer.

22.1 How to improve Support by Chat?

We've asked others about their opinion in our circles, to learn what people expect or dislike. This is what we found out:

Good experiences

- Getting personal support by a human being
- Getting a fast response
- Solving my problem quickly

Bad experiences

- A chat window on a website (while the chat being offline) with the hint to "Leave a message"
- Long waiting queues before even writing with a personal
- Receiving a message like "My name is Nina, what can I do for you?" after sending a message with my issue.
- A Chat that doesn't integrate itself into the Website properly

22.2 Our answer: The Zammad Chat Widget

The task is clear: Work on the disadvantages of a regular support chat and improve them.

Our approach is as follows

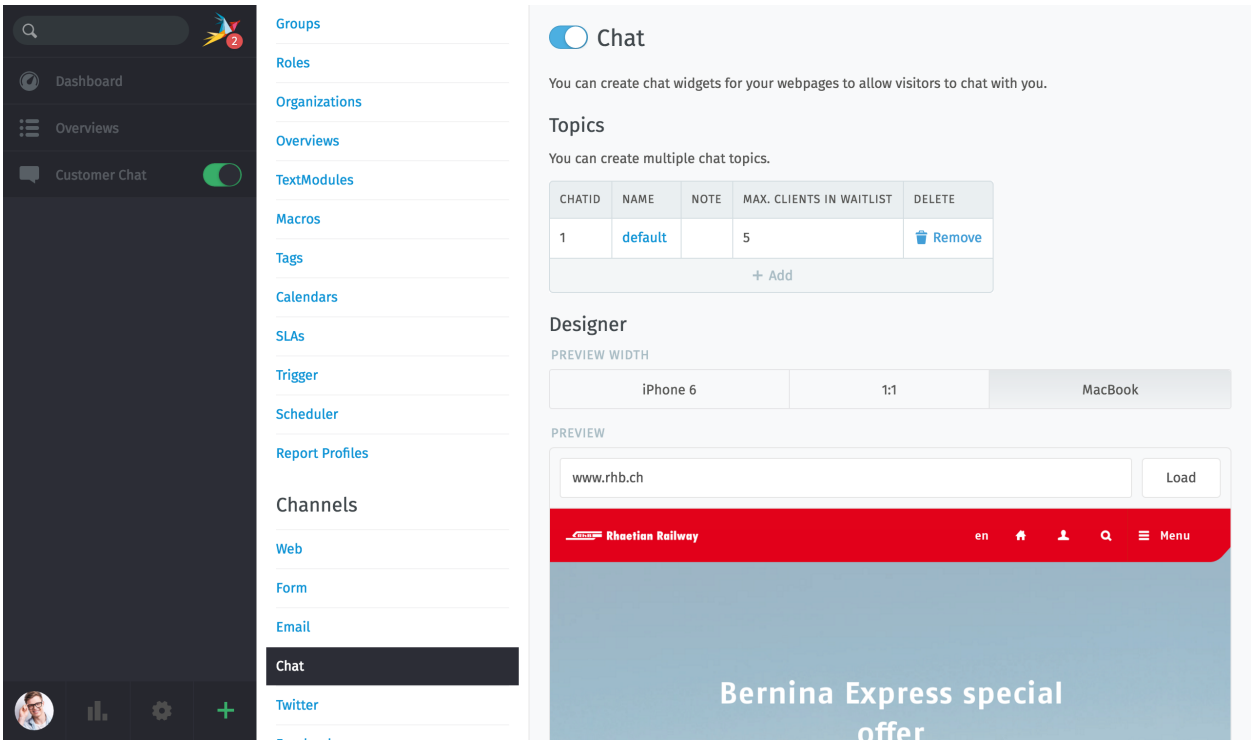
- We'll only display the chat widget, if at least one agent is available and the agent still has capacity.
  - If no agent is online or the agents are absend, the chat won't be available
- We're setting an agent as inactive, if the agent doesn't accept new chat requests or the WebApp is offline. This way, your support staff can take breaks without your customers waiting ages for a reaction (see point above)
- Zammad does not respond to chat messages on it's own to ensure that there's no strange delay coming afterwards. Zammad will fire a (configurable by agent) auto response as soon as the agents accepts the Chat request.

- Zammad will try to adapt your main website colors to the chat. You can also adjust those colors allowing you to integrate the chat into your website, like it has been there before.

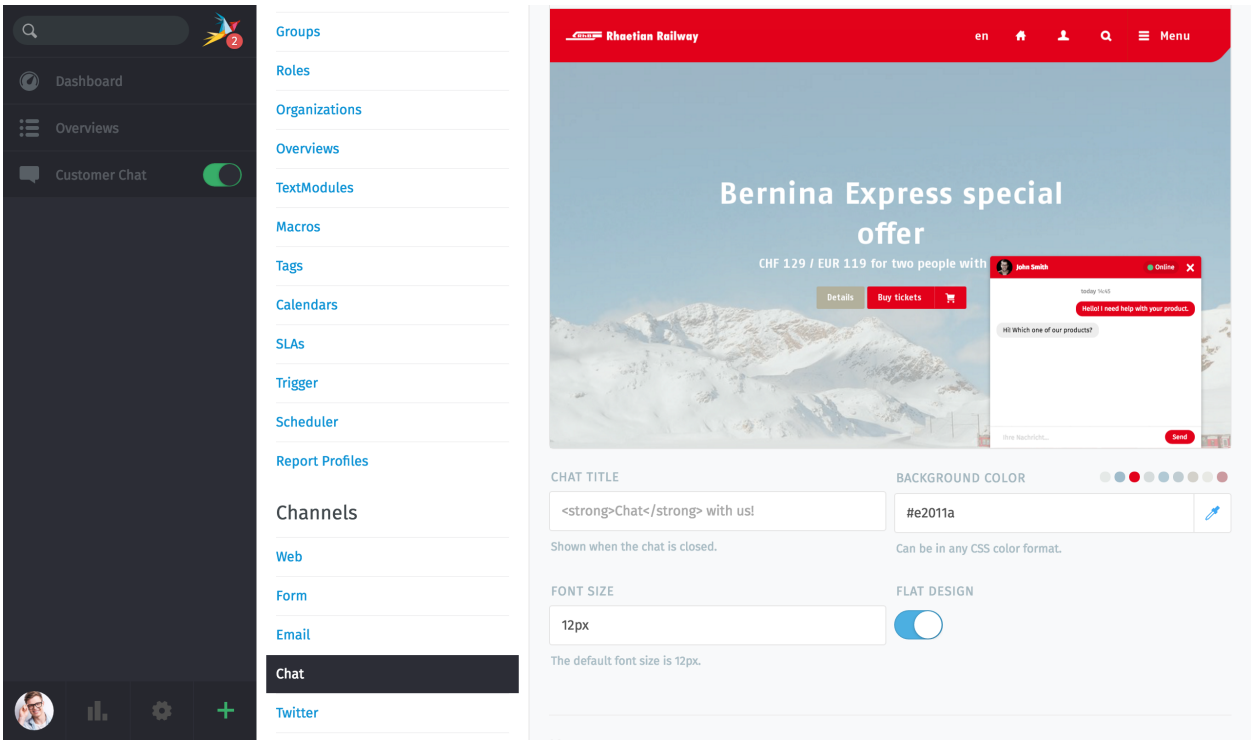
22.3 Configuration of the Chat widget

You can create chat widgets for your web pages to allow visitors to chat with you.

The area for configuring the chat can be found in the admin area at Channels → Chat:

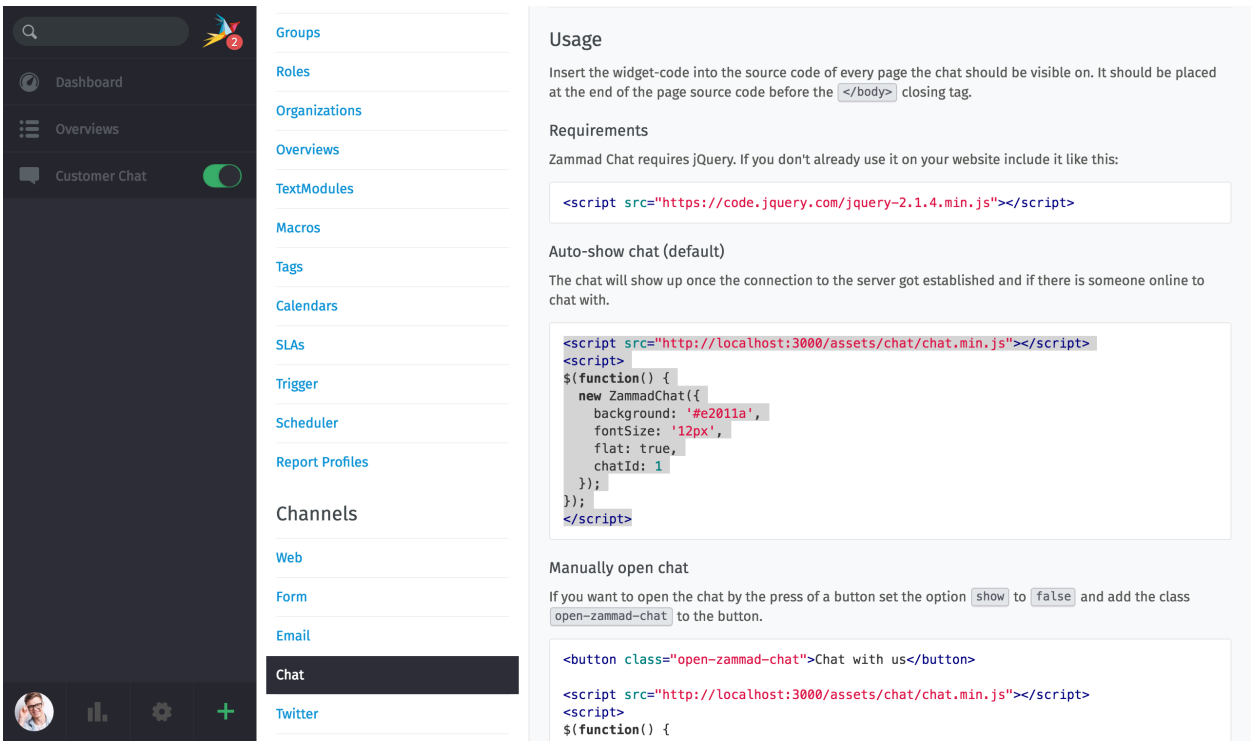


You can set up chats for different websites and edit them independently. The integrated designer helps the chat-widget to adapt to the website color. If you don't like the proposed design, you can manually adjust the design. Through the different previews you have the possibility to display directly how the presentation looks on different devices.



Usage

Insert the widget code into the source code of every page on which you want the chat to be visible on. It should be placed at the end of the page's source code before the </body> closing tag.



Result

The final result will look like the following:

Requirements

Zammad chat requires jQuery. If you don't already use it on your website include it like this:

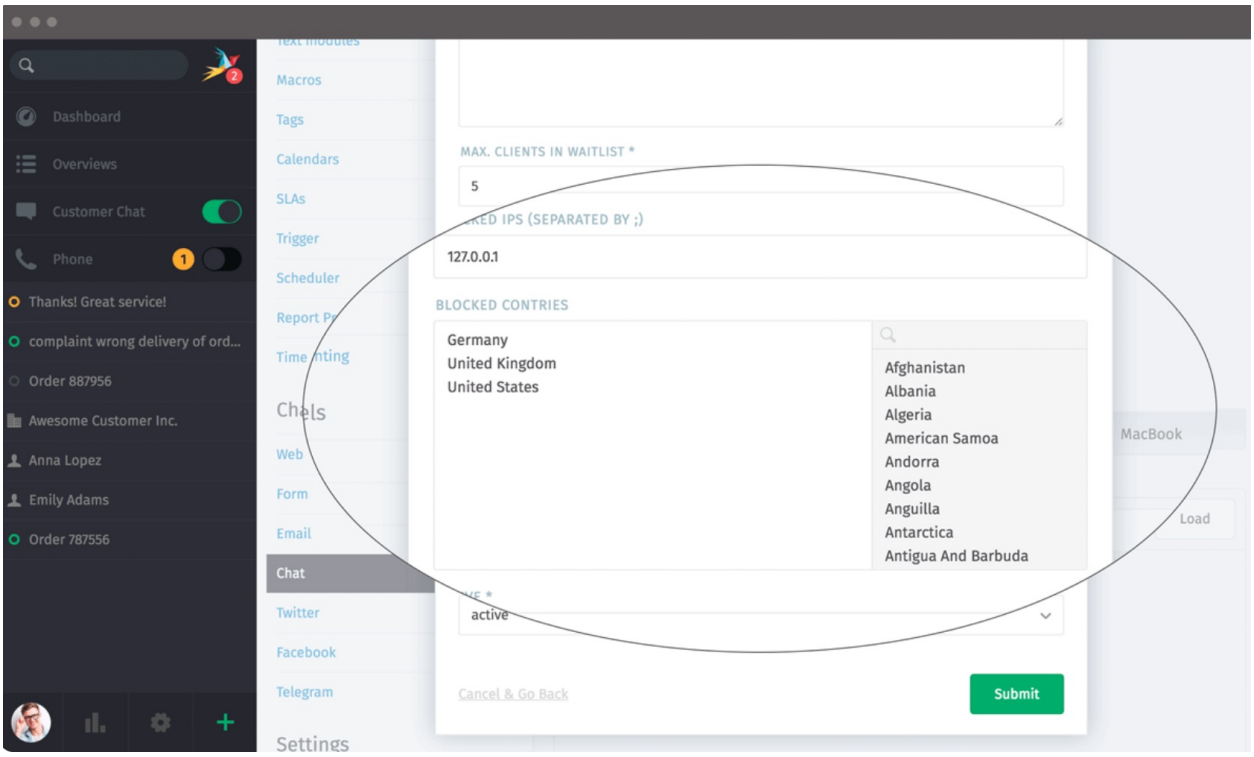
```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

You have two options to implement the chat on your website:

- Automatically show chat (that's the default-setting)
- or manually open chat.

Chat restrictions

You offer a chat for your target group, but you don't want to activate chat for certain IP addresses or countries? Then you have the possibility to block the wished IP addresses and countries fast and easily via the chat configuration in the admin panel. The configuration panel looks like this:



You can also find more information about the chat customization in the admin area.

CHAPTER

TWENTYTHREE

GOOGLE

23.1 Accounts

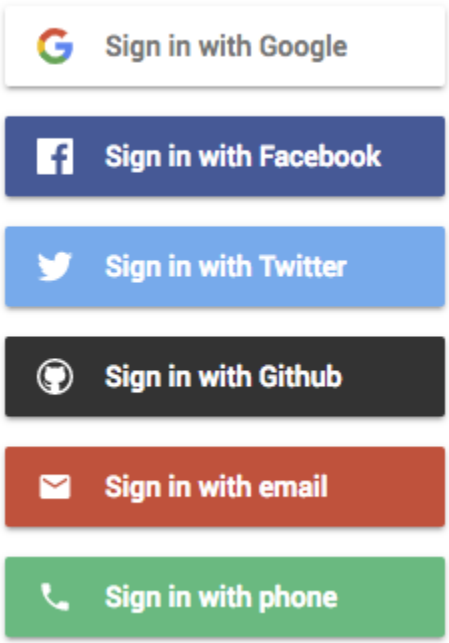
23.1.1 Register an OAuth App

Setting up a new Gmail / G Suite account? Because of Google's strict security policies, it's not as simple as entering your username and password.

First, you'll have to connect Zammad to your Google account as an **OAuth app** via the Google Developer settings panel. Once that's done, you'll be able to connect as many Gmail accounts to Zammad as you want, using only active Gmail browser sessions (no usernames or passwords required).

Note: What the heck is OAuth?

If you've ever used a website that lets you "Sign in with Google/Facebook/Twitter", you've used OAuth. OAuth is a way for you to let a third-party website see a tiny slice of your Google/Facebook/Twitter account data without giving them your password (which would let them see *everything*).



When a third-party website wants to use OAuth, it has to **register with the provider first** (*i.e.,* Google). This way, the provider knows who's receiving its users' data, and can give users a way to revoke access if they change their minds.

In this case, **Zammad is that third-party website**. That's why adding a Gmail account is a two-stage process: first, you have to register Zammad as a website that wishes to access Google user data; then, you have to add yourself as a Google user who agrees to let Zammad fetch your email.

Step-by-Step

To get started, head over to Google's Developer settings panel.

1. **Create a new project**

   For the purposes of this guide, a "project" and an OAuth app are the same thing. You may name it whatever you wish.

2. **Enable & add the Gmail API**

   Use the **Enable APIs and Services** button to start your search.

3. **Set up the OAuth consent screen**

   Configure who can use your app, what kind of access it's asking for, and a few fine print details (like a link to Zammad's privacy policy).

   This information will be displayed in the process of connecting a Gmail account to Zammad, when users are redirected to Google for sign-in/confirmation.

   User Type This option is only available to G Suite users.

   If you have the option, choose **Internal** (unless you plan on creating channels for Gmail addresses from outside your G Suite domain).

   Scopes for Google APIs Add **Gmail API: https://mail.google.com**.

4. **Generate OAuth credentials**

   Click on **Create credentials > OAuth client ID** to begin.

   Application type Choose **Web application**.

   Authorized redirect URIs *E.g.,* `https://your-domain.com/api/v1/external_credentials/google/callback`

   Find it in the Zammad admin panel under **Channels > Google > Connect Google App > Your callback URL**.

5. **Connect your Google app in Zammad**

   Copy your new OAuth app's credentials (**client ID** and **client secret**) into Zammad in the admin panel, under **Channels > Google > Connect Google App**.

Congratulations! Now you're ready to connect Gmail accounts to Zammad.

Troubleshooting

My OAuth credentials stopped working all of a sudden Did you recently reset your Google password?

(Google invalidates all your OAuth tokens whenever you change your password. Generate a new one per Steps 4 and 5 above.)

23.1.2 Account Setup

After you've *registered Zammad as an OAuth app* in your Google Developer settings, you can begin connecting Gmail accounts to Zammad.

But first, a word of warning! The import process does things you might not expect:

> **Danger: Zammad will delete all emails in your inbox during the import process.**
>
> Use the *Keep Messages on Server* setting to *disable this behavior*.

> **Warning: Zammad will send an auto-reply message to every email it imports.** (Including the old ones!)
>
> Make sure to *disable this behavior* prior adding an email account, and to turn it back on once all your messages have been imported.

Migrate an Existing Email Channel

If you've already added your Google account as a regular email channel, you'll have to convert it to a Google channel eventually: Google is planning to end support for simple password authentication in third-party email clients (like Zammad).

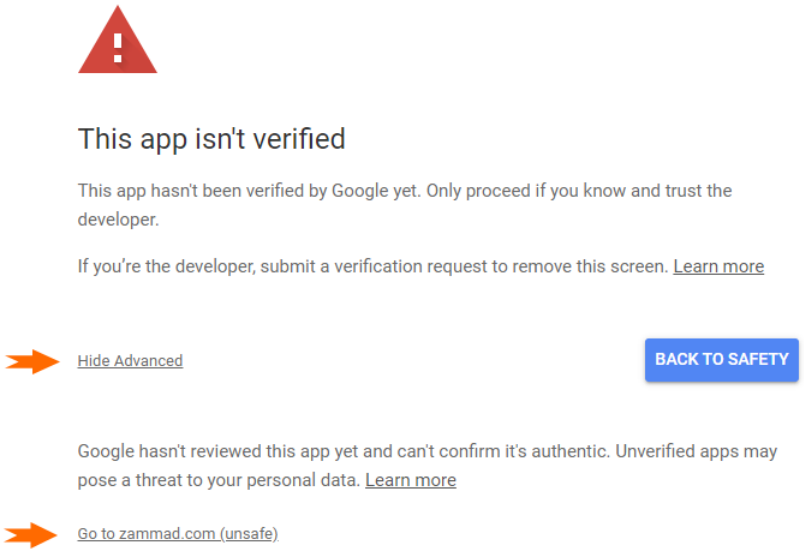
Please refer the *Migrate from Email channel to Google channel* guide.

Add a New Account

Click **Add Account** to connect your Gmail / G Suite accounts to Zammad. You will be redirected to a Google sign-in and confirmation page.

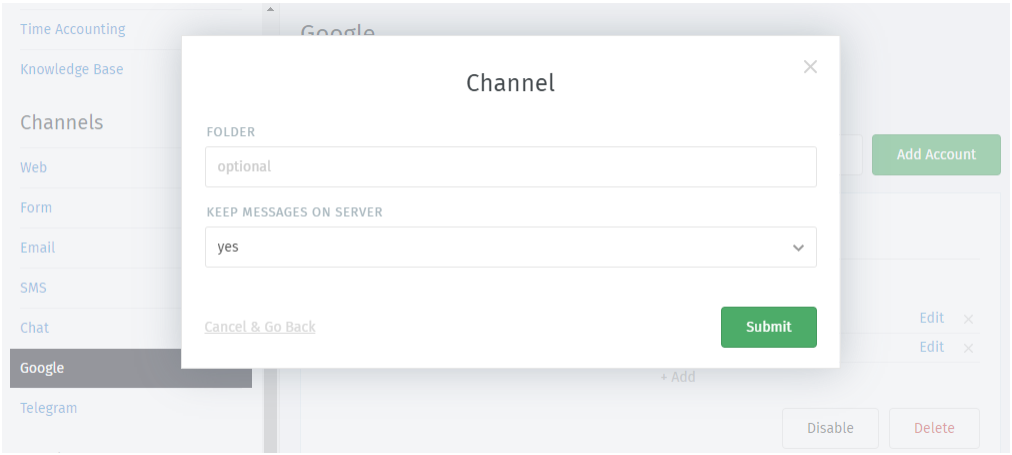
Any aliases registered in your Gmail settings will be imported automatically.

Note:



Google has a stringent verification process to protect users from third-party websites that use OAuth to access their data. Since you *are* the third-party website here, you can safely ignore this warning.

Channel



Folder Specify which folder (or *label*) to fetch from, or leave empty to fetch from `INBOX`.

If specifying a nested folder, be sure to use the full path; *e.g.,* `Inquiries/Tech-Support`.

Keep messages on server Specify what happens to your emails after Zammad imports them:

- `no` Zammad deletes all imported messages
- `yes` Zammad marks imported messages as read

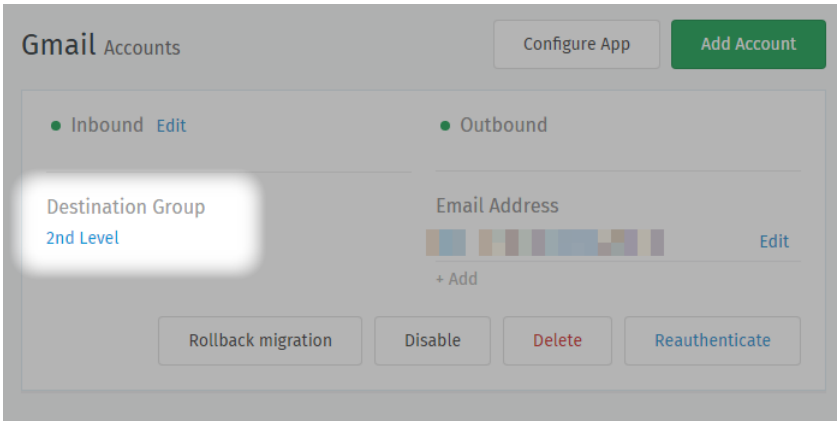
  (With this option, Zammad will only import unread messages. This means Zammad may miss messages if the mailbox is externally modified.)

Note: Why does Zammad delete messages by default?

If you never clean out your inbox, it'll eventually reach its storage limit, and your mail server will start **rejecting incoming messages**. Most Zammad users never even look at their inbox once it's set up, so they rely on Zammad to keep it clean for them.

If you choose **yes** here, remember that it's your responsibility to clean out your inbox from time to time to keep it below its storage limit.

After adding the account After successfully adding the Microsoft 365 mail account, you can adjust the default group Zammad is going to assign incoming new tickets to.



Only **active** groups will be displayed.

Changing this setting will not reassign existing tickets to the new group.

Hint: Don't forget to set the outgoing email address

In Zammad each group decides about the email address that's used for outgoing emails. The incoming group technically has no effect on it.

For this reason make sure to also adjust each affected *groups setting*.

Troubleshooting

I successfully added my account, but Zammad isn't fetching new email If you specified a *custom folder/label to fetch from*, are you sure incoming mail is arriving in that folder?

23.1.3 Migrate from Email channel to Google channel

Zammad provides a migration logic that allows you to migrate existing Google accounts from the Email channel to the Google channel.

Note: Zammad is expecting specific settings

In order for Zammad to display the migration option, it expects the channels hostname to be `imap.gmail.com` for IMAP and `smtp.gmail.com` for SMTP.

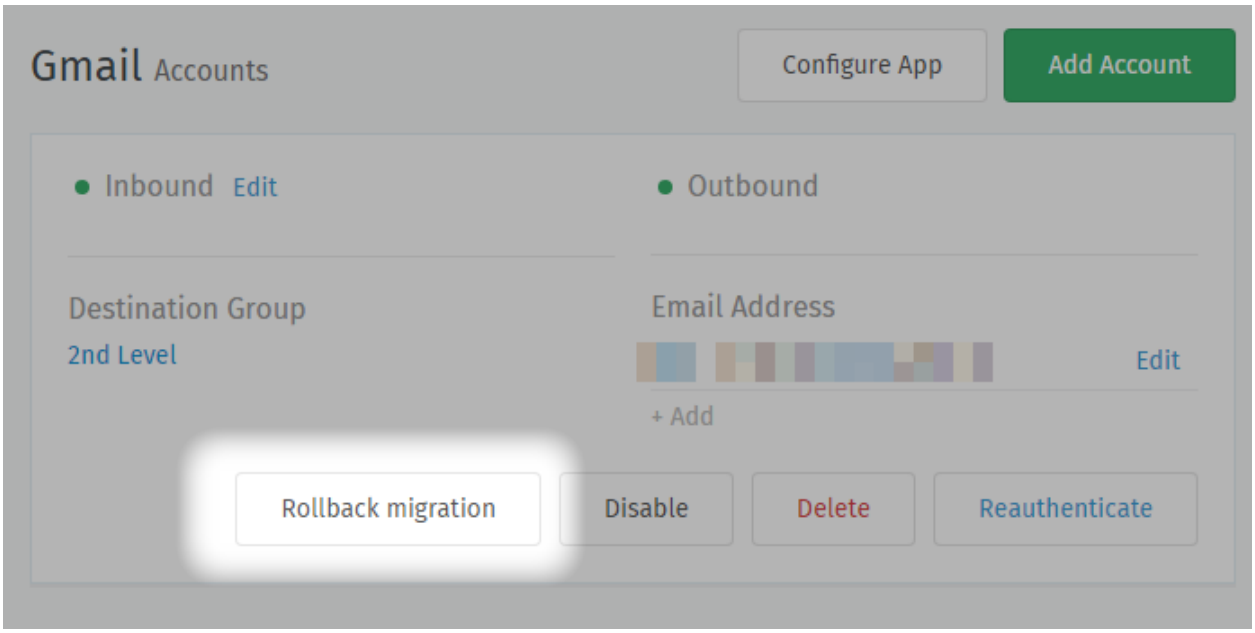
The easiest way to start the migration is to *Register an OAuth App* for your Google accounts before migrating. However, if you don't, Zammad will ask you to provide your app credentials before allowing you to continue.

If you're ready to go, simply click on the *Migrate now!* button in the red banner of the email channel in question. Zammad will redirect you to Google and request you to authenticate and consent to said account.

After you pressed next you'll be redirect to Zammad's Google channel overview. Your channel, if successful, is now migrated to an Google channel.

Rolling back the migration

In case something went wrong, Zammad allows you to roll back the migration for up to 7 days. For this time period Zammad will remember your original credentials and restore it if needed. These information will be removed entirely after 7 days.



Note: Secondary addresses in Google channels work (almost) **just like they do in email channels**, so this article is lifted (almost) verbatim from *here*.

23.1.4 Secondary Addresses

Secondary addresses (also known as *aliases*) allow you to send emails with a different "From:" address from the one on the account.

Fig. 1: Once you add a secondary address, you can configure a group to start sending emails with it.

> **Warning:** **Secondary addresses must be added and verified in your Gmail settings first.**
>
> Gmail has its own process for adding and verifying aliases (under **Settings > Accounts and Import > Send mail as**). If you add an alias here before adding it in your Gmail settings, Google will refuse to dispatch it.
>
> G Suite users may need to contact their administrators in order to add aliases in their Gmail settings.

Your email provider may also be set up to *receive incoming messages* for many addresses in the same mailbox. If this is the case, be sure to add your alternate inbox addresses here.

Display Name The display name used for outgoing email.

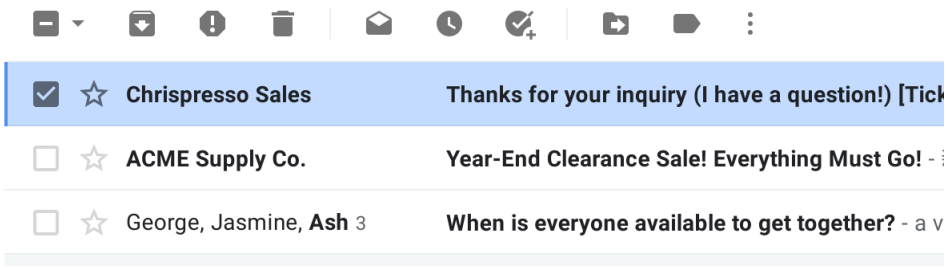


Fig. 2: A customer's inbox with an auto-reply from **Chrispresso Sales**.

Email display names value can be *further customized in the Settings tab*.

Email The alias address to send outgoing messages as.

Channel The email account to be used when sending outgoing messages from this alias.

Note Optional. Only visible from this dialog, the REST API, and the Rails console.

Note: Managing accounts in Google channels is (almost) **just like it is in email channels**, so this article is lifted (almost) verbatim from *here*.

23.1.5 Managing Accounts

Once an account has been added, use the Accounts panel to edit its configuration.

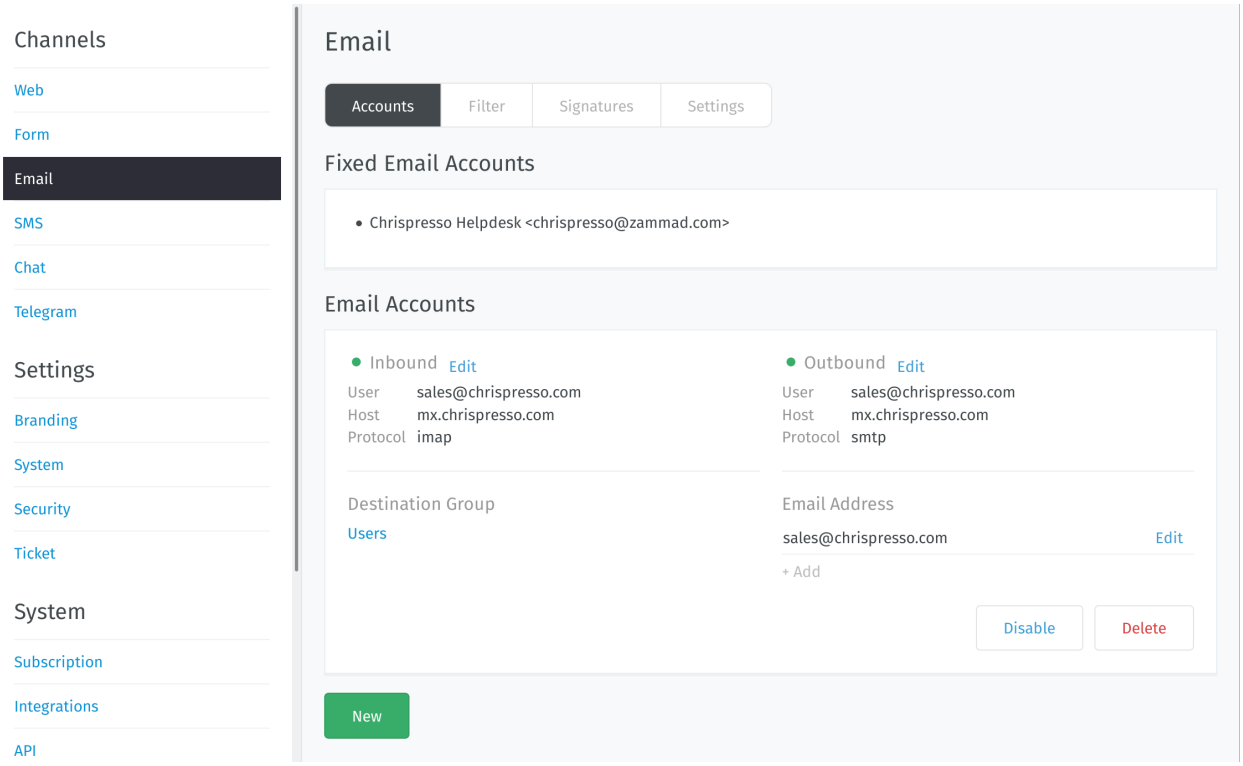
Fetch Preferences Click **Edit** on inbound account details to change how messages are retrieved from your account.

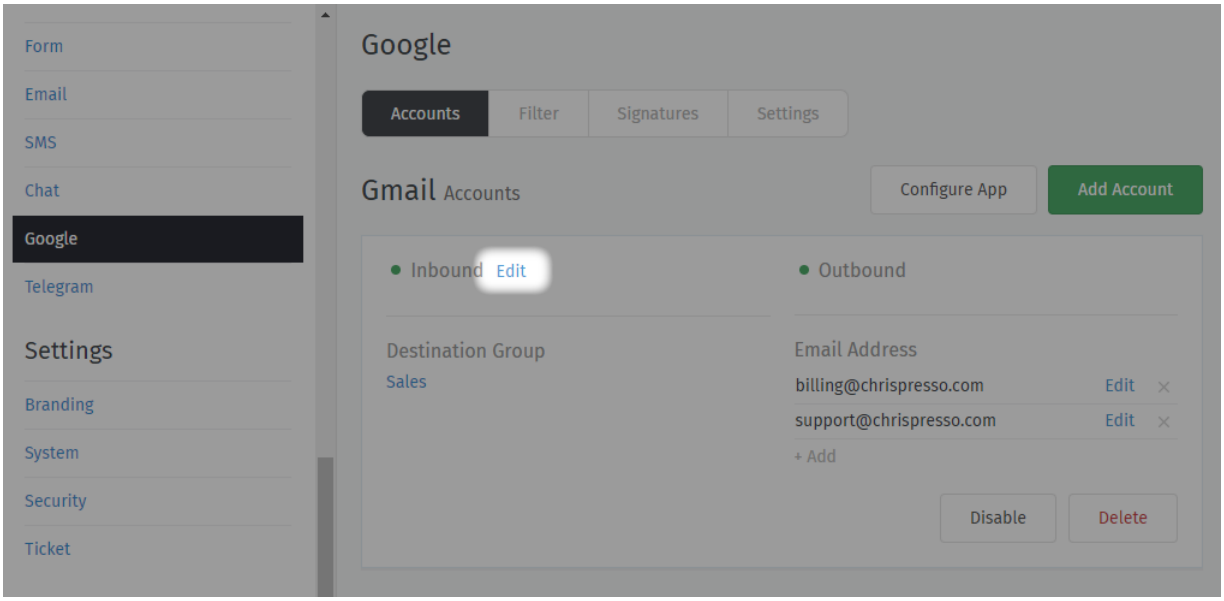
See *New Account Settings* for a detailed description of each option.

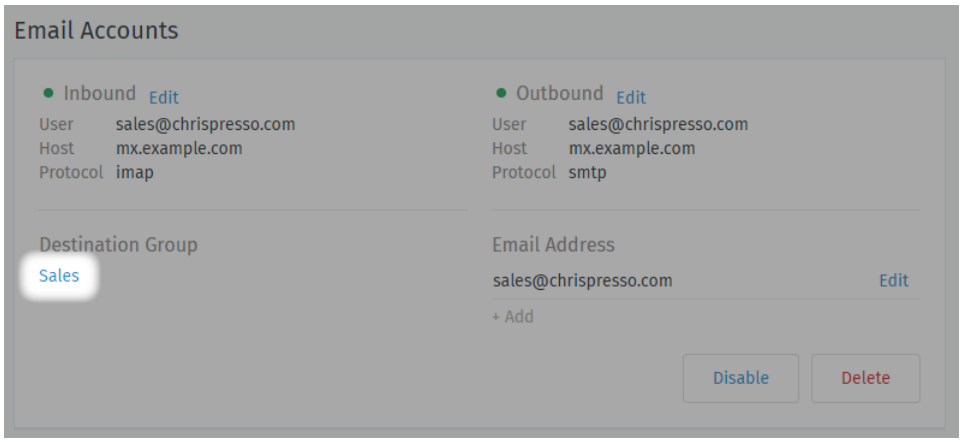
Destination Group Click on the group name to reassign the account.

Only **active** groups will be displayed.

Changing this setting will not reassign existing tickets to the new group.


Channels
Web
Form
Email
SMS
Chat
Telegram
Settings
Branding
System
Security
Ticket
System
Subscription
Integrations
API
Email
Accounts
Filter
Signatures
Settings
Fixed Email Accounts
Chrispresso Helpdesk <chrispresso@zammad.com>
Email Accounts
Inbound Edit
User sales@chrispresso.com
Host mx.chrispresso.com
Protocol imap
Outbound Edit
User sales@chrispresso.com
Host mx.chrispresso.com
Protocol smtp
Destination Group
Users
Email Address
sales@chrispresso.com Edit
+ Add
Disable
Delete
New

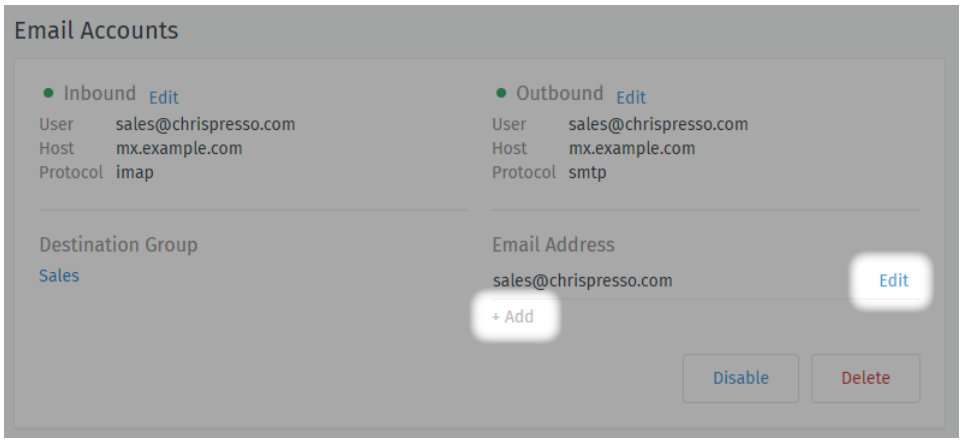

Form
Email
SMS
Chat
Google
Telegram
Settings
Branding
System
Security
Ticket
Google
Accounts
Filter
Signatures
Settings
Gmail Accounts
Configure App
Add Account
Inbound Edit
Outbound
Destination Group
Sales
Email Address
billing@chrispresso.com Edit
support@chrispresso.com Edit
+ Add
Disable
Delete



Hint: **Still can't send outgoing email tickets?** Check your *group settings*.

Email Address Use the **+ Add** or **Edit** buttons to set up secondary addresses on this account.

See *Secondary Addresses* for a detailed description of each option.



Enabled / Disabled Disabling an account temporarily prevents Zammad from importing its messages.

This may be necessary during scheduled maintenance or when migrating your installation to a new host.

Note: Disabling an account disables *outgoing* messages for it, as well.

Delete Deleting an account removes its configuration from Zammad entirely.

Note: **Additional Steps Required**

Groups need an assigned an address to send outgoing emails. If you delete a group's assigned address, agents belonging to that group won't be able to send emails until you assign it a new one.

(There's no need to manage orphaned email addresses like you would on an email channel. In Google channels, aliases are connected to your Gmail account, which means they can be imported and purged automatically.)

Register an OAuth App Use the **Connect Google App** dialog to register Zammad as an OAuth app on Google.

(This step is **required**; read on to learn why.)

Account Setup Use the **Add Account** dialog to connect your account.

You're migrating existing email channels? Look below!

Migrate from Email channel to Google channel Use the *Migrate now!* button within your email channels to quickly move your mailboxes to Microsoft 365. You can roll back if things hit the fan!

Secondary Addresses Send and receive email at **additional email addresses**, all through the same mailbox/account.

Managing Accounts Edit the configuration of existing accounts in the **Accounts Panel**.

Note: How do I use my Gmail account for outgoing system notifications?

On **subscription/cloud-hosted instances**, you can't. Notifications will always come from "Notification Master <noreply@your.zammad.domain>".

On **self-hosted instances**, we still don't recommend it. Using a Gmail account for automated, outgoing messages is risky: users who exceed Google's email sending limits can have their accounts suspended.

Set up a generic *email channel* instead, the use the *Email Notification* setting.

Note: Filters in Google channels are **just like filters in email channels**, so this article is lifted verbatim from *here*.

23.2 Filters

Postmaster filters allow you to match email headers (e.g. `From`, `To`, `Subject`, `X-Spam-Flag` etc.) and execute a set of actions whenever Zammad's email parser encounters a matching email. The actions will be applied to the ticket that is created or updated by this email. Here are some examples of what is possible with filters:

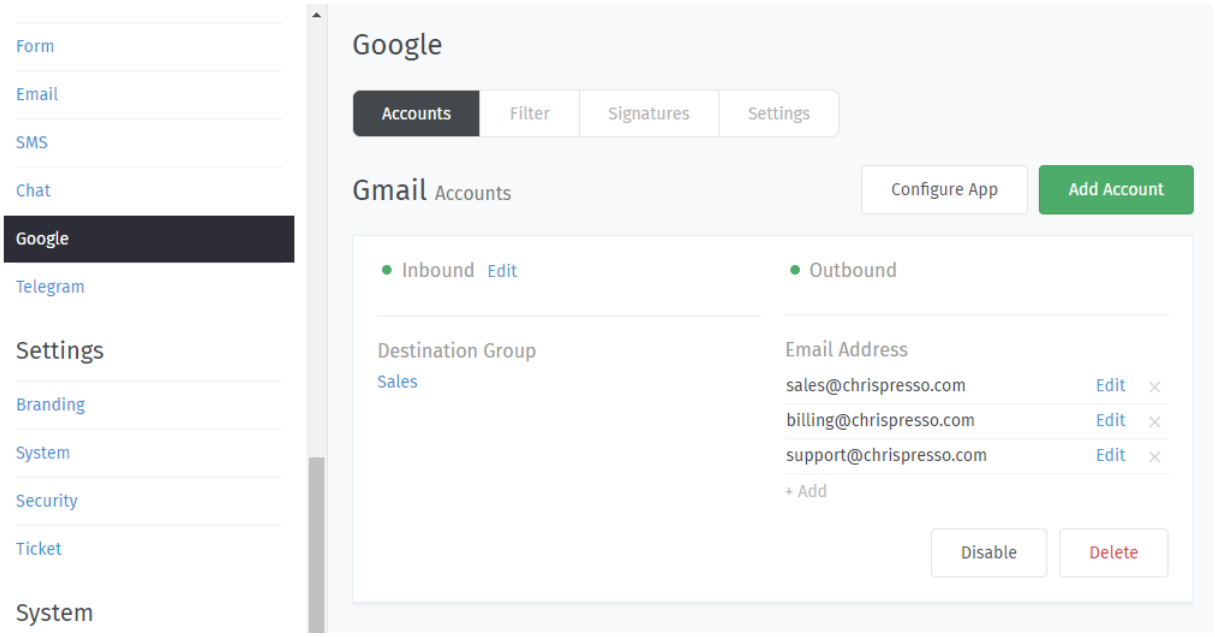
Automatically dispatch tickets into certain groups: For example, tickets from `amazon.com` could automatically be dispatched to the Purchasing group.

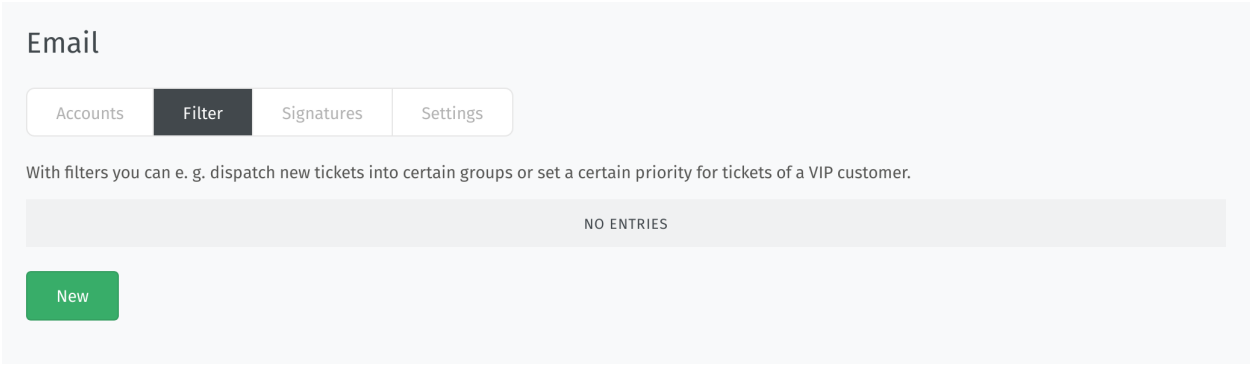
From: contains: `regex:(\.|@)amazon\.com`
Group: Purchasing

Note: Note that the Group action only has an effect when the matching email results in a new ticket. Zammad will not change the group of existing tickets.

Automatically increase the priority of tickets from a VIP customer:

From: contains: `ourvipcustomer@example.com`
Priority: 3 high


Form
Email
SMS
Chat
Google
Telegram
Settings
Branding
System
Security
Ticket
System
Google
Accounts
Filter
Signatures
Settings
Gmail Accounts
Configure App
Add Account
Inbound Edit
Outbound
Destination Group
Sales
Email Address
sales@chrispresso.com Edit
billing@chrispresso.com Edit
support@chrispresso.com Edit
+ Add
Disable
Delete


Email
Accounts
Filter
Signatures
Settings
With filters you can e. g. dispatch new tickets into certain groups or set a certain priority for tickets of a VIP customer.
NO ENTRIES
New

Note: Note that the Priority action only has an effect when the matching email results in a new ticket. Zammad will not change the priority of existing tickets.

Automatically tag and close spam tickets that have been marked as spam by anexternal spam filter (e.g. SpamAssassin):

X-Spam-Flag: contains: `YES`
Tag: add: `spam`
State: closed

Note: Note that the State action only has an effect when the matching email results in a new ticket. Zammad will not change the state of existing tickets. It will add the tag though if it missing, even if the mail is an update to an existing ticket.

The following actions are only effective when creating tickets:

- Group
- State
- Priority
- Owner

Different attributes of a filter can be combined with each other. Likewise, the following actions can be combined. The supported matches are "contains" and "contains not"; for advanced matching, you can use regular expressions by prefixing the string with `regex:`.

Note that Zammad matches against the full header, e.g. for a mail with `From: Display Name <display.name@example.com>`, the `From` condition will test against `Display Name <display.name@example.com>`. This is especially important when using anchored regular expressions; `regex:^display\.name@example.com$` would not match this mail!

It should be borne in mind that the combined attributes build on each other. If a filter is no longer needed, it can either be temporarily set inactive or deleted directly.

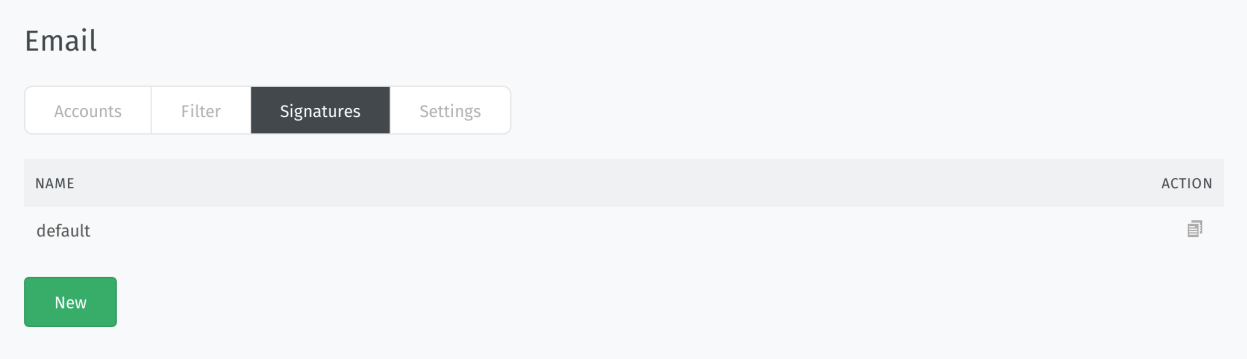
Note: Signatures in Google channels are **just like signatures in email channels**, so this article is lifted verbatim from *here*.

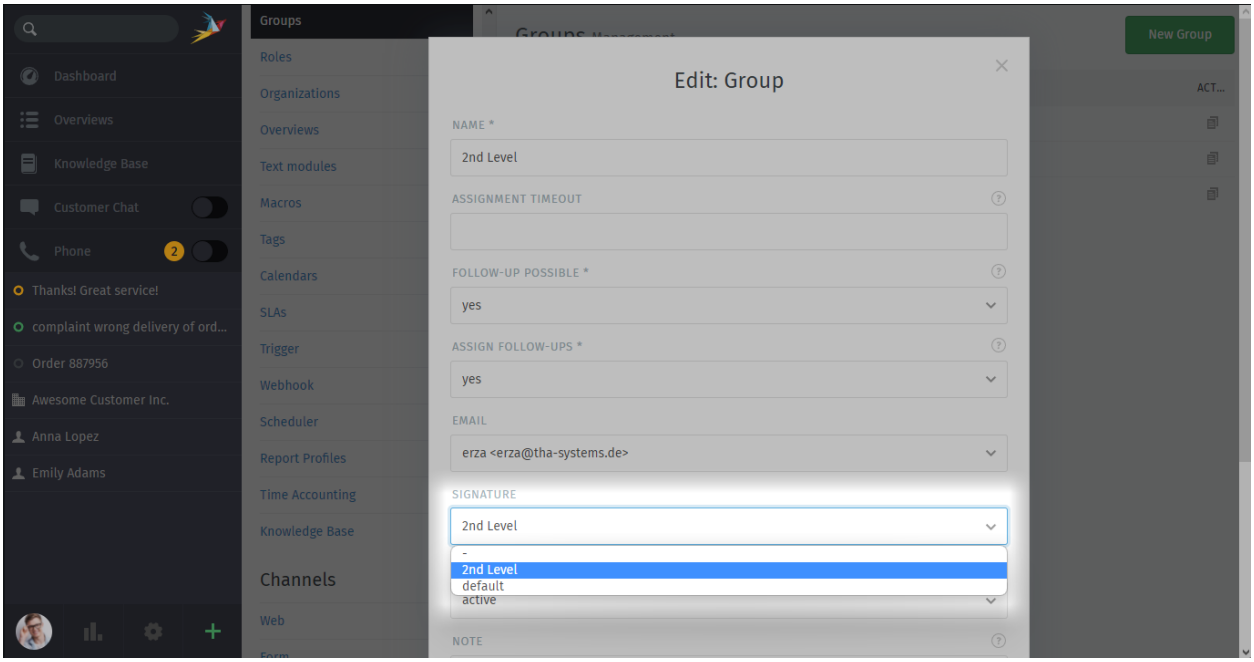
23.3 Signatures

You can create a separate signature for each group in Zammad. The individual signatures can be created and edited here.

Afterwards, the existing (and active) signatures are available in the group editing mask:

Each group can be assigned its own signature, but they can also all use the same signature.


Email
Accounts
Filter
Signatures
Settings
NAME
ACTION
default
New

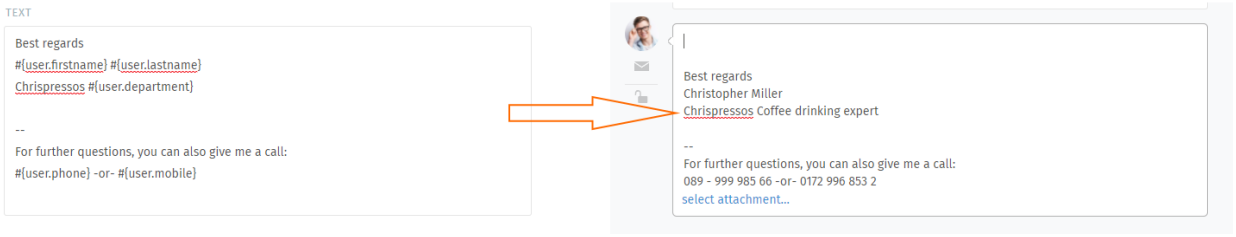

Groups
Roles
Organizations
Overviews
Text modules
Macros
Tags
Calendars
SLAs
Trigger
Webhook
Scheduler
Report Profiles
Time Accounting
Knowledge Base
Channels
Web
Dashboard
Overviews
Knowledge Base
Customer Chat
Phone
Thanks! Great service!
complaint wrong delivery of ord...
Order 887956
Awesome Customer Inc.
Anna Lopez
Emily Adams
New Group
Edit: Group
NAME *
2nd Level
ASSIGNMENT TIMEOUT
FOLLOW-UP POSSIBLE *
yes
ASSIGN FOLLOW-UPS *
yes
EMAIL
erza <erza@tha-systems.de>
SIGNATURE
2nd Level
-
2nd Level
default
active
NOTE

23.3.1 Dynamic Signatures

To individualize the signatures, it is possible to automatically load specific information into a signature via *Variables*. All information stored on the ticket, assigned customers or agents can be inserted. This makes it possible to design the signature individually. To load a list of available variables, enter two colons (`::`) into the *Text* box of the signature editor.

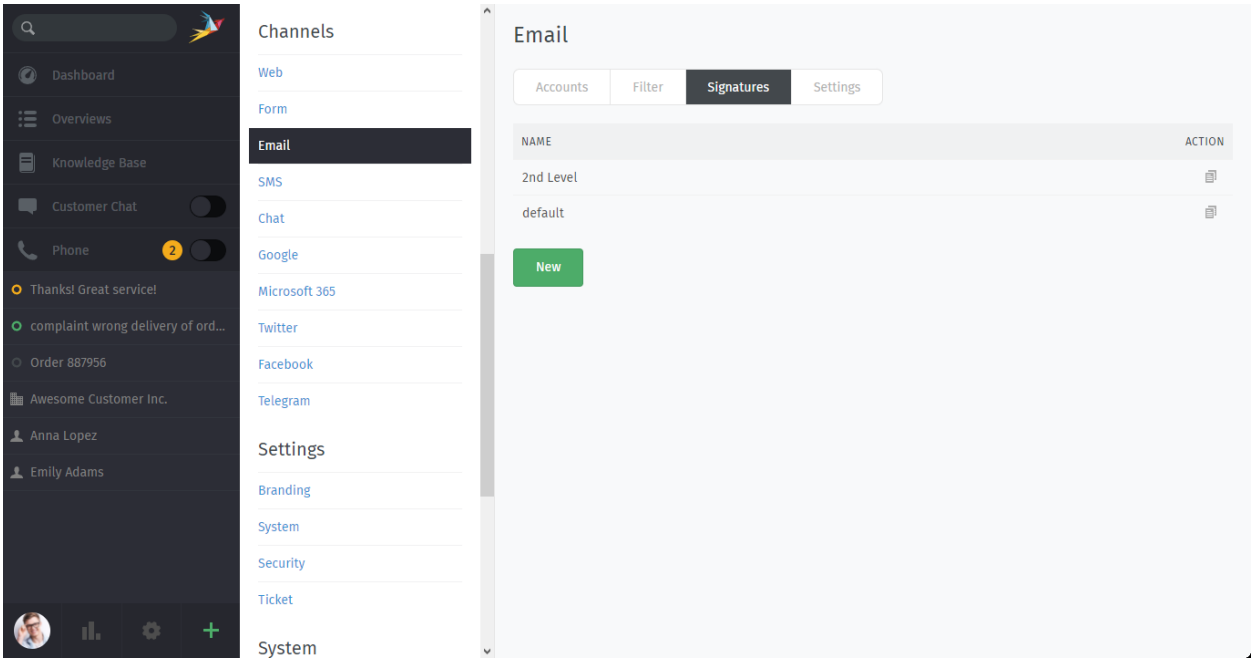
Hint: Please keep in mind that specific information might not be available during ticket creation. The best example here is the ticket number / id. Specific information are created with submitting the ticket and thus are not available before submitting.

Here is an example of a signature with variables and the result when you write a mail:



Note: Settings in Google channels are **just like settings in email channels**, so this article is lifted verbatim from *here*.

23.4 Settings



Below you can find the currently available email-related settings. Most of these settings have default values which can be found in this list as well.

Note: Some email-related settings are ticket-based settings, which is why they can be found in the *Composer Settings*.

23.4.1 List of Settings

Notification Sender: Default value `Notification Master <noreply@#{config.fqdn}>` This is the default sender address for Zammad that affects all mails but those generated because of replies (like triggers or agent-based mails). Your customers normally will not see this address. This email address does not need to receive and can't be assigned to a group.

Note: This address is relevant for agent notifications and password reset mails (also affects customers).

Additional follow-up detection In some situations the normal follow-up detection is not enough. This might be due to missing references in the subject (the ticket hook and number). These options can help to recognize follow-ups to existing tickets.

Note: Please note that searching in attachment and body might lead to false follow-up detection.

Maximum Email Size: Default value `10 MB` This one is pretty obvious: It defines the maximum allowed size of an email Zammad will fetch. Zammad will not fetch Mails that are bigger than this option.

Hint: This technically also affects attachments for articles.

Send postmaster mail if mail too large: Default value `yes (enabled)`

Note: Upgraded installations will, by default, have the value set to `no (disabled)`.

Option set to `yes` This setting will cause Zammad to automatically reply to mails that exceed the above mail size limit with a postmaster style mail. This will help your user to understand that his mail did not arrive and won't be reviewed by you.

Note: Zammad will still download and remove (if enabled) the mail from the mailbox. Instead of importing it to the database, it will save the affected mail to `/opt/zammad/tmp/oversized_mail/`.

Option set to `no` If the option is set to no, Zammad will not reply to mails that are too big. Your customer will **not notice** that the mail was too large! Instead, Zammad will use the monitoring endpoint to alert its administrators that it can't fetch a too large mail.

Learn more about *Monitoring*.

Sender based on Reply-To header: Default value `not set (-)` This setting decides how Zammad should recognize its customers from emails that contain a `Reply-To` header. This comes in useful if you're working with contact forms that need to use reply to headers.

Option set to `Take reply-to header as sender/from of email.` This setting will overwrite the initial `FROM` to the value used in `Reply-To` completely.

Option set to - or `Take Reply-To header as sender/from of email and use the real name of origin from.` This setting will partially overwrite the initial `FROM`. It uses the mail address from the `Reply-To` header and uses the given name of the `FROM` header, if given.

Customer selection based on sender and receiver list: Default value `yes` This option decides how Zammad should react if an agent sends a email to it.

Option set to `yes` The first user / email address from the recipient list will be used as the ticket customer.

Option set to `no` The agent will be set as ticket customer.

Note: Currently agents can't be customers within the UI. While Email communication works, agents can't see their own tickets (as a customer) if they don't have access to the group.

Block Notifications With the regex that can be defined here, you can ensure not to send any notifications to specific systems. By default this especially affects typical system addresses which can't receive emails anyway.

The default value is: `(mailer-daemon|postmaster|abuse|root|noreply|noreply.+?|no-reply|no-reply.+?)@.+?`

Sender Format: Default value `Agent Name + FromSeparator + System Address Display Name` This configures the display name used in the `FROM` header of mails Zammad sends.

Note: This does not affect Notification mails (to agents) and password reset mails. Emails that are not sent by agents (e.g. trigger-based notifications) will always fallback to `System Address Display Name` if needed.

Option set to `Agent Name + FromSeparator + System Address Display Name` This will cause Zammad to set the `FROM` header to agent name and the channel's display name, divided by a separator (configured below).

Example: `Christopher Miller via Chrispresso Inc.`.

Option set to `System Address Display Name` This will cause Zammad to always use the display name of the used channel in the `FROM` header.

Example: `Chrispresso Inc.`

Option set to `Agent Name` Zammad will use the agent's name which is very personal.

Tip:

Usually you'd also want to remove the ticket slug from the subject in those cases.
Learn more in *Settings → Ticket*.

Sender Format Separator: Default value `via` This is a can be a string you can freely choose. It divides the agent's name and the display name of the channel whenever needed.

Ticket Subject Forward: Default value `FWD` The above string will be used on the subject if you forward an email from Zammad.

Note: `:` will be automatically appended to the above string.

Ticket Subject Reply: Default value `RE` The above string will be used on the subject if you reply to a mail from Zammad.

Note: : will be automatically appended to the above string.

Ticket Subject Size: Default value `110` This setting enforces a maximum length for subjects when replying. If the subject you're using for your reply is too long, Zammad will automatically truncate the length and insert `[...]` to show it has shortened the subject.

Example: `RE: Test somew[...] [Ticket#123456]`

Note: This does **not** limit ticket titles within the UI, just the subjects when replying to an email.

23.4.2 Enhanced settings

Some less relevant settings can be changed via rails console if needed. As an example, Zammad allows you to send all outgoing communication to a BCC address for archiving reasons if needed. You can find the needed commands within the advanced customization settings.

Note: EMail header manipulation in Google channels work **just like in email channels**, so this article is lifted verbatim from *here*.

23.5 Email header manipulation

Email header manipulation allows you to re-route or adjust tickets apart from filters or triggers. Like an API call, but with emails.

Header checks are case insensitive.

Warning: Trusted channels required

Below options are a potential risk with external communication and thus require channels being set to trusted explicitly.

Tip: Below headers are examples and –in our opinion– the most relevant ones. However: You can adjust mostly any article or ticket attribute (yes, custom ones as well) if you know the attribute's exact name.

The name column within *object's management* provides easy access to objects attribute names.

23.5.1 Trigger auto responses

Normally Zammad runs internal checks to see if an email is an automatic response. In these cases Zammad will not send trigger based responses.

There may be use cases where this behavior may be in your way, below options allow you to overcome this issue.

Note: In some cases combining below headers is crucial. This is intentional but may be confusing.

`x-zammad-send-auto-response` Set to `false` to disable trigger based responses. If set to `true` Zammad will send a response.

Hint: This option *does not* work if e.g. `precedence: list` is set unless you use below auto response header as well.

`x-zammad-is-auto-response` Providing this header allows you to tell Zammad that the mail in question is an auto generated response (`true`). This will cause email based triggers to be skipped.

Set this header to `false` if you want to generate auto responses.

Tip: This header allows you to overwrite auto detects for e.g. `precedence: list`.

23.5.2 Ticket attributes

Zammad allows you to use headers to manipulate ticket creations or follow ups. The manipulation can be used instead of triggers. Triggers are considered *after* header settings and thus can still overrule.

Note: Zammad differentiates between ticket creation and follow up

For creations use: `X-Zammad-Ticket-{Attribute Name}`
For follow ups use: `X-Zammad-Ticket-FollowUp-{Attribute Name}`

This allows you to ensure the changes are only applied in the required situation.

Warning: About values

- While headers are not case sensitive, values like e.g. priority names are case censitive: `1 low` will work, but `1 lOw` will not!
- When using attributes that require date / time values, ensure to use Time Zoned Times. e.g. for 28th September 2021 on 8 am CEST, either use:
  - `2021-09-28T08:00:00+0200`
  - `2021-09-28T08:00:00+02:00`
  - `2021-09-28T06:00:00.000Z`

`X-Zammad-Ticket-Priority` & `X-Zammad-Ticket-FollowUp-Priority`

Allows you to adjust a ticket's priority.
Example: `X-Zammad-Ticket-Priority: 1 low`

X-Zammad-Ticket-Group & X-Zammad-Ticket-FollowUp-Group

Allows you interfere with regular channel routing of the ticket.
Example: `X-Zammad-Ticket-Group: Sales`

X-Zammad-Ticket-Owner & X-Zammad-Ticket-FollowUp-Owner

Directly assign or change the ticket owner. Valid values are either `login` or `Email`
Example: `X-Zammad-Ticket-Owner: jdoe`

X-Zammad-Ticket-State & X-Zammad-Ticket-FollowUp-State

Set a specific ticket state.
Example: `X-Zammad-Ticket-State: closed`

Note:

Pending states always require the `pending_time` attribute on top.
Like so: `X-Zammad-Ticket-Pending_Time: 2021-09-26T08:00:00+0200`

X-Zammad-Customer-Email

Manipulate the ticket customer - this can be a different user than the actual sender. Replying to the original sender is still possible.
Example: `X-Zammad-Customer-Email: jdoe@example.com`

Note: This header is not available for follow ups.

X-Zammad-Customer-Login

Manipulate the ticket customer - this can be a different user than the actual sender. Replying to the original sender is still possible.
Example: `X-Zammad-Customer-Login: jdoe`

Note: This header is not available for follow ups.

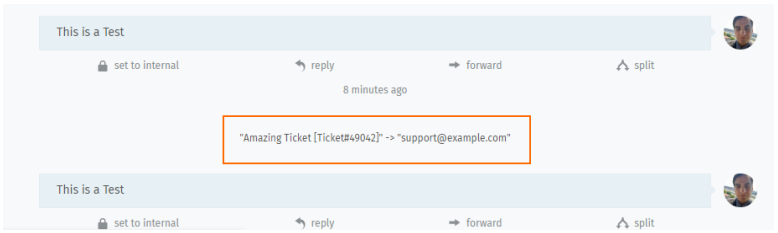
23.5.3 Article attributes

If needed Zammad allows you to manipulate attributes or states of fetched email articles.

X-Zammad-Article-Sender

Manipulate the sender type (Agent, Customer, or System)
Example: `X-Zammad-Article-Sender: System`

Warning: System Emails are indicated in a similar way as trigger-response like entries Users can't see them natively.



`X-Zammad-Article-Type`

Change the article type of your incoming mail. This requires you to know which article types are available in your system.

Example: `X-Zammad-Article-Type: phone`

> **Warning:** This header can cause *serious issues* in your instance and may lead to unexpected behavior. Only use with absolute care!

`X-Zammad-Article-Internal`

Manipulate the default article visibility.

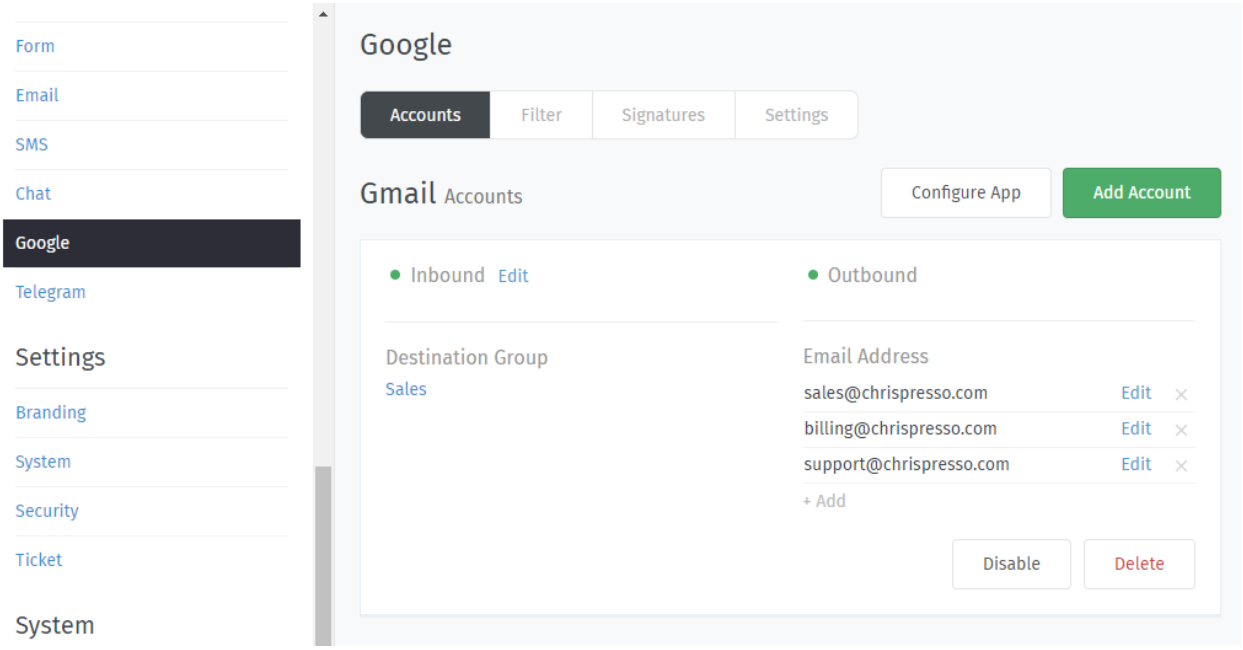
Example: `X-Zammad-Article-Internal: true`

`X-Zammad-Ignore`

Tell Zammad to silently drop the Email.

Example: `X-Zammad-Ignore: true`

Connect a Gmail or G Suite account to Zammad.



Note: Google channels are a specialized kind of *email channel*.

If you're already familiar with email channels, you can skip most of this—but **watch out for the "Accounts" section**, which has a few extra quirks due to Google's strict security measures.

Accounts **Connect Zammad to your email provider** so that it can watch your inbox, send auto-replies, and more.

(Self-hosted users may have already completed this step during new system setup.)

Filters **Make sure new tickets show up in the right place** with automated, if-this-then-that rules for all incoming email.

Signatures Customize signatures for all outgoing email.

Settings Manage options like:

- set the "From:" address on system notifications
- raise the limit on attachment sizes
- modify subject-line prefixes (*e.g.,* use "AW:" instead of "RE:")

Hint: Want to **manually edit email subjects** or **always copy parent messages into your replies**?

Check the *Composer Settings*.

Header manipulation Manipulate auto response behavior or incoming routing.

Warning: This is a very advanced topic.

CHAPTER

TWENTYFOUR

MICROSOFT 365

24.1 Accounts

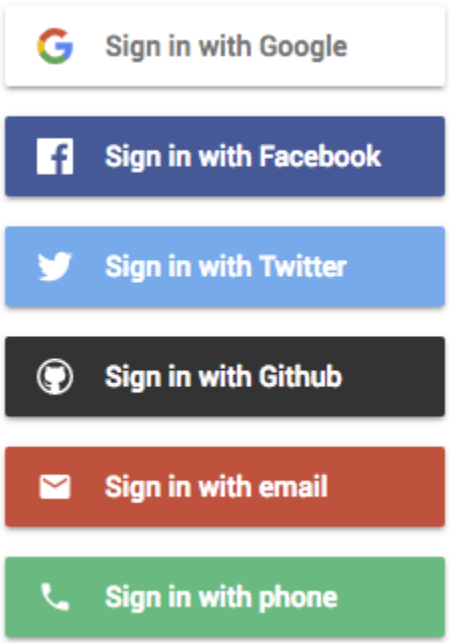
24.1.1 Register an OAuth App

Setting up a new Microsoft365 / Outlook account? Because of Microsoft's strict security policies, it's not as simple as entering your username and password.

First, you'll have to connect Zammad to your Microsoft account as an **OAuth app** via the Microsoft Azure Portal. Once that's done, you'll be able to connect as many Microsoft 365 accounts to Zammad as you want, using only active Microsoft 365 browser sessions (no usernames or passwords required).

Note: What the heck is OAuth?

If you've ever used a website that lets you "Sign in with Google/Facebook/Twitter", you've used OAuth. OAuth is a way for you to let a third-party website see a tiny slice of your Microsoft/Facebook/Twitter account data without giving them your password (which would let them see *everything*).



When a third-party website wants to use OAuth, it has to **register with the provider first** (*i.e.,* Microsoft). This way, the provider knows who's receiving its users' data, and can give users a way to revoke access if they change their minds.

In this case, **Zammad is that third-party website**. That's why adding a Microsoft account is a two-stage process: first, you have to register Zammad as a website that wishes to access Microsoft user data; then, you have to add yourself as a Microsoft user who agrees to let Zammad fetch your email.

Step-by-Step

To get started, head over to Microsoft's Azure Portal.

Note: Use an admin account for your organization.

Otherwise, an admin will have to approve your changes before they can take effect.

1. **Add an App Registration**

   Under **App Registrations > New Registration**, use the following:

   Supported account types Choose the option that's right for your organization (or click *Help me choose…* if you're not sure).

   - Accounts in this organizational directory only (Default Directory only - Single tenant)
   - Accounts in any organizational directory (Any Azure AD directory - Multitenant)
   - Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

   Note: The "Personal Microsoft accounts only" option is not supported.

   Redirect URI **Web** > *E.g.,* `https://your-domain.com/api/v1/external_credentials/microsoft365/callback`

   Find it in the Zammad admin panel under **Channels > Microsoft 365 > Connect Microsoft 365 App > Your callback URL**.

2. **Add API permissions**

   Under **API Permissions > Add a permission > Microsoft Graph > Delegated permissions**, add the following:

   OpenId permissions

   - `email`
   - `offline_access`
   - `openid`
   - `profile`

   IMAP

   - `IMAP.AccessAsUser.all`

   SMTP

   - `SMTP.Send`

3. **Admin Consent related information** (optional)

   Hint: This step is only required for tenants that require admin consent. Admin consent adds another layer of security for your tenants users and allows administrators to define who may share user information.

   Within *Enterprise applications* select your app. When creating an app within *App registrations*, Microsoft will automatically also create an enterprise application for you.

   Set *Assignment required* to `Yes` within the apps properties and hit *Save*. This generally activates your app requesting admin consent already.

   3.1 **Tightening your app even further** (Recommended, optional)

   Note: This step is not exactly required but recommended. This step will protect your tenant users from -to admins- unwanted permission changes (e.g. requesting more permissions than originally noted).

   It also has another benefit: You can add the email account in question immediately without any administrator being forced to stepped in manually.

   Still within *Enterprise applications* select *Users and groups*. In this section you can select specific users and / or groups (users must be direct members!) that are allowed to use your app for adding mailboxes to Zammad.

   After adding users and groups, go back to the Azure portals home and select *App registrations*. Within your desired app go to *API permissions* and use the *Grant admin consent for {company name}* button to generally allow connecting users you previously consented.

   Note: Not adding users / groups nor providing the granted admin consent right away will cause Microsoft to enforce at least the first user to be requested to provide a reason for the consent request. After that Microsoft will automatically grant the consent for your tenant.

   Administrative accounts can also use the option *Consent on behalf of your organization* in above mentioned permission dialogue.

4. **Connect your Microsoft app in Zammad**

   Copy your new app registration's **Application (client) ID** and **Directory (tenant) ID** (found under **Overview > Essentials**) into Zammad in the admin panel, under **Channels > Microsoft 365 > Connect Microsoft 365 App**.

   Then, create a new client secret under **Certificates and Secrets > New Client Secret** and copy that into the Zammad admin panel, as well.

Congratulations! Now you're ready to connect Microsoft 365 or Outlook accounts to Zammad.

24.1.2 Account Setup

After you've *registered Zammad as an OAuth app* in your Azure Portal, you can begin connecting Microsoft accounts to Zammad.

But first, a word of warning! The import process does things you might not expect:

Danger: Zammad will delete all emails in your inbox during the import process.

Use the *Keep Messages on Server* setting to *disable this behavior*.

Warning: Zammad will send an auto-reply message to every email it imports. (Including the old ones!)

Make sure to *disable this behavior* prior adding an email account, and to turn it back on once all your messages have been imported.

Tip: Shared mailboxes are possible…

For this to function ensure to set a password for the account in question. You'll need these credentials to authenticate against later during adding the account.

Migrate an Existing Email Channel

If you've already added your Microsoft 365 account as a regular email channel, you'll have to convert it to a Microsoft 365 channel eventually: Microsoft is planning to end support for simple password authentication in third-party email clients (like Zammad).

Please refer the *Migrate from Email channel to Microsoft 365 channel* guide.

Requesting administrator consent

Hint: This step is only required if you didn't consent the permissions in App registration as administrator on behalf of your users. If you do not use this kind of security measurement, simply skip to *Add a New Account*.

Requesting the consent Within Zammad click on **Request Admin Consent** in order to request consent from your administrators. This is required in some tenants.

> **Note:** If your admin already provided consent, you'll be automatically authenticated, if this happens, continue with *Add a New Account*.

After authenticating against Microsoft and providing a reason for your request, you'll be redirect to the Zammad app. Zammad will then tell you to wait for your administrators consent. The consent grant or denial will be sent by mail.

Granting the consent (admin users) Within **Enterprise applications** go to **Admin consent requests**. You'll find all your user grant requests here.

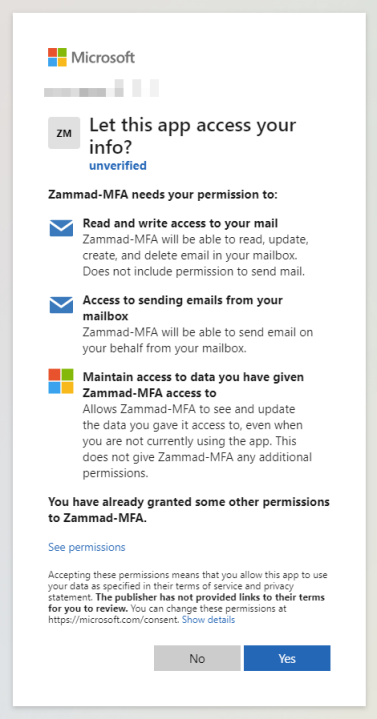
Clicking on a request allows you to review which user requested the consent for what app. Click on **Review permissions and consent** to start the grant process. You'll be asked for credentials - these credentials are the administrator credentials *not* the ones of the mailbox user to add.

After you've granted the request, continue with *Add a New Account*. (As mentioned in *Step 3.1 of the App registration*)

Add a New Account

Note: Below shown screencast shows the authentication process with enabled admin consent and already tenant wide granted permission.

If you're not using admin consent in your organization or you're authenticating a personal microsoft account, you'll see an additional permission dialogue you'll have to approve.

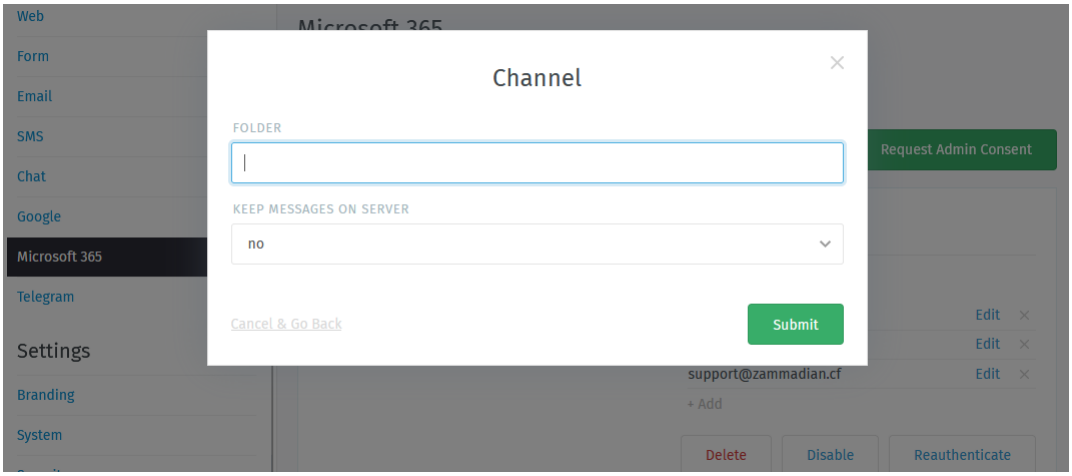


Click **Add Account** to connect your Microsoft 365 / Outlook accounts to Zammad. You will be redirected to a Microsoft sign-in and confirmation page.

Note: Aliases are not imported automatically.

See *Secondary Addresses* to add them yourself.

Channel



Folder Specify which folder (or *label*) to fetch from, or leave empty to fetch from `INBOX`.

If specifying a nested folder, be sure to use the full path; *e.g.,* `Inquiries/Tech-Support`.

Keep messages on server Specify what happens to your emails after Zammad imports them:

- `no` Zammad deletes all imported messages
- `yes` Zammad marks imported messages as read

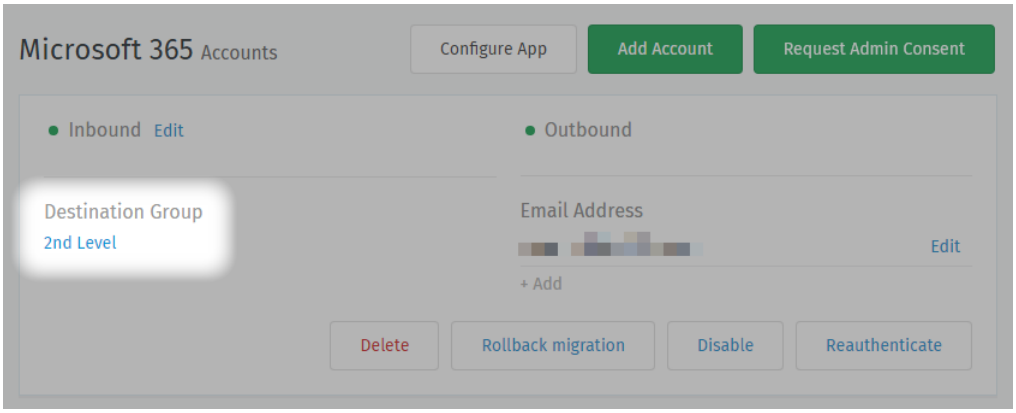
  (With this option, Zammad will only import unread messages. This means Zammad may miss messages if the mailbox is externally modified.)

Note: Why does Zammad delete messages by default?

If you never clean out your inbox, it'll eventually reach its storage limit, and your mail server will start **rejecting incoming messages**. Most Zammad users never even look at their inbox once it's set up, so they rely on Zammad to keep it clean for them.

If you choose **yes** here, remember that it's your responsibility to clean out your inbox from time to time to keep it below its storage limit.

After adding the account After successfully adding the Microsoft 365 mail account, you can adjust the default group Zammad is going to assign incoming new tickets to.



Only **active** groups will be displayed.

Changing this setting will not reassign existing tickets to the new group.

Hint: Don't forget to set the outgoing email address

In Zammad each group decides about the email address that's used for outgoing emails. The incoming group technically has no effect on it.

For this reason make sure to also adjust each affected *groups setting*.

Troubleshooting

I successfully added my account, but Zammad isn't fetching new email If you specified a *custom folder/label to fetch from*, are you sure incoming mail is arriving in that folder?

My mailbox was working fine but suddenly it fails with `EXPUNGE FAILED` This is a Microsoft 365 specific issue which you have to solve with Microsoft. However, you can have a look at Microsofts documentation which might allow you to solve the issue on your own.

As soon as the issue has been fixed, the Zammad channel will recover automatically.

24.1.3 Migrate from Email channel to Microsoft 365 channel

Zammad provides a migration logic that allows you to migrate existing Microsoft 365 accounts from the Email channel to the Microsoft 365 channel.

Note: Zammad is expecting specific settings

In order for Zammad to display the migration option, it expects the channels hostname to be `outlook.office365.com` for IMAP and `smtp.office365.com` for SMTP.

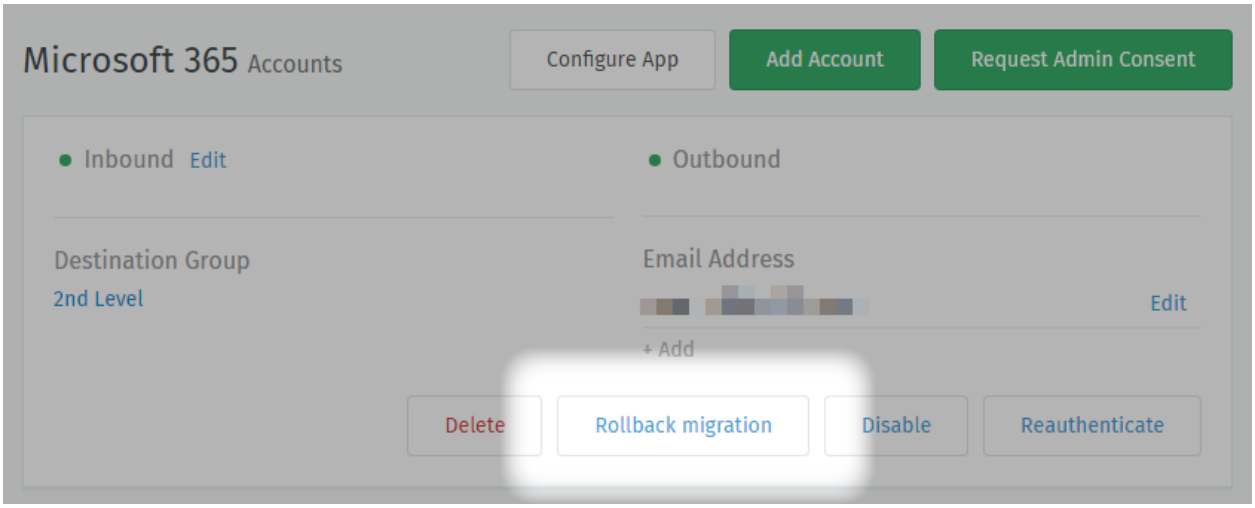
The easiest way to start the migration is to *Register an OAuth App* for your Microsoft accounts before migrating. However, if you don't, Zammad will ask you to provide your app credentials before allowing you to continue.

If you're ready to go, simply click on the *Migrate now!* button in the red banner of the email channel in question. Zammad will redirect you to Microsoft and request you to authenticate and consent to said account.

After you pressed next you'll be redirect to Zammad's Microsoft 365 channel overview. Your channel, if successful, is now migrated to an Microsoft 365 channel.

Rolling back the migration

In case something went wrong, Zammad allows you to roll back the migration for up to 7 days. For this time period Zammad will remember your original credentials and restore it if needed. These information will be removed entirely after 7 days.



Note: Secondary addresses in Microsoft 365 channels work (almost) **just like they do in email channels**, so this article is lifted (almost) verbatim from *here*.

24.1.4 Secondary Addresses

Secondary addresses (also known as *aliases*) allow you to send emails with a different "From:" address from the one on the account.

Fig. 1: Once you add a secondary address, you can configure a group to start sending emails with it.

Warning: Secondary addresses must be added to your Microsoft account first.

Personal accounts Use the Manage how you sign in to Microsoft panel at https://account.live.com.

Hosted Exchange accounts In your Exchange admin center:

1. select a user under **recipients > mailboxes**,
2. edit it (double-click or), and
3. add an an alias under **email address**.

Contact your administrator if you don't have access to an admin account.

Your email provider may also be set up to *receive incoming messages* for many addresses in the same mailbox. If this is the case, be sure to add your alternate inbox addresses here.

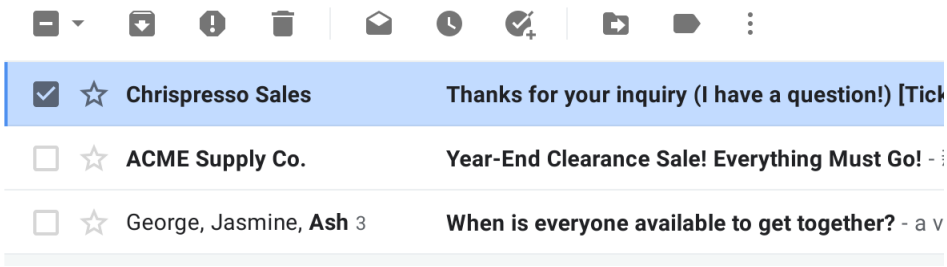


Fig. 2: A customer's inbox with an auto-reply from **Chrispresso Sales**.

Display Name The display name used for outgoing email.

Email display names value can be *further customized in the Settings tab*.

Email The alias address to send outgoing messages as.

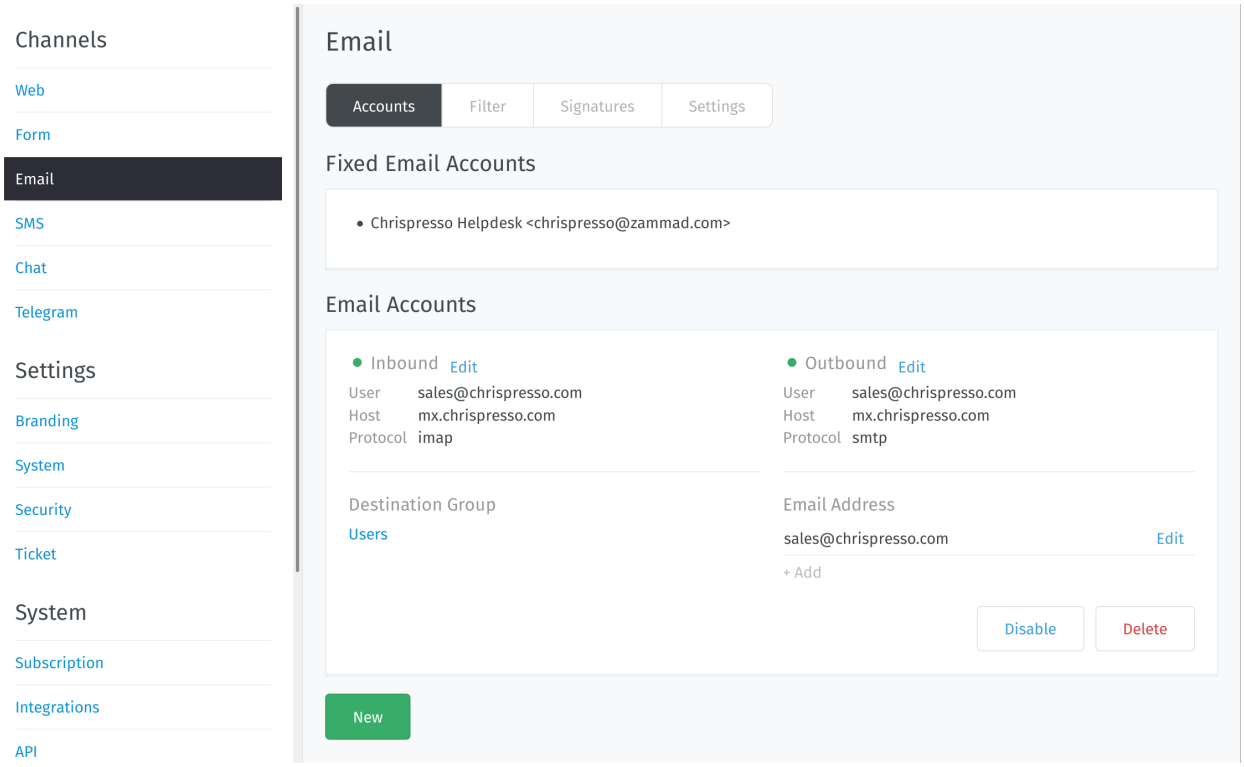
Channel The email account to be used when sending outgoing messages from this alias.

Note Optional. Only visible from this dialog, the REST API, and the Rails console.

Note: Managing accounts in Microsoft 365 channels is (almost) **just like it is in email channels**, so this article is lifted (almost) verbatim from *here*.

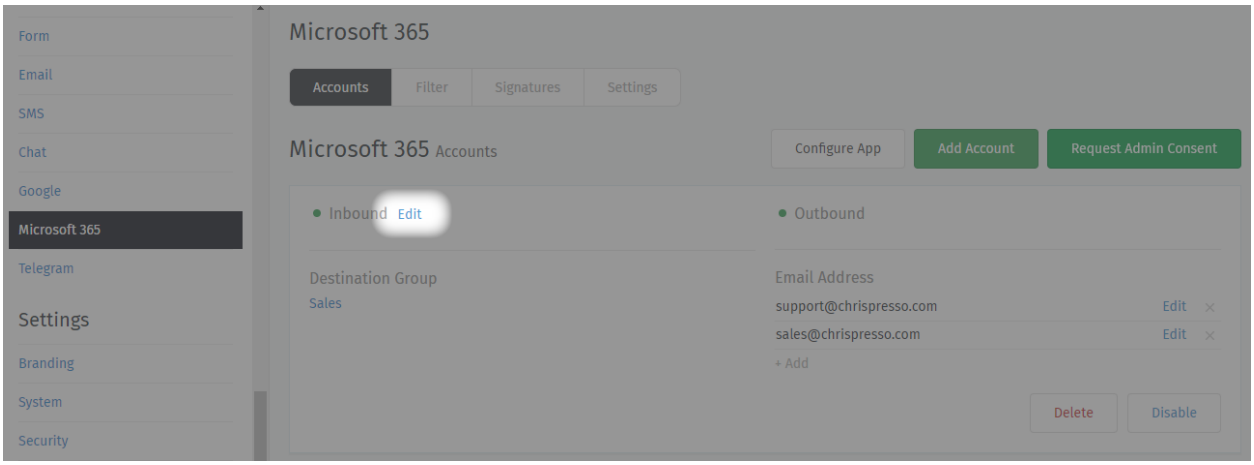
24.1.5 Managing Accounts

Once an account has been added, use the Accounts panel to edit its configuration.

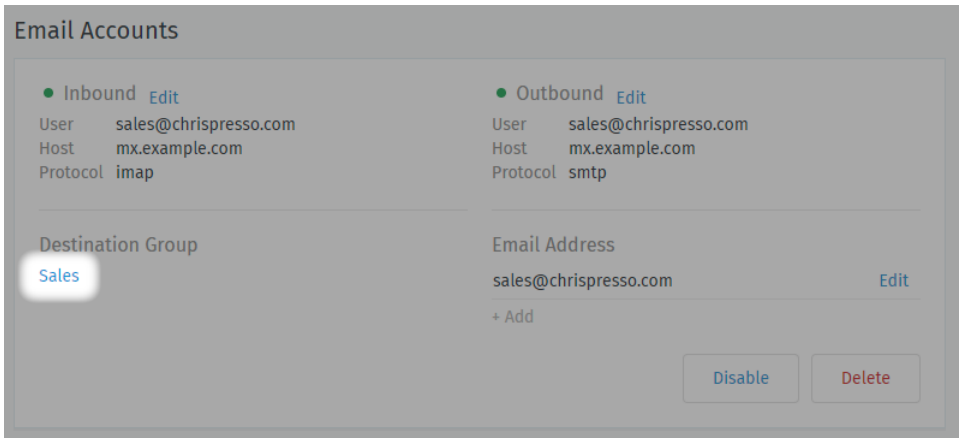


Fetch Preferences Click **Edit** on inbound account details to change how messages are retrieved from your account.

See *New Account Settings* for a detailed description of each option.



Destination Group Click on the group name to reassign the account.



Only **active** groups will be displayed.

Changing this setting will not reassign existing tickets to the new group.

Hint: **Still can't send outgoing email tickets?** Check your *group settings*.

Email Address Use the **+ Add** or **Edit** buttons to set up secondary addresses on this account.

See *Secondary Addresses* for a detailed description of each option.

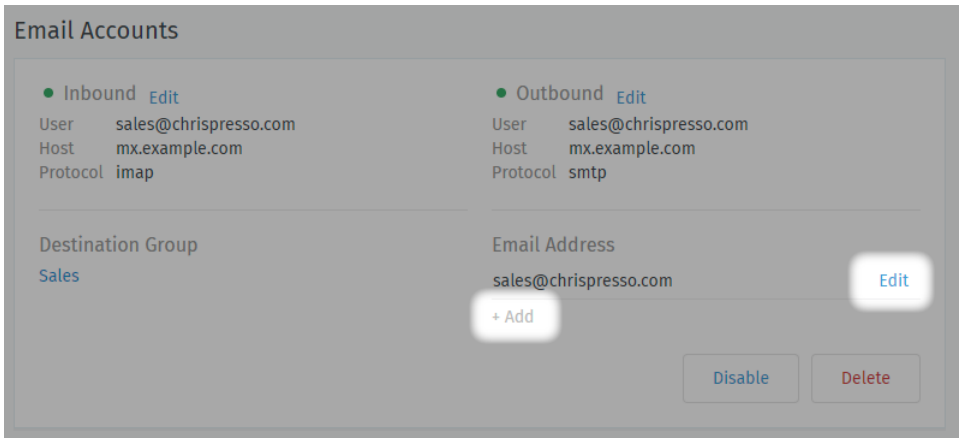
Enabled / Disabled Disabling an account temporarily prevents Zammad from importing its messages.

This may be necessary during scheduled maintenance or when migrating your installation to a new host.

Note: Disabling an account disables *outgoing* messages for it, as well.

Delete Deleting an account removes its configuration from Zammad entirely.

Note: **Additional Steps Required**



Groups need an assigned an address to send outgoing emails. If you delete a group's assigned address, agents belonging to that group won't be able to send emails until you assign it a new one.

Register an OAuth App Use the **Connect Microsoft 365 App** dialog to register Zammad as an OAuth app on Microsoft.

(This step is **required**; read on to learn why.)

Account Setup Use the **Add Account** dialog to connect your account.

You're migrating existing email channels? Look below!

Migrate from Email channel to Microsoft 365 channel Use the *Migrate now!* button within your email channels to quickly move your mailboxes to Microsoft 365. You can roll back if things hit the fan!

Secondary Addresses Send and receive email at **additional email addresses**, all through the same mailbox/account.

Managing Accounts Edit the configuration of existing accounts in the **Accounts Panel**.

Note:

How do I use my Microsoft 365 account for outgoing system notifications?

On **subscription/cloud-hosted instances**, you can't. Notifications will always come from "Notification Master <noreply@your.zammad.domain>".

On **self-hosted instances**, we still don't recommend it. Using a Microsoft account for automated, outgoing messages is risky: users who exceed Microsoft's email sending limits can have their accounts suspended.

Set up a generic *email channel* instead, then use the *Email Notification* setting.

Note: Filters in Microsoft 365 channels are **just like filters in email channels**, so this article is lifted verbatim from *here*.

24.2 Filters

Postmaster filters allow you to match email headers (e.g. `From`, `To`, `Subject`, `X-Spam-Flag` etc.) and execute a set of actions whenever Zammad's email parser encounters a matching email. The actions will be applied to the ticket that is created or updated by this email. Here are some examples of what is possible with filters:

Automatically dispatch tickets into certain groups: For example, tickets from `amazon.com` could automatically be dispatched to the Purchasing group.

From: contains: `regex:(\.|@)amazon\.com`
Group: Purchasing

Note: Note that the Group action only has an effect when the matching email results in a new ticket. Zammad will not change the group of existing tickets.

Automatically increase the priority of tickets from a VIP customer:

From: contains: `ourvipcustomer@example.com`
Priority: 3 high

Note: Note that the Priority action only has an effect when the matching email results in a new ticket. Zammad will not change the priority of existing tickets.

Automatically tag and close spam tickets that have been marked as spam by anexternal spam filter (e.g. SpamAssassin):

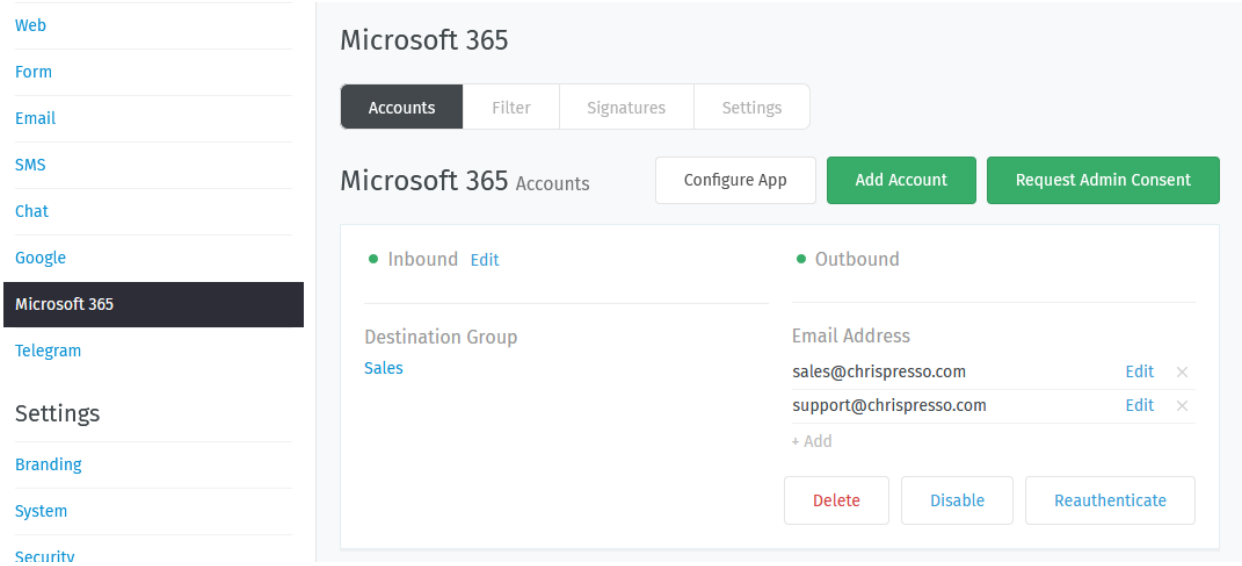
X-Spam-Flag: contains: `YES`
Tag: add: `spam`
State: closed

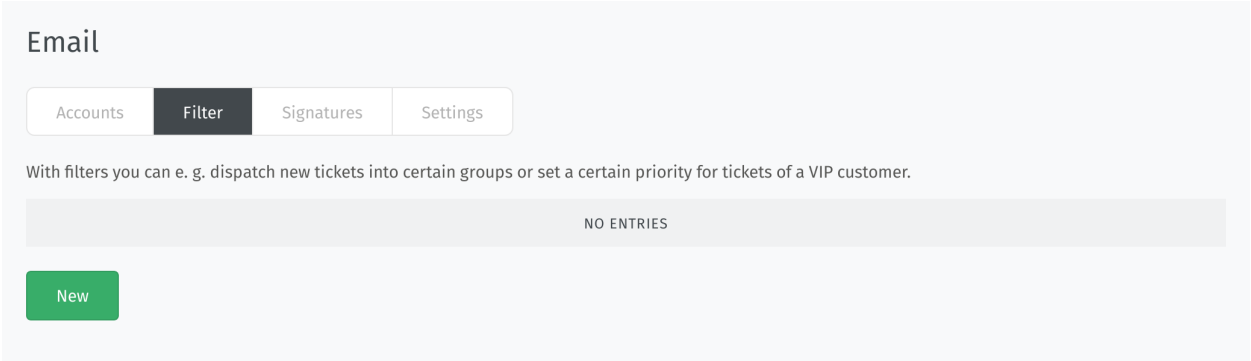
Note: Note that the State action only has an effect when the matching email results in a new ticket. Zammad will not change the state of existing tickets. It will add the tag though if it missing, even if the mail is an update to an existing ticket.

The following actions are only effective when creating tickets:

- Group
- State
- Priority
- Owner

Different attributes of a filter can be combined with each other. Likewise, the following actions can be combined. The supported matches are "contains" and "contains not"; for advanced matching, you can use regular expressions by prefixing the string with `regex:`.


Web
Form
Email
SMS
Chat
Google
Microsoft 365
Telegram
Settings
Branding
System
Security
Microsoft 365
Accounts
Filter
Signatures
Settings
Microsoft 365 Accounts
Configure App
Add Account
Request Admin Consent
Inbound Edit
Outbound
Destination Group
Sales
Email Address
sales@chrispresso.com Edit
support@chrispresso.com Edit
+ Add
Delete
Disable
Reauthenticate

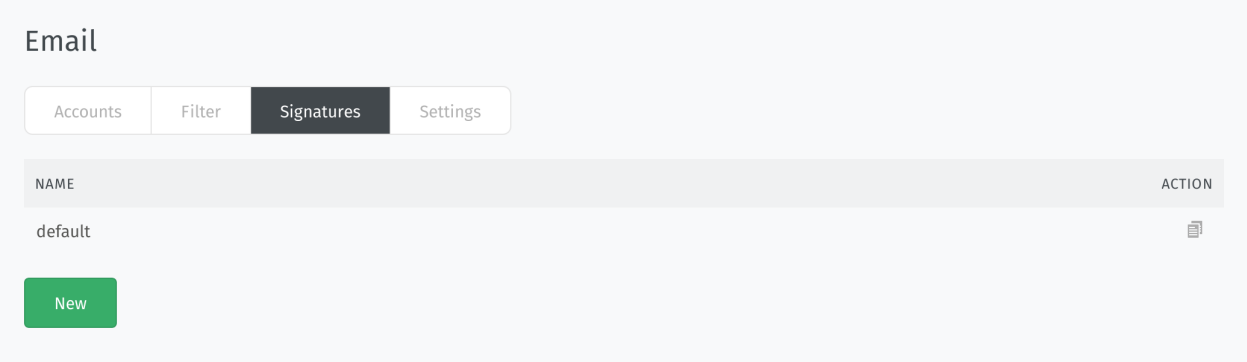

Email
Accounts
Filter
Signatures
Settings
With filters you can e. g. dispatch new tickets into certain groups or set a certain priority for tickets of a VIP customer.
NO ENTRIES
New

Note that Zammad matches against the full header, e.g. for a mail with `From: Display Name <display.name@example.com>`, the `From` condition will test against `Display Name <display.name@example.com>`. This is especially important when using anchored regular expressions; `regex:^display\.name@example.com$` would not match this mail!

It should be borne in mind that the combined attributes build on each other. If a filter is no longer needed, it can either be temporarily set inactive or deleted directly.

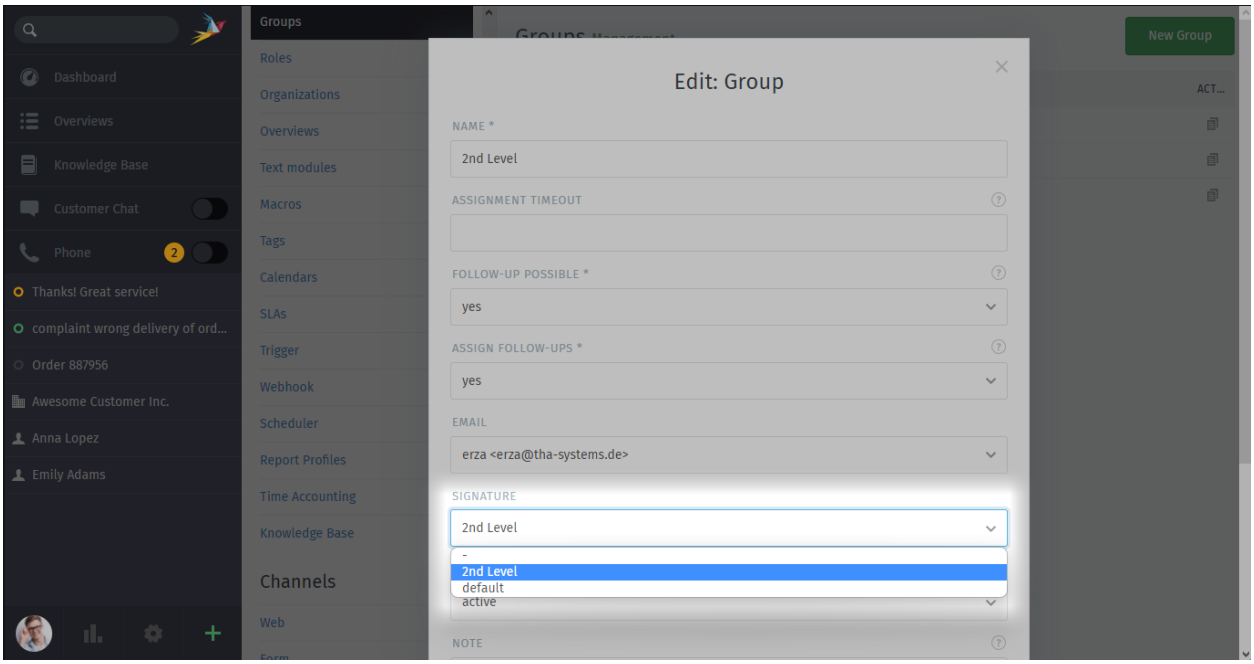
Note: Signatures in Microsoft 365 channels are **just like signatures in email channels**, so this article is lifted verbatim from *here*.

24.3 Signatures



You can create a separate signature for each group in Zammad. The individual signatures can be created and edited here.

Afterwards, the existing (and active) signatures are available in the group editing mask:



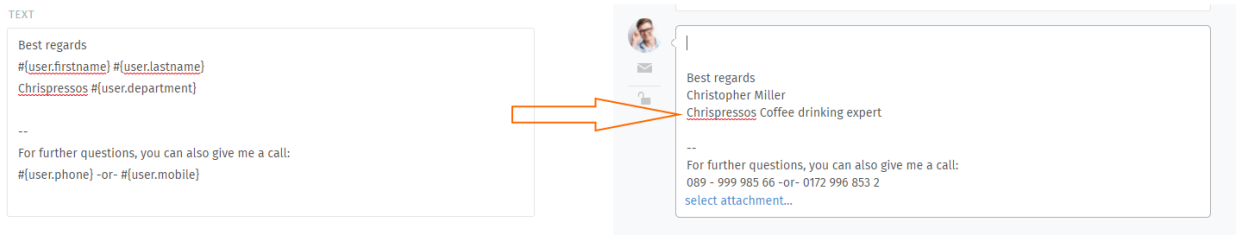
Each group can be assigned its own signature, but they can also all use the same signature.

24.3.1 Dynamic Signatures

To individualize the signatures, it is possible to automatically load specific information into a signature via *Variables*. All information stored on the ticket, assigned customers or agents can be inserted. This makes it possible to design the signature individually. To load a list of available variables, enter two colons (`::`) into the *Text* box of the signature editor.

Hint: Please keep in mind that specific information might not be available during ticket creation. The best example here is the ticket number / id. Specific information are created with submitting the ticket and thus are not available before submitting.

Here is an example of a signature with variables and the result when you write a mail:



Note: Settings in Microsoft 365 channels are **just like settings in email channels**, so this article is lifted verbatim from *here*.

24.4 Settings

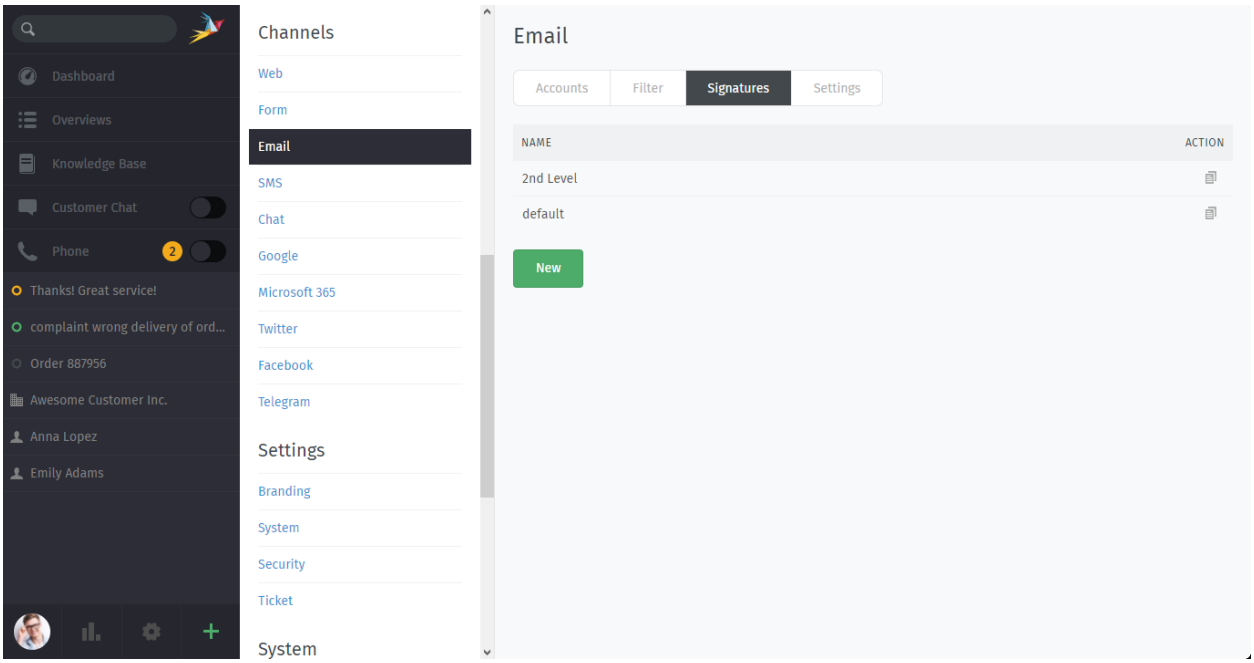
Below you can find the currently available email-related settings. Most of these settings have default values which can be found in this list as well.

Note: Some email-related settings are ticket-based settings, which is why they can be found in the *Composer Settings*.

24.4.1 List of Settings

Notification Sender: Default value `Notification Master <noreply@#{config.fqdn}>` This is the default sender address for Zammad that affects all mails but those generated because of replies (like triggers or agent-based mails). Your customers normally will not see this address. This email address does not need to receive and can't be assigned to a group.

Note: This address is relevant for agent notifications and password reset mails (also affects customers).



Additional follow-up detection In some situations the normal follow-up detection is not enough. This might be due to missing references in the subject (the ticket hook and number). These options can help to recognize follow-ups to existing tickets.

Note: Please note that searching in attachment and body might lead to false follow-up detection.

Maximum Email Size: Default value `10 MB` This one is pretty obvious: It defines the maximum allowed size of an email Zammad will fetch. Zammad will not fetch Mails that are bigger than this option.

Hint: This technically also affects attachments for articles.

Send postmaster mail if mail too large: Default value `yes (enabled)`

Note: Upgraded installations will, by default, have the value set to `no (disabled)`.

Option set to `yes` This setting will cause Zammad to automatically reply to mails that exceed the above mail size limit with a postmaster style mail. This will help your user to understand that his mail did not arrive and won't be reviewed by you.

Note: Zammad will still download and remove (if enabled) the mail from the mailbox. Instead of importing it to the database, it will save the affected mail to `/opt/zammad/tmp/oversized_mail/`.

Option set to `no` If the option is set to no, Zammad will not reply to mails that are too big. Your customer will **not notice** that the mail was too large! Instead, Zammad will use the monitoring endpoint to alert its administrators that it can't fetch a too large mail.

Learn more about *Monitoring*.

Sender based on Reply-To header: Default value `not set (-)` This setting decides how Zammad should recognize its customers from emails that contain a `Reply-To` header. This comes in useful if you're working with contact forms that need to use reply to headers.

Option set to `Take reply-to header as sender/from of email.` This setting will overwrite the initial `FROM` to the value used in `Reply-To` completely.

Option set to `-` or `Take Reply-To header as sender/from of email and use the real name of origin from.` This setting will partially overwrite the initial `FROM`. It uses the mail address from the `Reply-To` header and uses the given name of the `FROM` header, if given.

Customer selection based on sender and receiver list: Default value `yes` This option decides how Zammad should react if an agent sends a email to it.

Option set to `yes` The first user / email address from the recipient list will be used as the ticket customer.

Option set to `no` The agent will be set as ticket customer.

Note: Currently agents can't be customers within the UI. While Email communication works, agents can't see their own tickets (as a customer) if they don't have access to the group.

Block Notifications With the regex that can be defined here, you can ensure not to send any notifications to specific systems. By default this especially affects typical system addresses which can't receive emails anyway.

The default value is: `(mailer-daemon|postmaster|abuse|root|noreply|noreply.+?|no-reply|no-reply.+?)@.+?`

Sender Format: Default value `Agent Name + FromSeparator + System Address Display Name` This configures the display name used in the `FROM` header of mails Zammad sends.

Note: This does not affect Notification mails (to agents) and password reset mails. Emails that are not sent by agents (e.g. trigger-based notifications) will always fallback to `System Address Display Name` if needed.

Option set to `Agent Name + FromSeparator + System Address Display Name` This will cause Zammad to set the `FROM` header to agent name and the channel's display name, divided by a separator (configured below).

Example: `Christopher Miller via Chrispresso Inc.`.

Option set to `System Address Display Name` This will cause Zammad to always use the display name of the used channel in the `FROM` header.

Example: `Chrispresso Inc.`

Option set to `Agent Name` Zammad will use the agent's name which is very personal.

Tip:

Usually you'd also want to remove the ticket slug from the subject in those cases.

Learn more in *Settings → Ticket*.

Sender Format Separator: Default value `via` This is a can be a string you can freely choose. It divides the agent's name and the display name of the channel whenever needed.

Ticket Subject Forward: Default value `FWD` The above string will be used on the subject if you forward an email from Zammad.

Note: : will be automatically appended to the above string.

Ticket Subject Reply: Default value `RE` The above string will be used on the subject if you reply to a mail from Zammad.

Note: : will be automatically appended to the above string.

Ticket Subject Size: Default value `110` This setting enforces a maximum length for subjects when replying. If the subject you're using for your reply is too long, Zammad will automatically truncate the length and insert `[...]` to show it has shortened the subject.

Example: `RE: Test somew[...] [Ticket#123456]`

Note: This does **not** limit ticket titles within the UI, just the subjects when replying to an email.

24.4.2 Enhanced settings

Some less relevant settings can be changed via rails console if needed. As an example, Zammad allows you to send all outgoing communication to a BCC address for archiving reasons if needed. You can find the needed commands within the advanced customization settings.

Note: Email header manipulation in Microsoft 365 channels work **just like in email channels**, so this article is lifted verbatim from *here*.

24.5 Email header manipulation

Email header manipulation allows you to re-route or adjust tickets apart from filters or triggers. Like an API call, but with emails.

Header checks are case insensitive.

Warning: Trusted channels required

Below options are a potential risk with external communication and thus require channels being set to trusted explicitly.

Tip: Below headers are examples and –in our opinion– the most relevant ones. However: You can adjust mostly any article or ticket attribute (yes, custom ones as well) if you know the attribute's exact name.

The name column within *object's management* provides easy access to objects attribute names.

24.5.1 Trigger auto responses

Normally Zammad runs internal checks to see if an email is an automatic response. In these cases Zammad will not send trigger based responses.

There may be use cases where this behavior may be in your way, below options allow you to overcome this issue.

Note: In some cases combining below headers is crucial. This is intentional but may be confusing.

`x-zammad-send-auto-response` Set to `false` to disable trigger based responses. If set to `true` Zammad will send a response.

Hint: This option *does not* work if e.g. `precedence: list` is set unless you use below auto response header as well.

`x-zammad-is-auto-response` Providing this header allows you to tell Zammad that the mail in question is an auto generated response (`true`). This will cause email based triggers to be skipped.

Set this header to `false` if you want to generate auto responses.

Tip: This header allows you to overwrite auto detects for e.g. `precedence: list`.

24.5.2 Ticket attributes

Zammad allows you to use headers to manipulate ticket creations or follow ups. The manipulation can be used instead of triggers. Triggers are considered *after* header settings and thus can still overrule.

Note: Zammad differentiates between ticket creation and follow up

For creations use: `X-Zammad-Ticket-{Attribute Name}`
For follow ups use: `X-Zammad-Ticket-FollowUp-{Attribute Name}`

This allows you to ensure the changes are only applied in the required situation.

Warning: About values

- While headers are not case sensitive, values like e.g. priority names are case censitive: `1 low` will work, but `1 lOw` will not!
- When using attributes that require date / time values, ensure to use Time Zoned Times. e.g. for 28th September 2021 on 8 am CEST, either use:
  - `2021-09-28T08:00:00+0200`
  - `2021-09-28T08:00:00+02:00`
  - `2021-09-28T06:00:00.000Z`

`X-Zammad-Ticket-Priority` & `X-Zammad-Ticket-FollowUp-Priority`

Allows you to adjust a ticket's priority.
Example: `X-Zammad-Ticket-Priority: 1 low`

X-Zammad-Ticket-Group & X-Zammad-Ticket-FollowUp-Group

Allows you interfere with regular channel routing of the ticket.
Example: `X-Zammad-Ticket-Group: Sales`

X-Zammad-Ticket-Owner & X-Zammad-Ticket-FollowUp-Owner

Directly assign or change the ticket owner. Valid values are either `login` or `Email`
Example: `X-Zammad-Ticket-Owner: jdoe`

X-Zammad-Ticket-State & X-Zammad-Ticket-FollowUp-State

Set a specific ticket state.
Example: `X-Zammad-Ticket-State: closed`

Note:

Pending states always require the `pending_time` attribute on top.
Like so: `X-Zammad-Ticket-Pending_Time: 2021-09-26T08:00:00+0200`

X-Zammad-Customer-Email

Manipulate the ticket customer - this can be a different user than the actual sender. Replying to the original sender is still possible.
Example: `X-Zammad-Customer-Email: jdoe@example.com`

Note: This header is not available for follow ups.

X-Zammad-Customer-Login

Manipulate the ticket customer - this can be a different user than the actual sender. Replying to the original sender is still possible.
Example: `X-Zammad-Customer-Login: jdoe`

Note: This header is not available for follow ups.

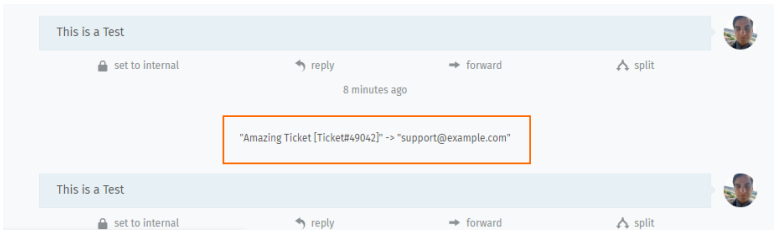
24.5.3 Article attributes

If needed Zammad allows you to manipulate attributes or states of fetched email articles.

X-Zammad-Article-Sender

Manipulate the sender type (Agent, Customer, or System)
Example: `X-Zammad-Article-Sender: System`

Warning: System Emails are indicated in a similar way as trigger-response like entries Users can't see them natively.



`X-Zammad-Article-Type`

Change the article type of your incoming mail. This requires you to know which article types are available in your system.

Example: `X-Zammad-Article-Type: phone`

> **Warning:** This header can cause *serious issues* in your instance and may lead to unexpected behavior. Only use with absolute care!

`X-Zammad-Article-Internal`

Manipulate the default article visibility.

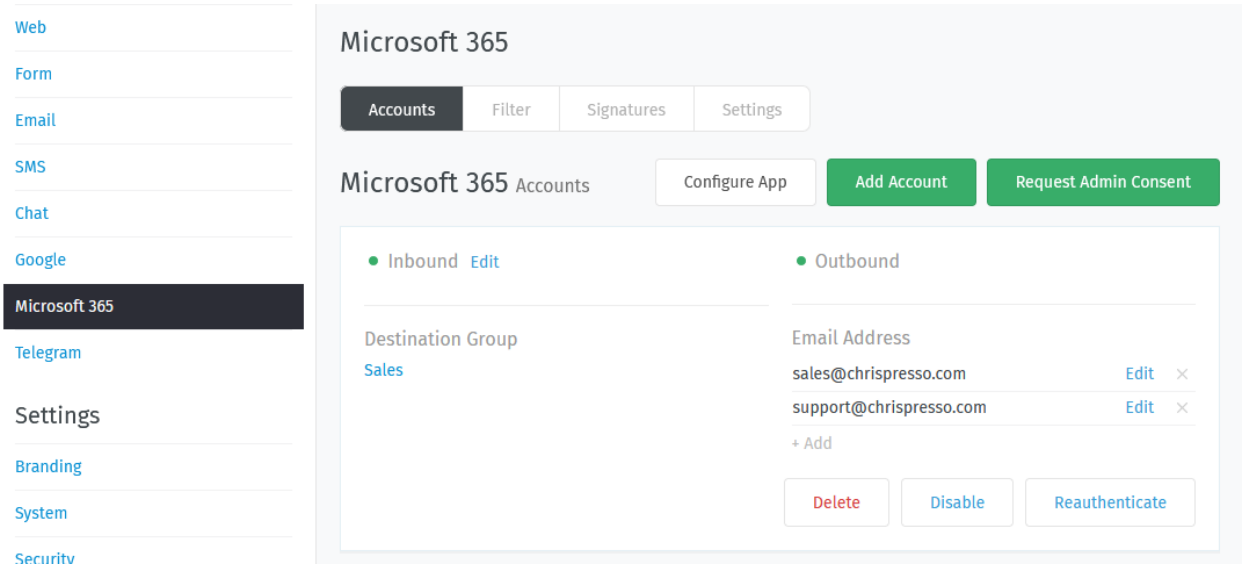
Example: `X-Zammad-Article-Internal: true`

`X-Zammad-Ignore`

Tell Zammad to silently drop the Email.

Example: `X-Zammad-Ignore: true`

Connect a Microsoft 365 account (formerly "Office 365") to Zammad.



Note: Microsoft 365 channels are a specialized kind of *email channel*.

This documentation part does not cover *user authentication*.

If you're already familiar with email channels, you can skip most of this—but **watch out for the "Accounts" section**, which has a few extra quirks due to Microsoft's strict security measures.

Accounts **Connect Zammad to your email provider** so that it can watch your inbox, send auto-replies, and more.

(Self-hosted users may have already completed this step during new system setup.)

Filters **Make sure new tickets show up in the right place** with automated, if-this-then-that rules for all incoming email.

Signatures Customize signatures for all outgoing email.

Settings Manage options like:

- set the "From:" address on system notifications
- raise the limit on attachment sizes
- modify subject-line prefixes (*e.g.,* use "AW:" instead of "RE:")

Hint: Want to **manually edit email subjects** or **always copy parent messages into your replies**?

Check the *Composer Settings*.

Header manipulation Manipulate auto response behavior or incoming routing.

> **Warning:** This is a very advanced topic.

CHAPTER

TWENTYFIVE

TWITTER

Zammad supports Twitter integration, meaning you can send and receive tweets and DMs just like emails!

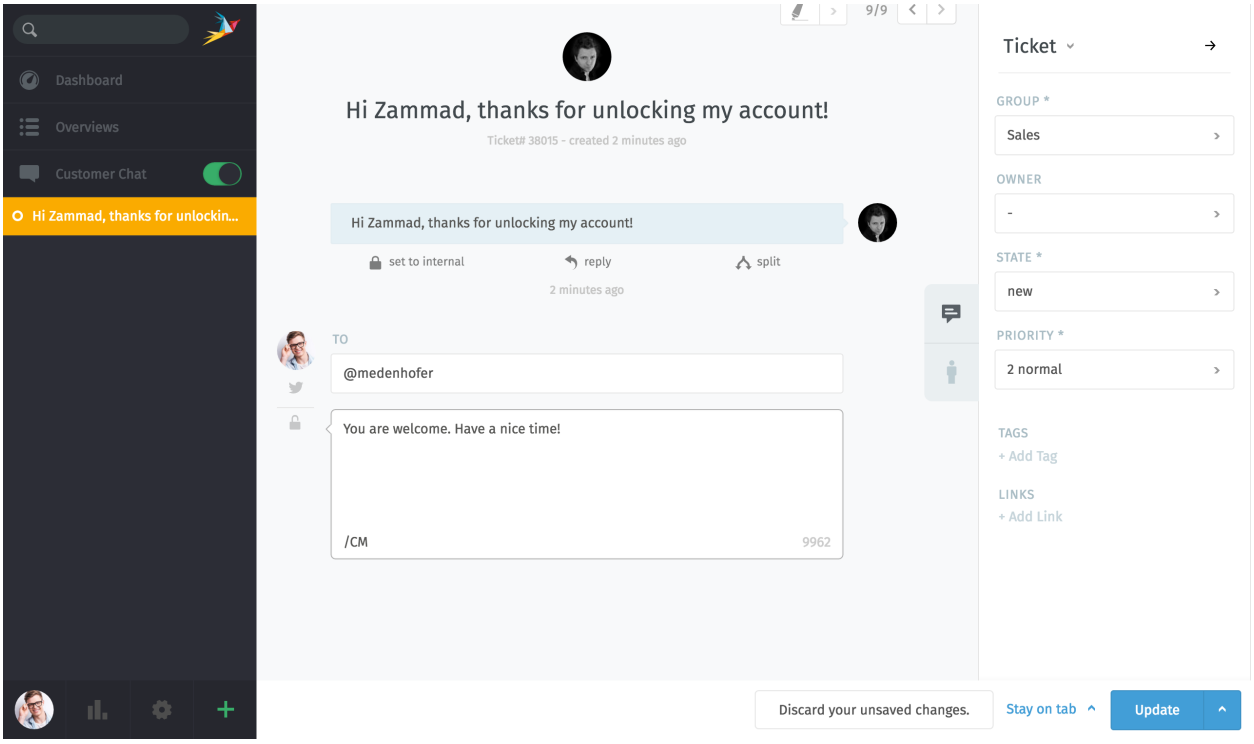


Fig. 1: Twitter tickets will show a Twitter bird in the reply area. Just click on the reply button (as you would for an email) to tweet back.

Note: To set it up, follow the steps below:

1. Apply for a Twitter Developer account.
2. Create a new Twitter app for Zammad.
3. Set your new app's permissions to *Read, write, and access direct messages.*
4. Generate a new access token & secret.
5. Set up a dev environment for the Account Activity API.
6. Add your new Twitter app in Zammad.
7. Add your Twitter account in Zammad.

8. Configure filters for creating new tickets based on **#tags** and **@mentions**.

Read on for details about each step.

25.1 1. Apply for a Twitter Developer account

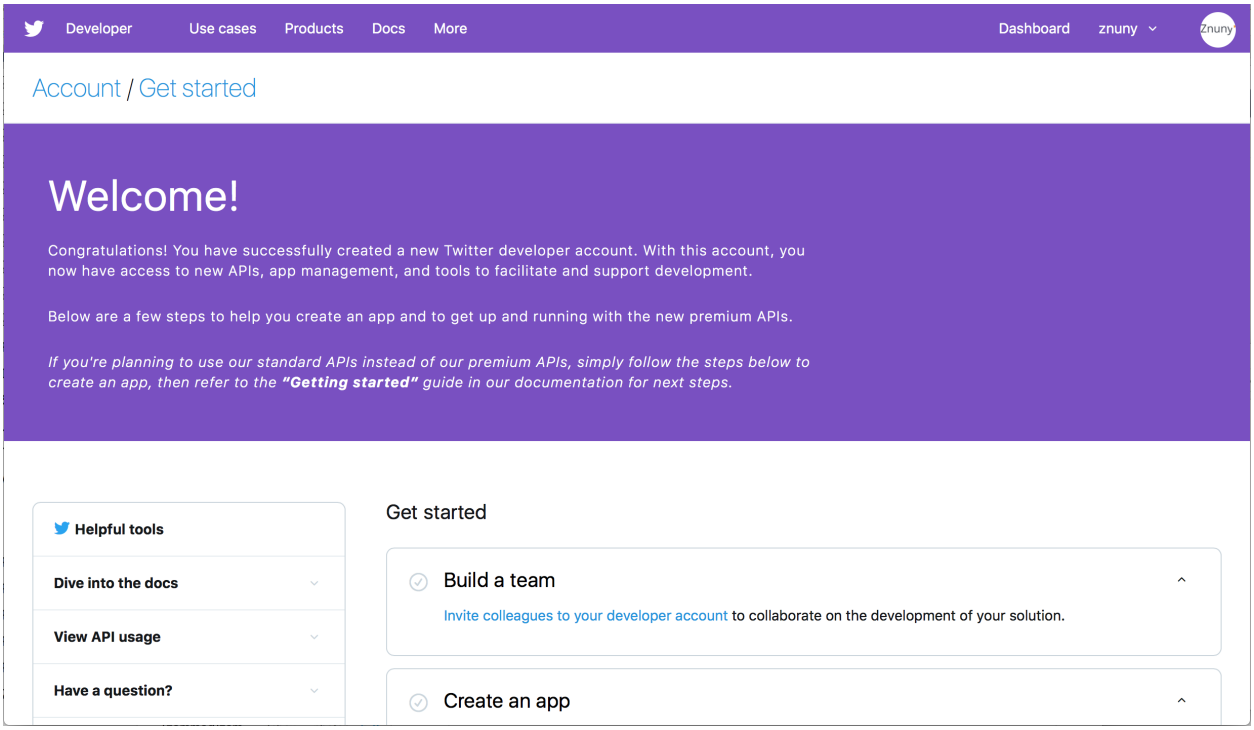


Fig. 2: This welcome page is displayed after completing the application for a Developer account.

You will need a regular Twitter account with a verified phone number to apply for a Twitter Developer account (at https://developer.twitter.com).

During the application process, you will be asked to **Describe in your own words what you are building**. You may use the answer below:

```
1. To manage customer service communications for our organization.
2. No.
3. Our use case involves posting original tweets in response to tweets and
   DMs we receive. We will not use the Twitter API to post or like “content”.
4. Our application will display individual tweets in their original form to
   authorized customer service agents of our organization only.
```

25.2 2. Create a new Twitter app for Zammad

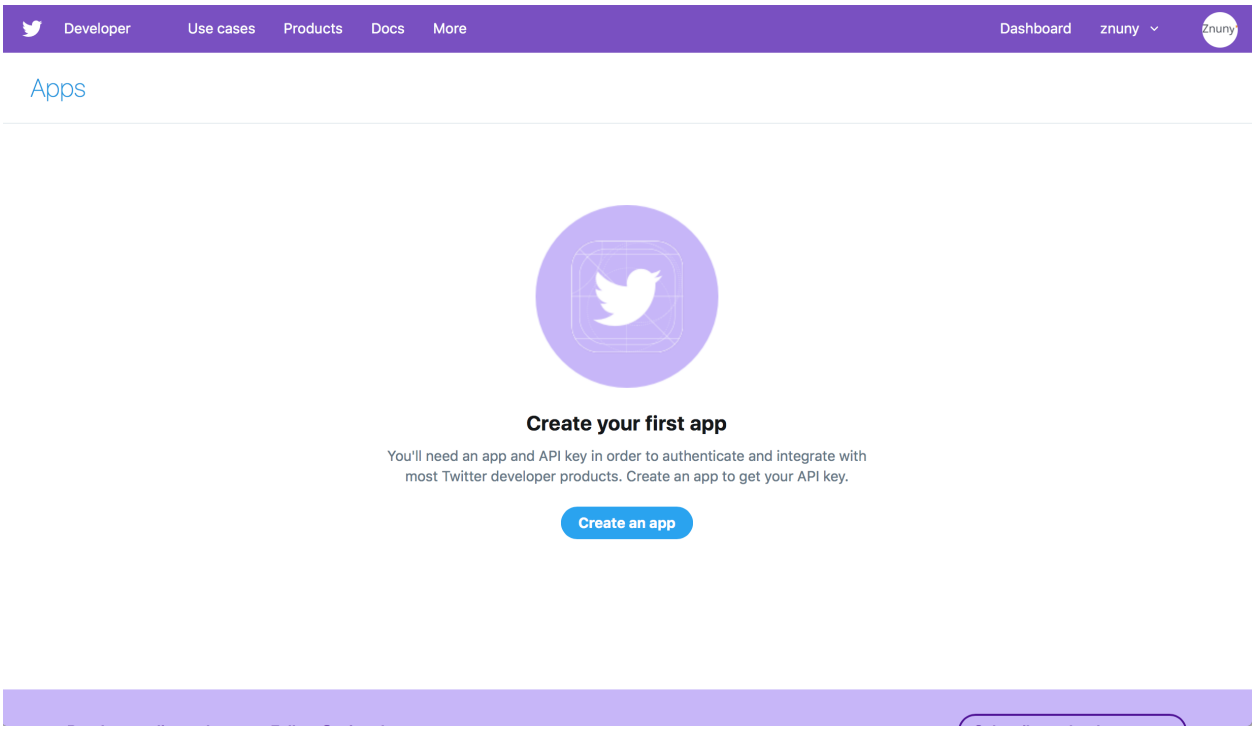


Fig. 3: To create a new app, select **Apps** under your user menu, then click **Create an app**.

Once you have finished setting up your Developer account, use it to create a new Twitter app. The following fields are **required:**

App name Must be unique *across all of Twitter*. No other developer account may create an app with the same name.

Application description Anything is fine here.

Website URL The URL of your Zammad instance.

Callback URLs `https://<zammad_instance>/api/v1/external_credentials/twitter/callback`

This URL is also visible in the **Connect Twitter App** dialog of your Zammad admin settings panel.

Tell us how this app will be used Anything is fine here. We suggest the following:

```
This app will be used to manage Twitter communications between our
customers and our organization's customer service agents on Zammad.
```

25.3 3. Set your new app's permissions

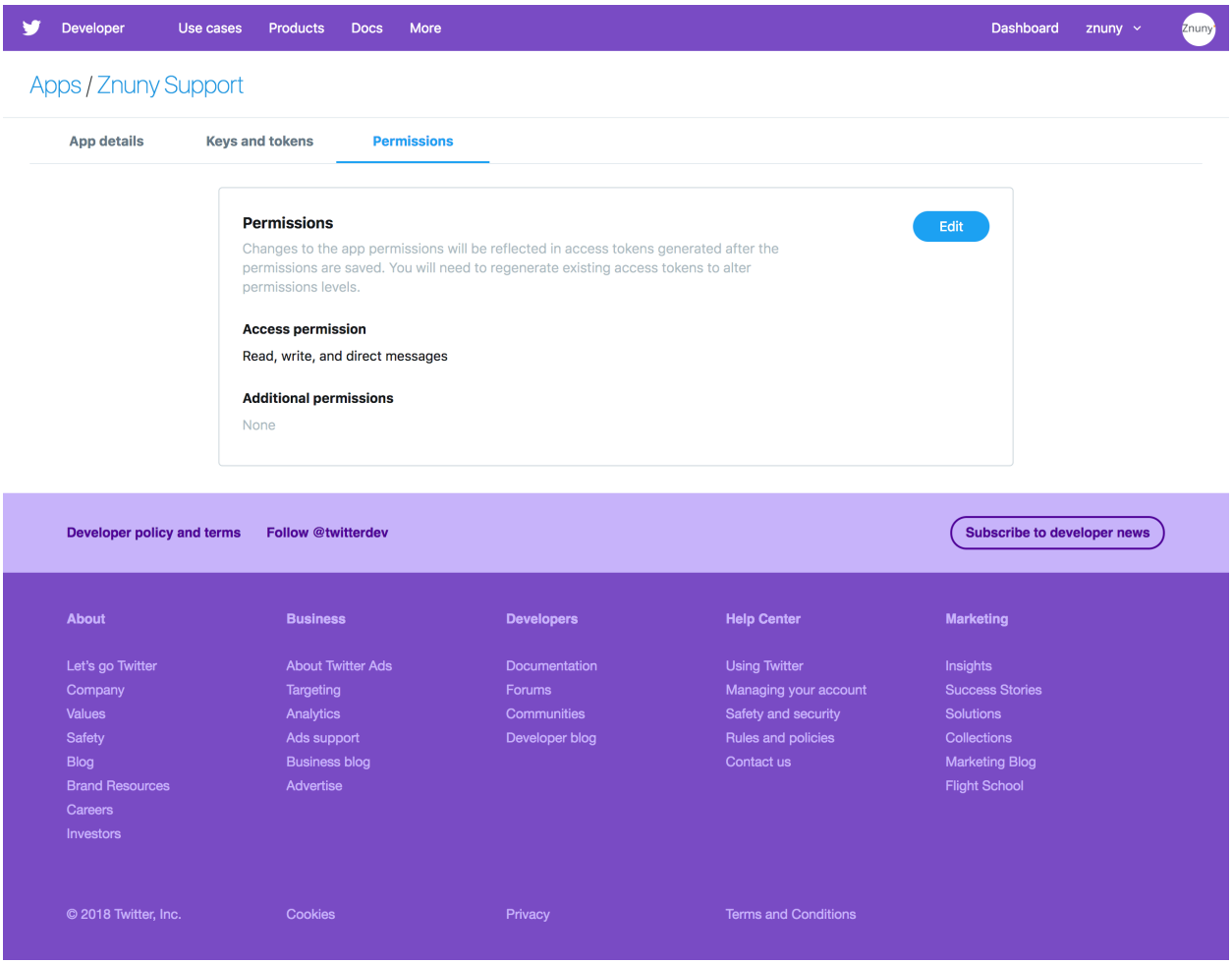


Fig. 4: Select *Read, write, and access direct messages.*

25.4 4. Generate a new access token & secret

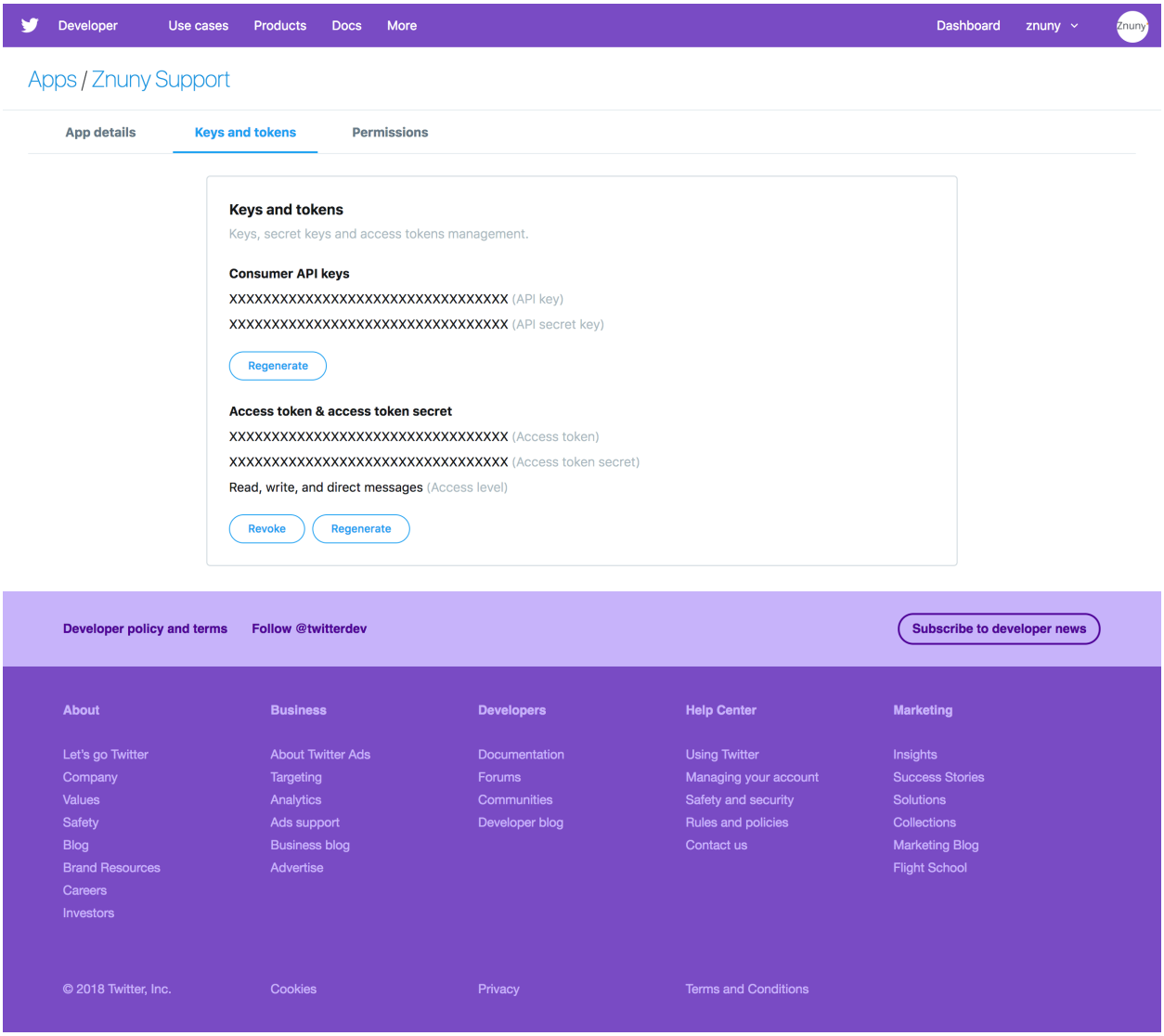


Fig. 5: You will need all four keys/tokens later, so don't close this tab.

25.5 5. Set up a dev environment

After creating your app, set up a dev environment for the Account Activity API.

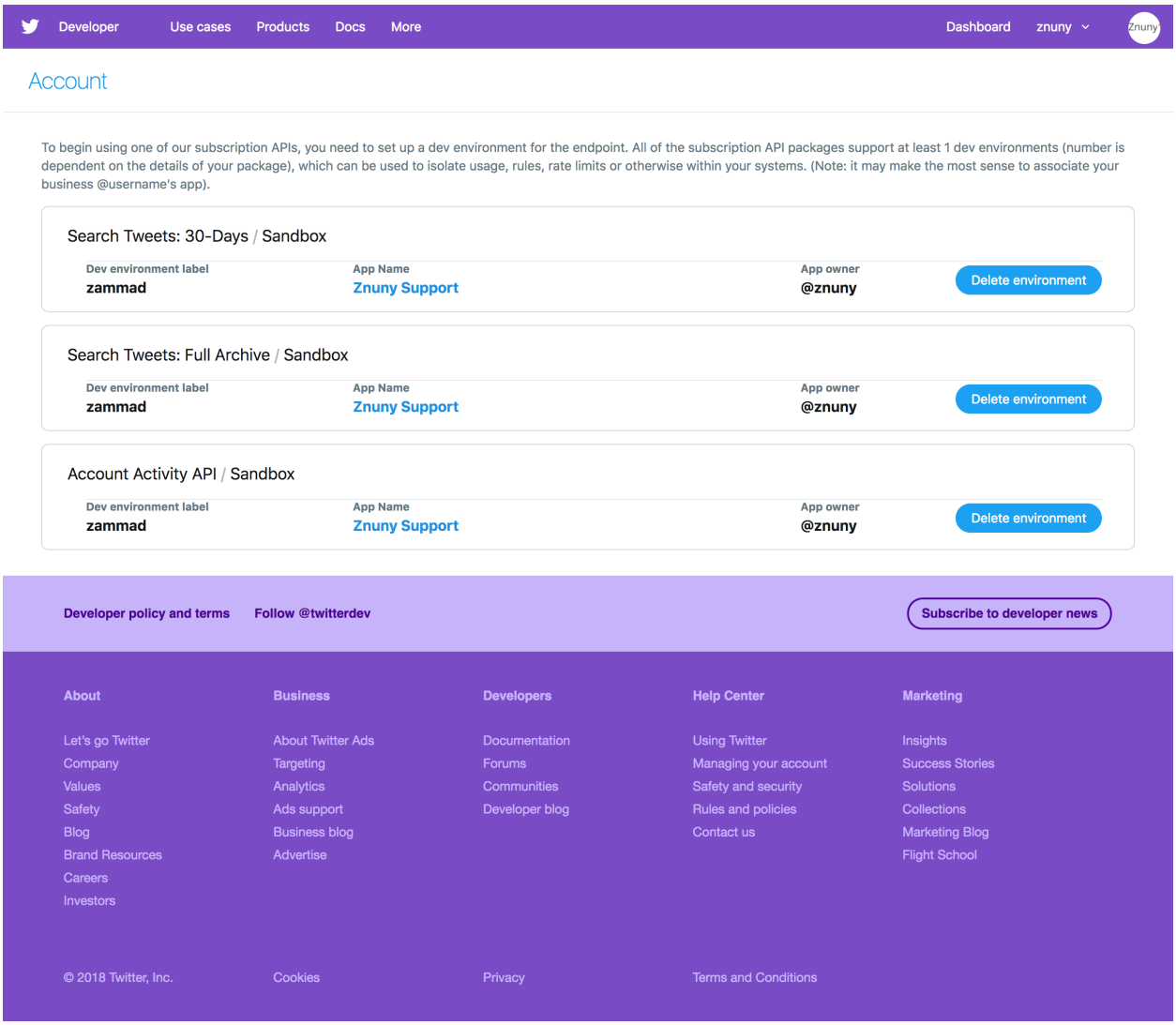


Fig. 6: Name it whatever you like (*e.g.*, `zammad`). You will need the label later, so don't forget it.

25.6 6. Add your new Twitter app in Zammad

25.7 7. Add your Twitter account in Zammad

25.8 8. Configure filters

That's it! Now, incoming tweets and DMs will be automatically turned into Zammad tickets.

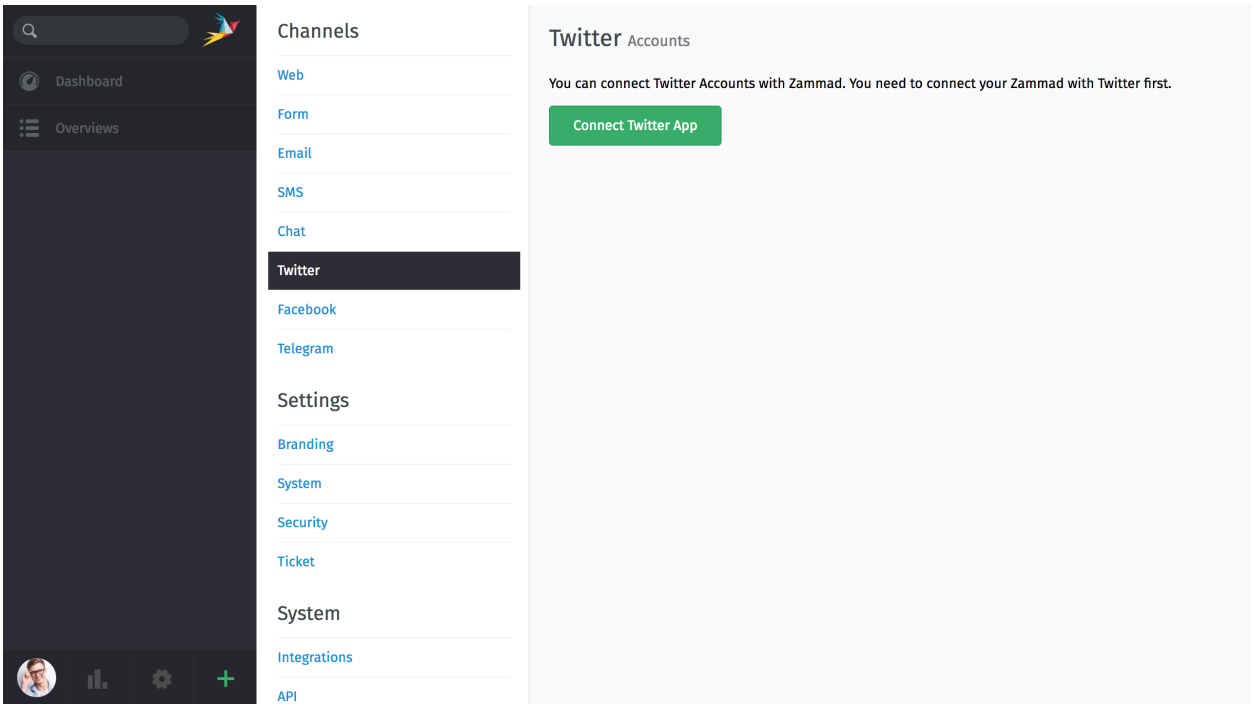


Fig. 7: Add your new Twitter app under **Channels > Twitter** in the admin settings panel. You will need the keys, tokens, and dev environment label from Steps 4 and 5.

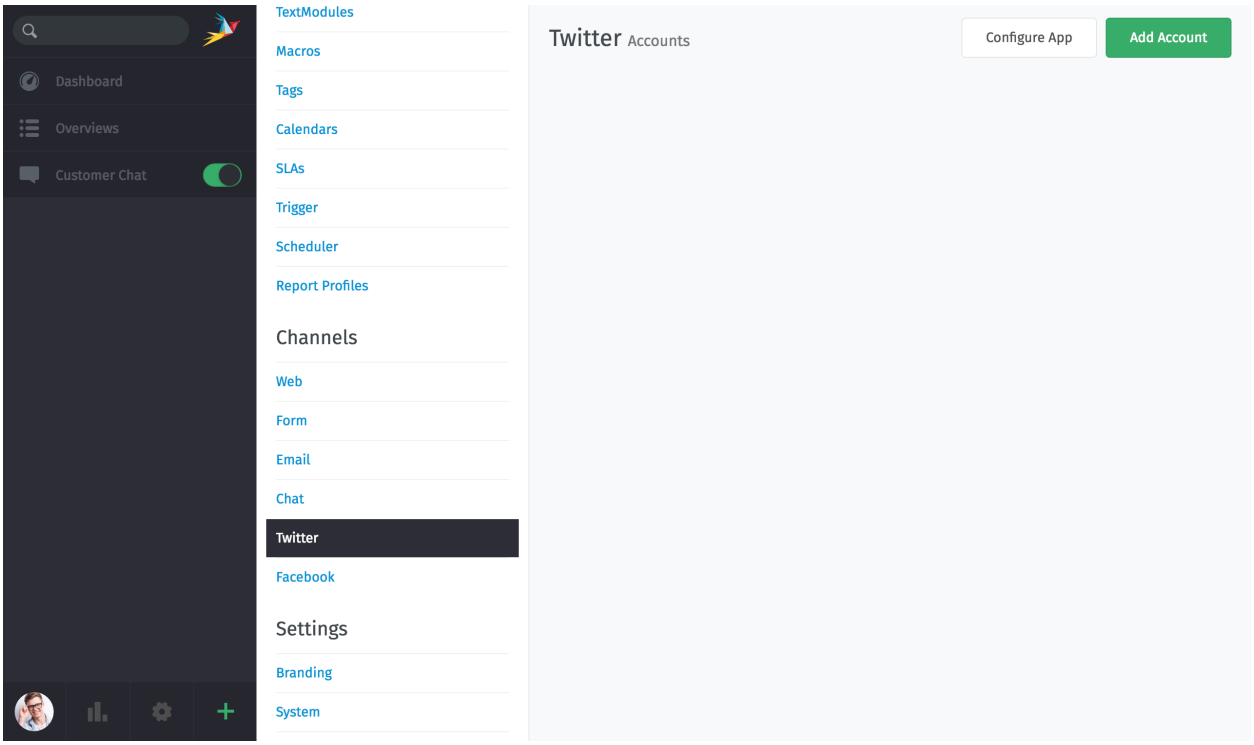


Fig. 8: Click **Add Account** under **Channels > Twitter** in the admin settings panel.

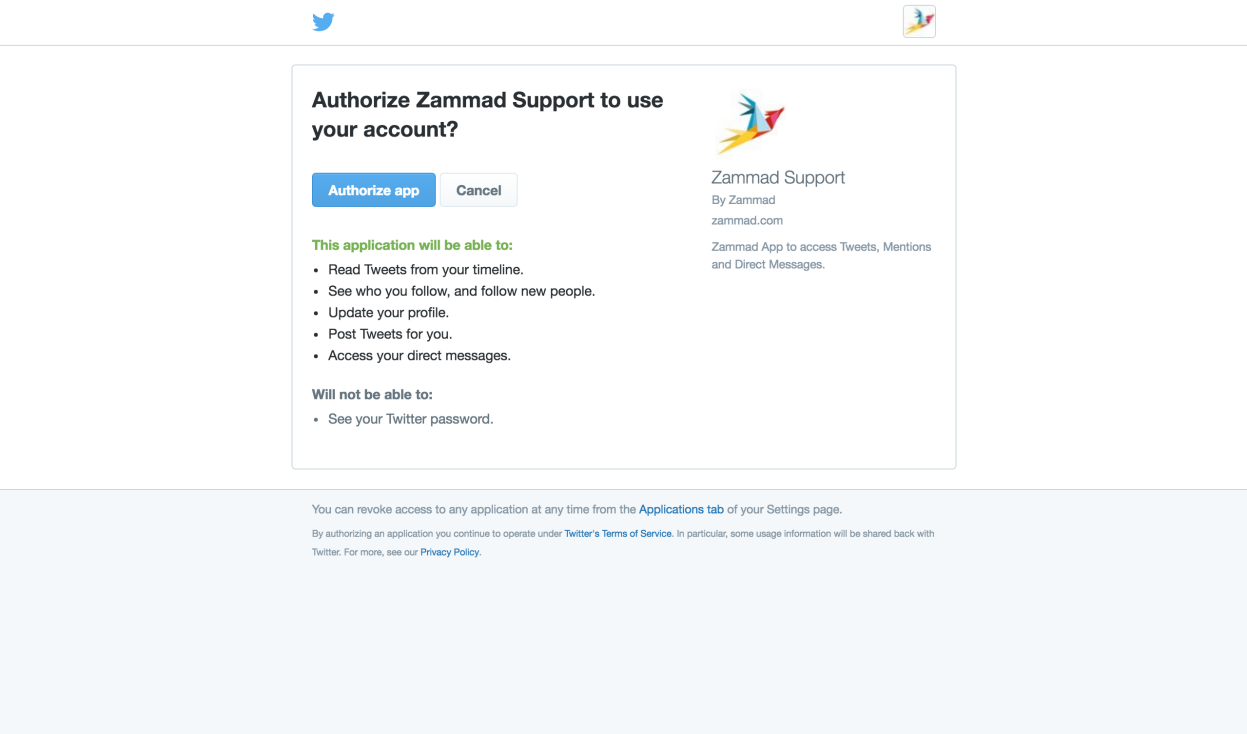


Fig. 9: You will be redirected to Twitter and asked to authorize Zammad to access your account.

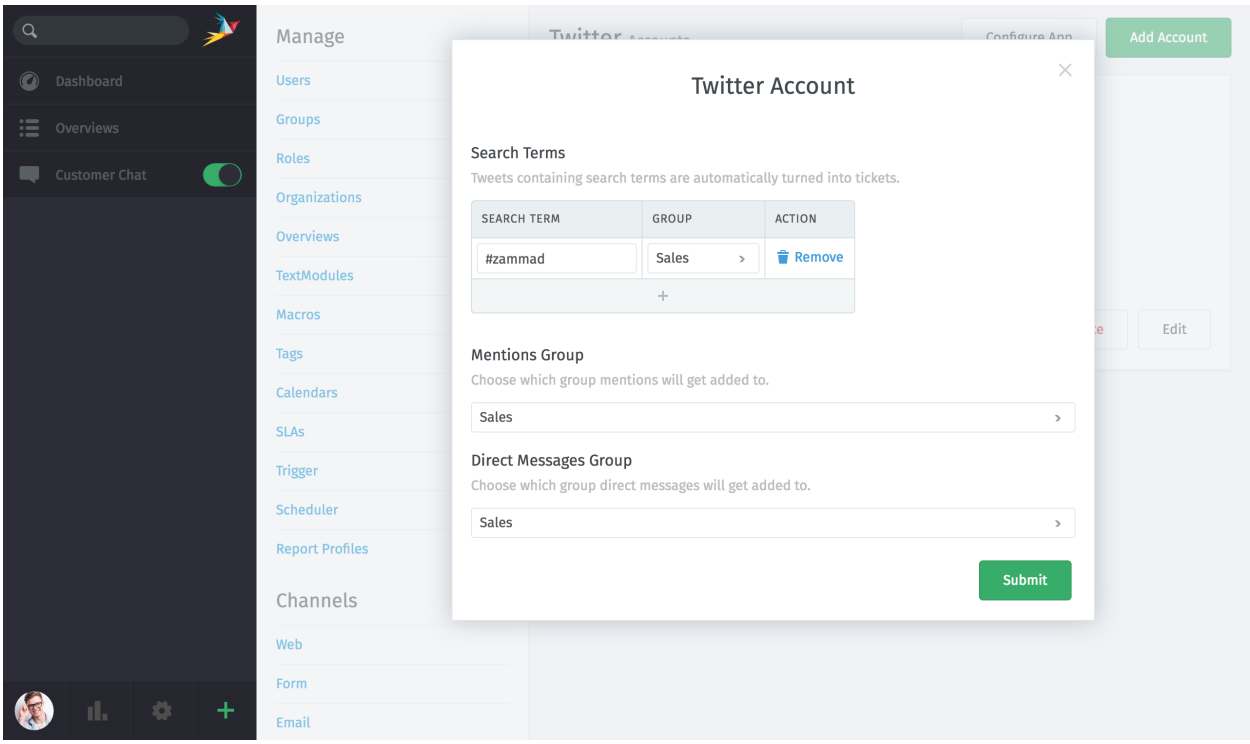


Fig. 10: Set up filters to automatically create new tickets based on **#tags** or **@mentions**.

CHAPTER

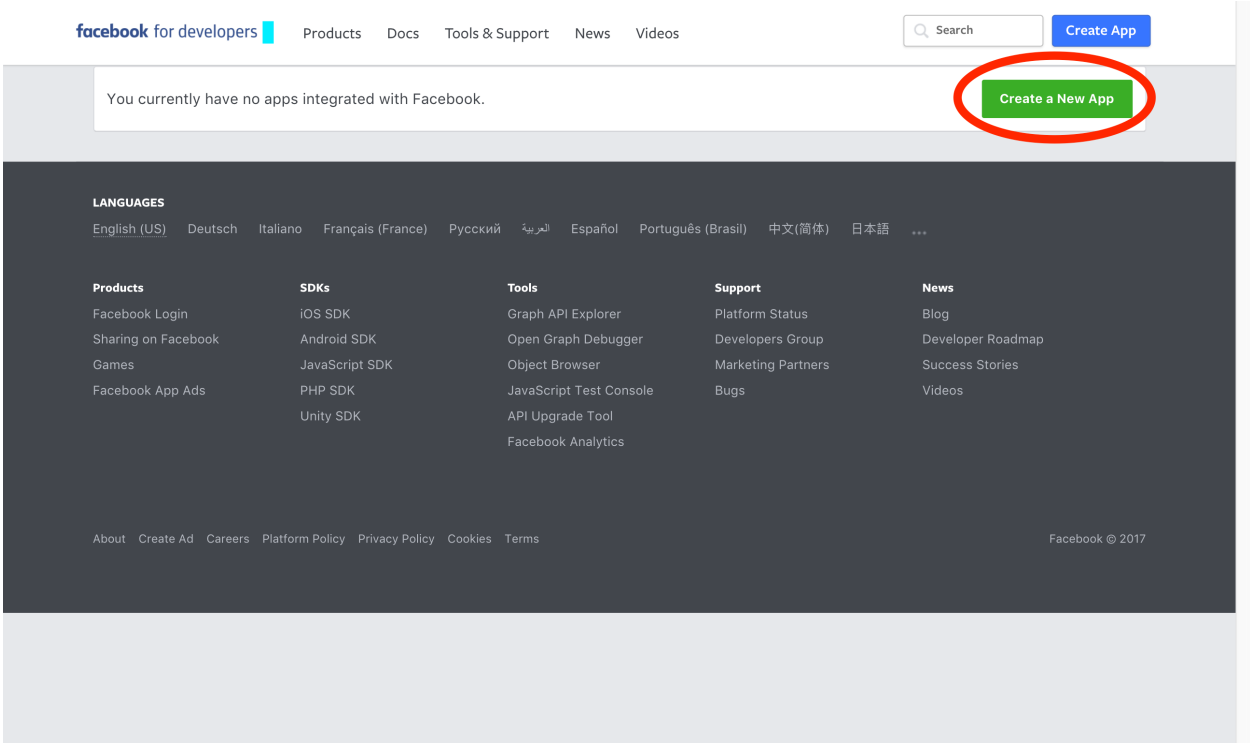
TWENTYSIX

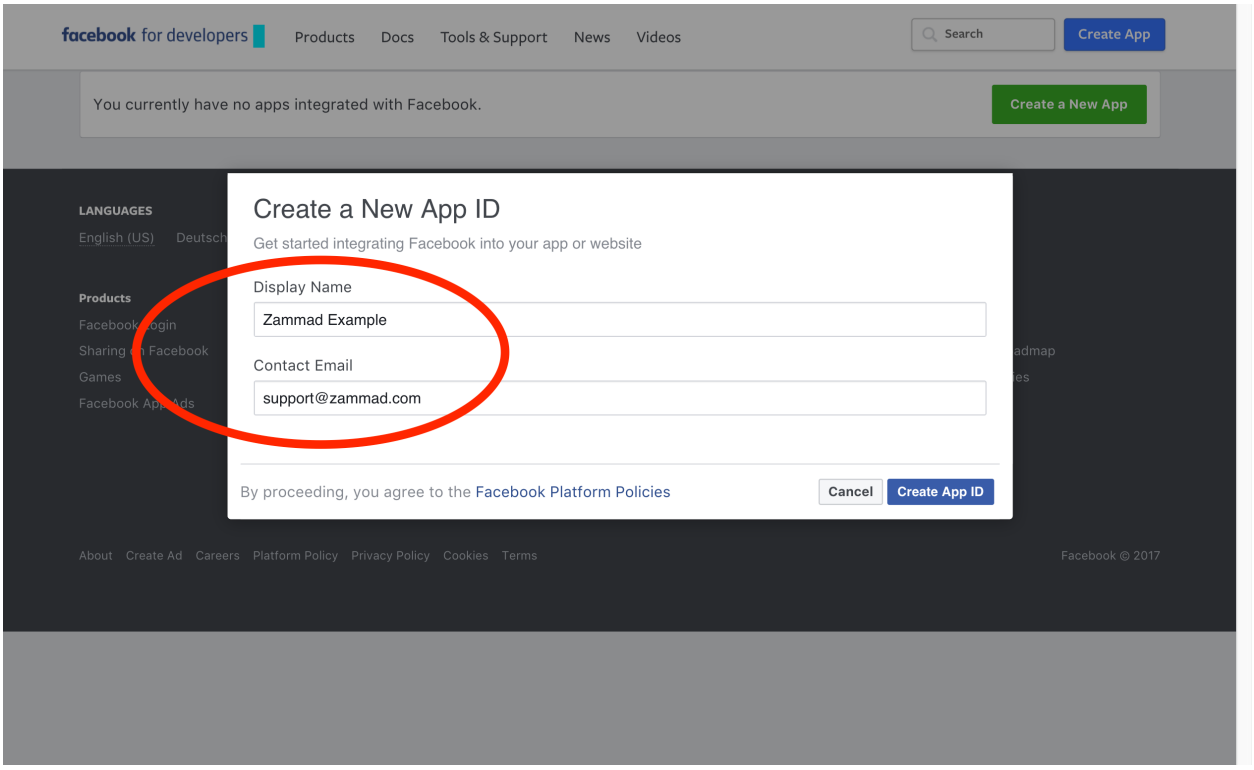
FACEBOOK

Hint: Please note that this part of our documentation currently is outdated. We currently are working on solutions for this topic.

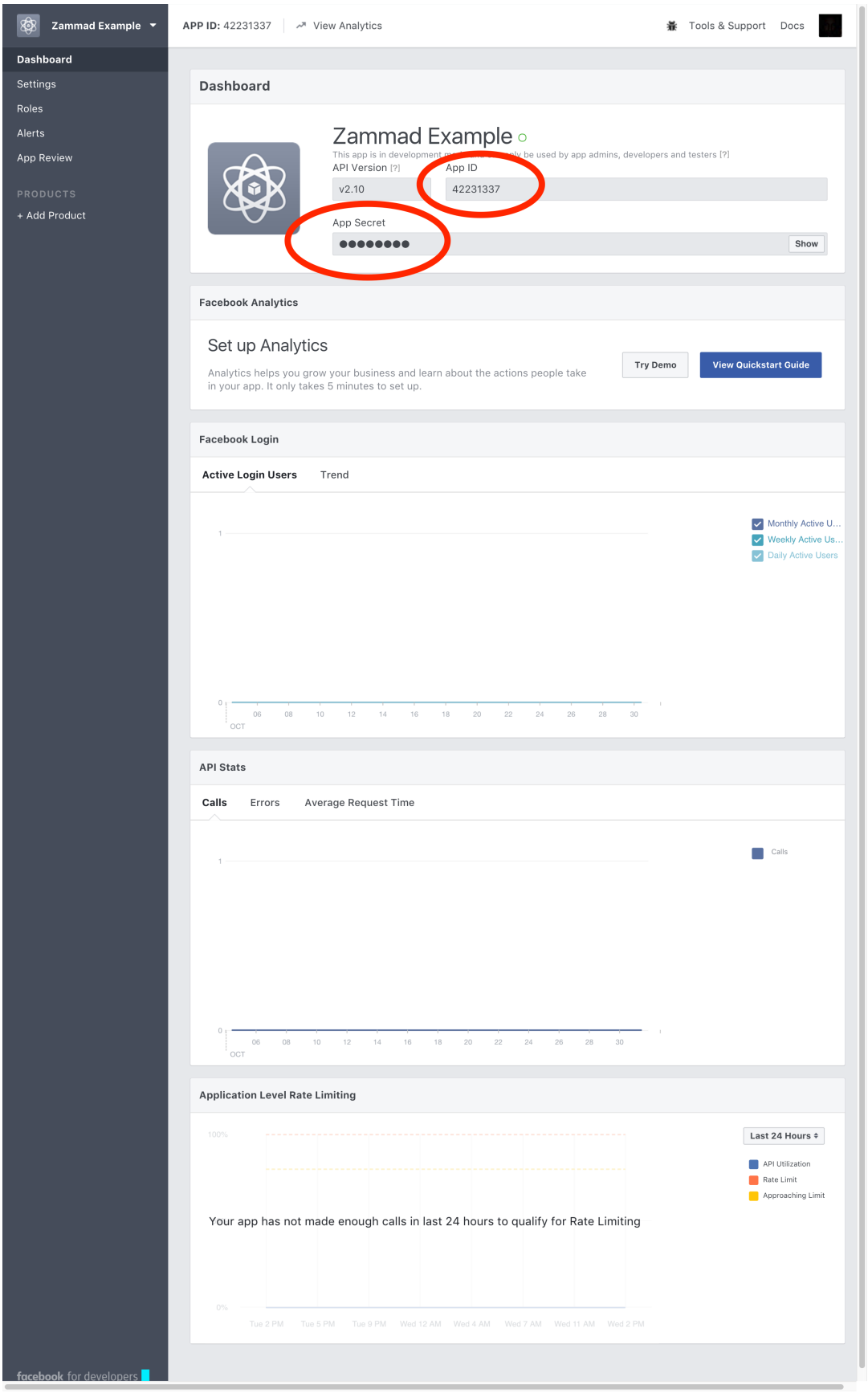
You can connect Facebook Accounts with Zammad. You need to connect your Zammad with Facebook first:

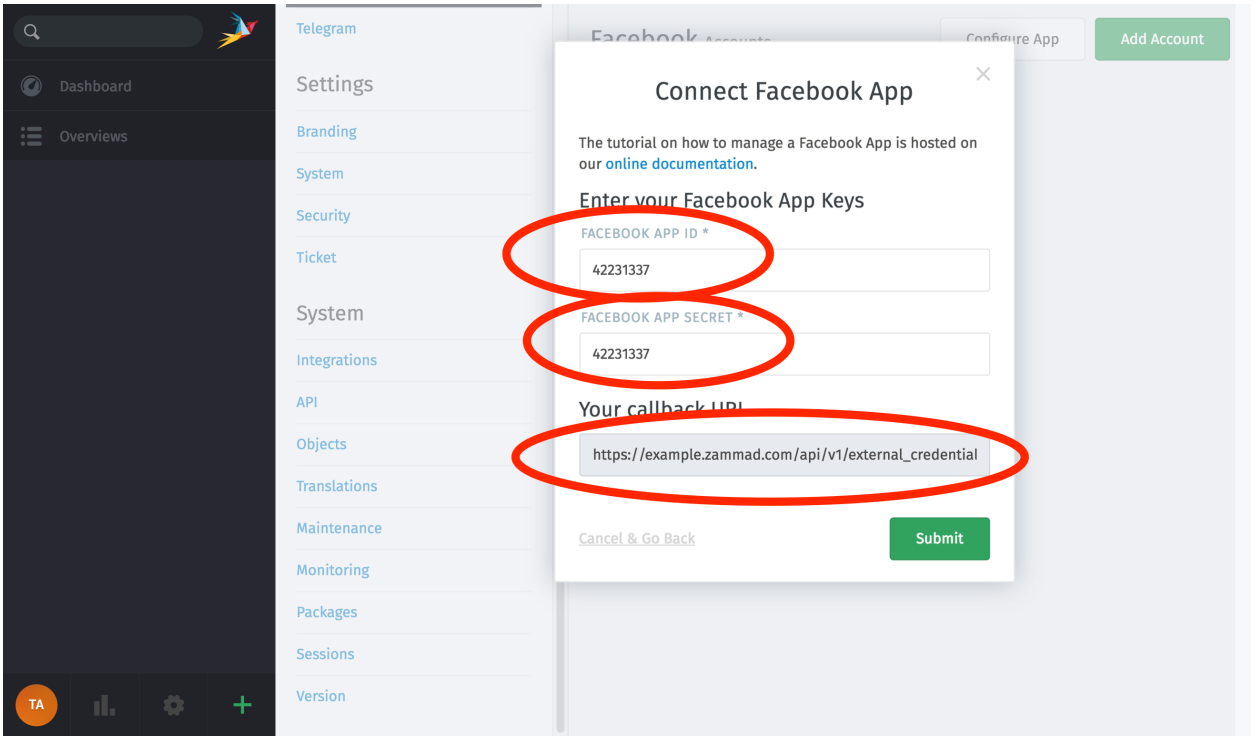
For this start at: https://developers.facebook.com/apps/

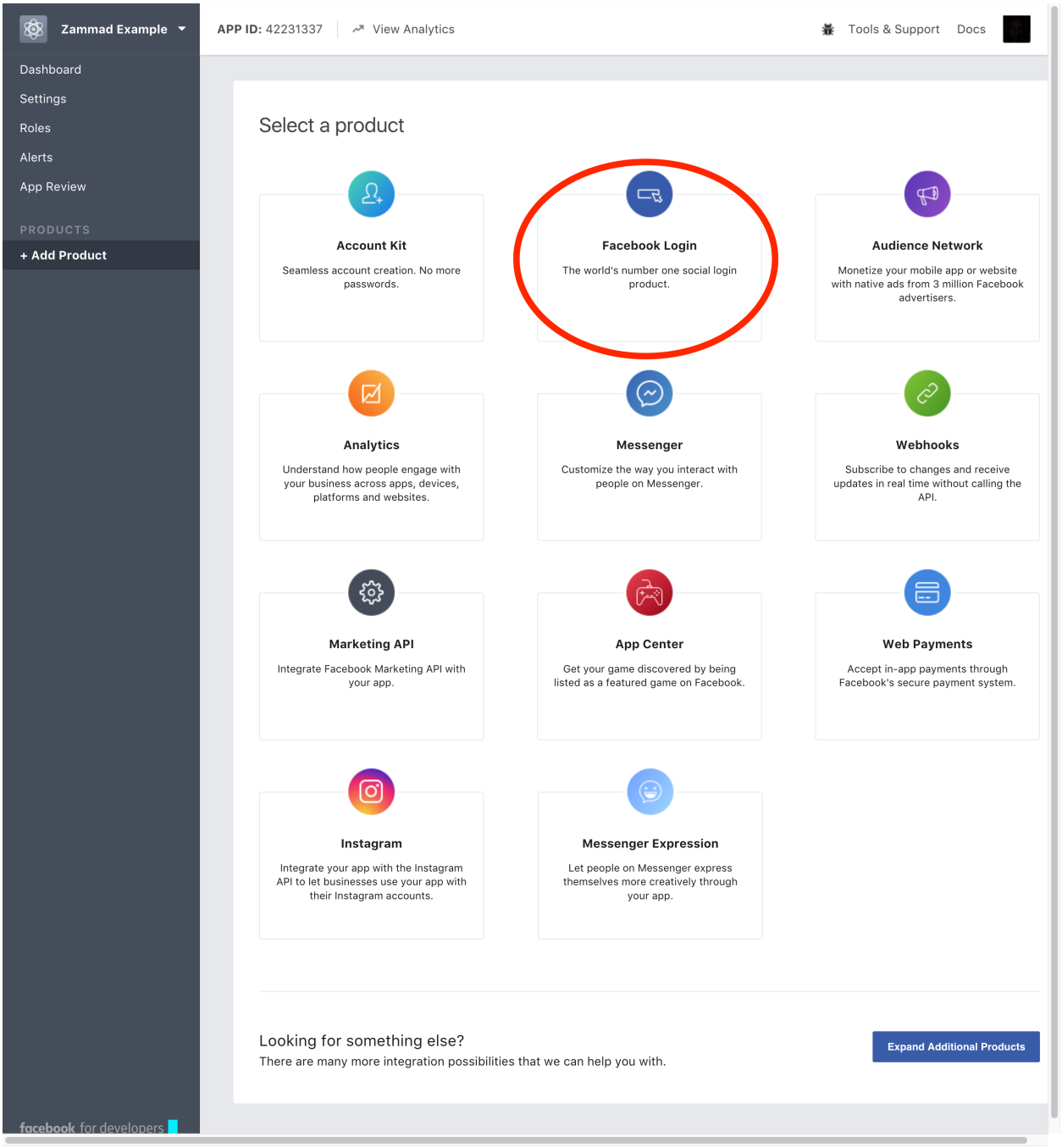


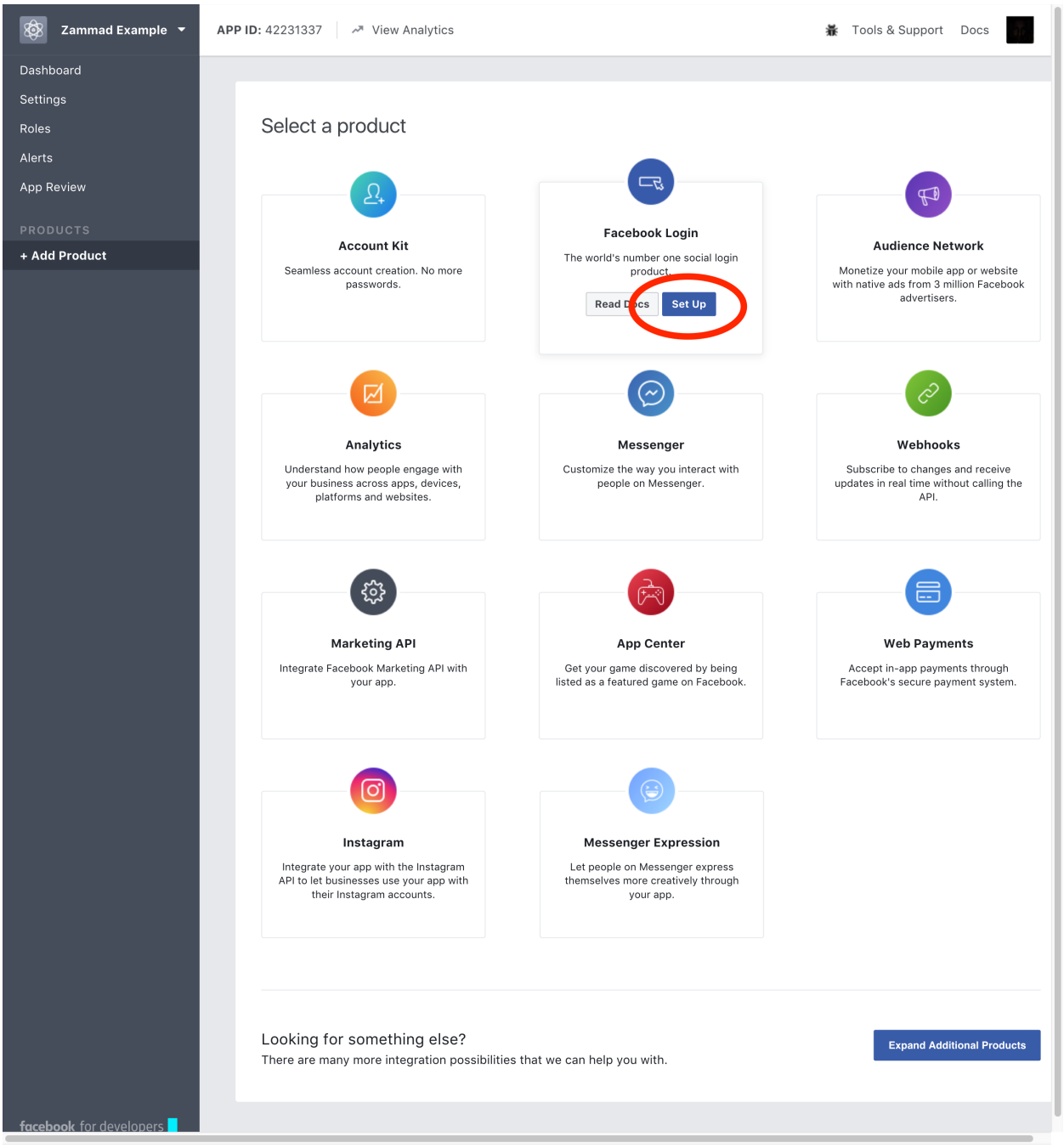


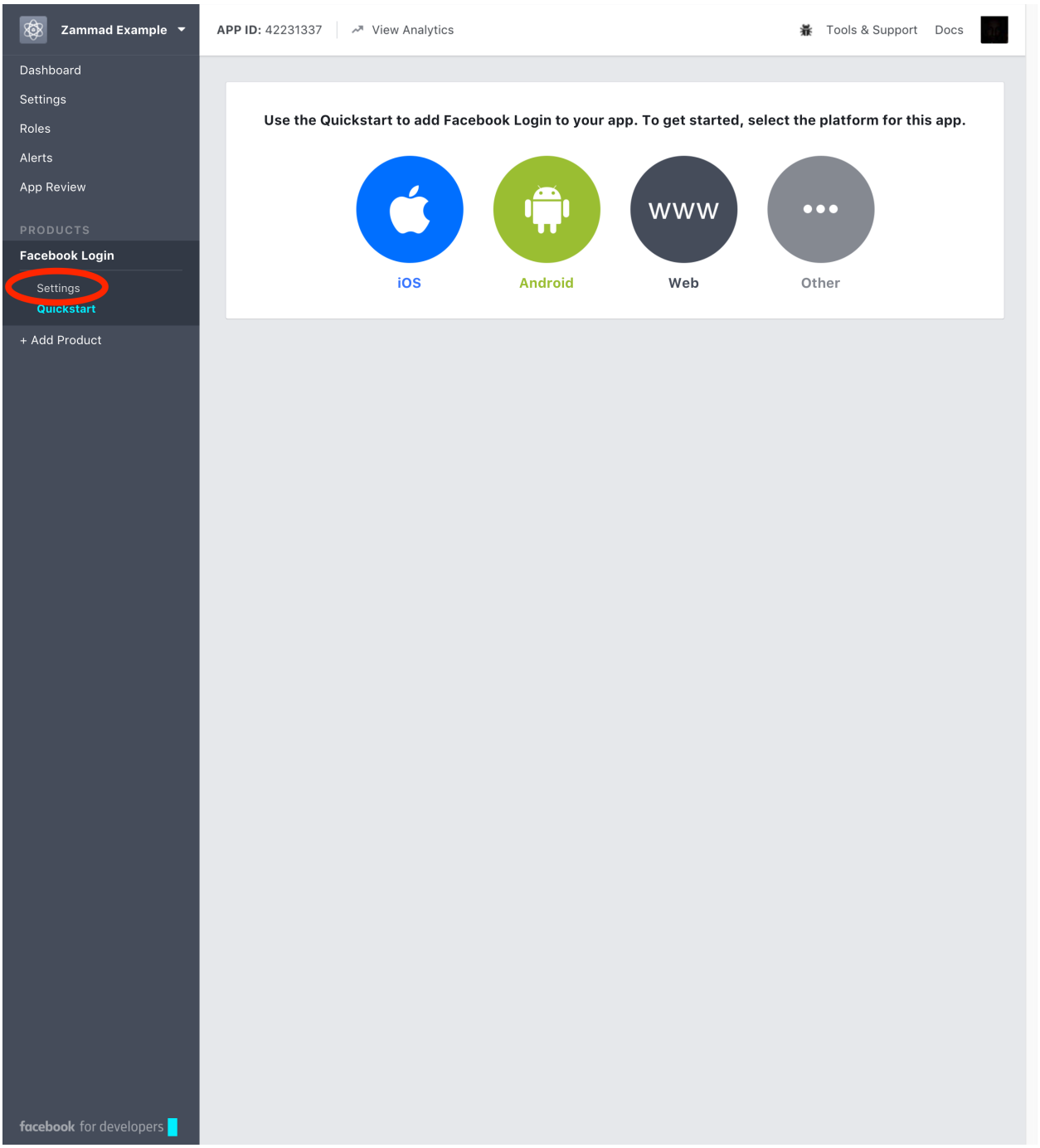
Click on “Create App ID” and enter app name

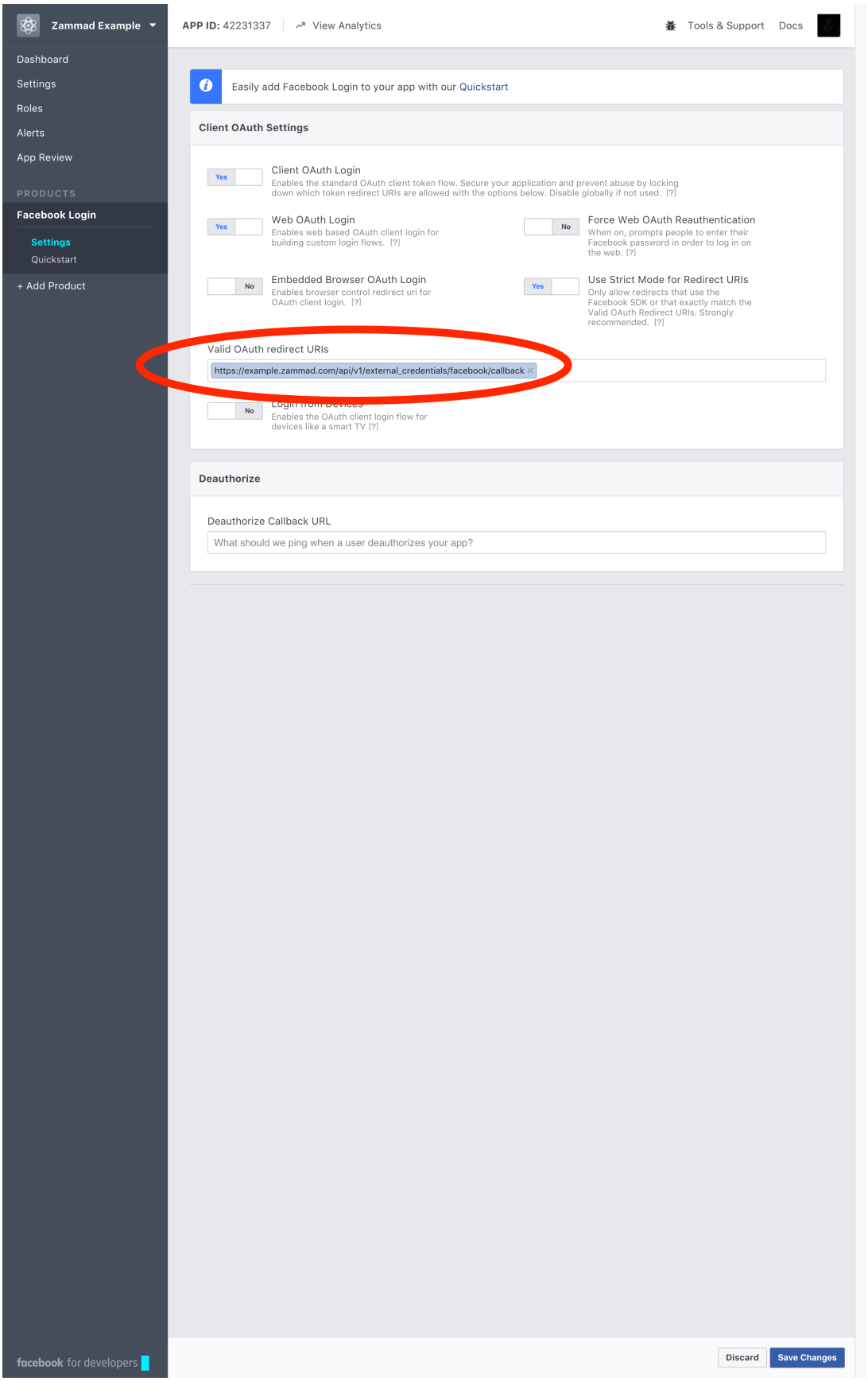

Zammad Example
APP ID: 42231337
View Analytics
Tools & Support
Docs
Dashboard
Settings
Roles
Alerts
App Review
PRODUCTS
+ Add Product
Dashboard
Zammad Example
This app is in development mode and can only be used by app admins, developers and testers [?]
API Version [?]
v2.10
App ID
42231337
App Secret
Show
Facebook Analytics
Set up Analytics
Analytics helps you grow your business and learn about the actions people take in your app. It only takes 5 minutes to set up.
Try Demo
View Quickstart Guide
Facebook Login
Active Login Users
Trend
Monthly Active U...
Weekly Active Us...
Daily Active Users
OCT
API Stats
Calls
Errors
Average Request Time
Calls
Application Level Rate Limiting
Last 24 Hours
API Utilization
Rate Limit
Approaching Limit
Your app has not made enough calls in last 24 hours to qualify for Rate Limiting
facebook for developers

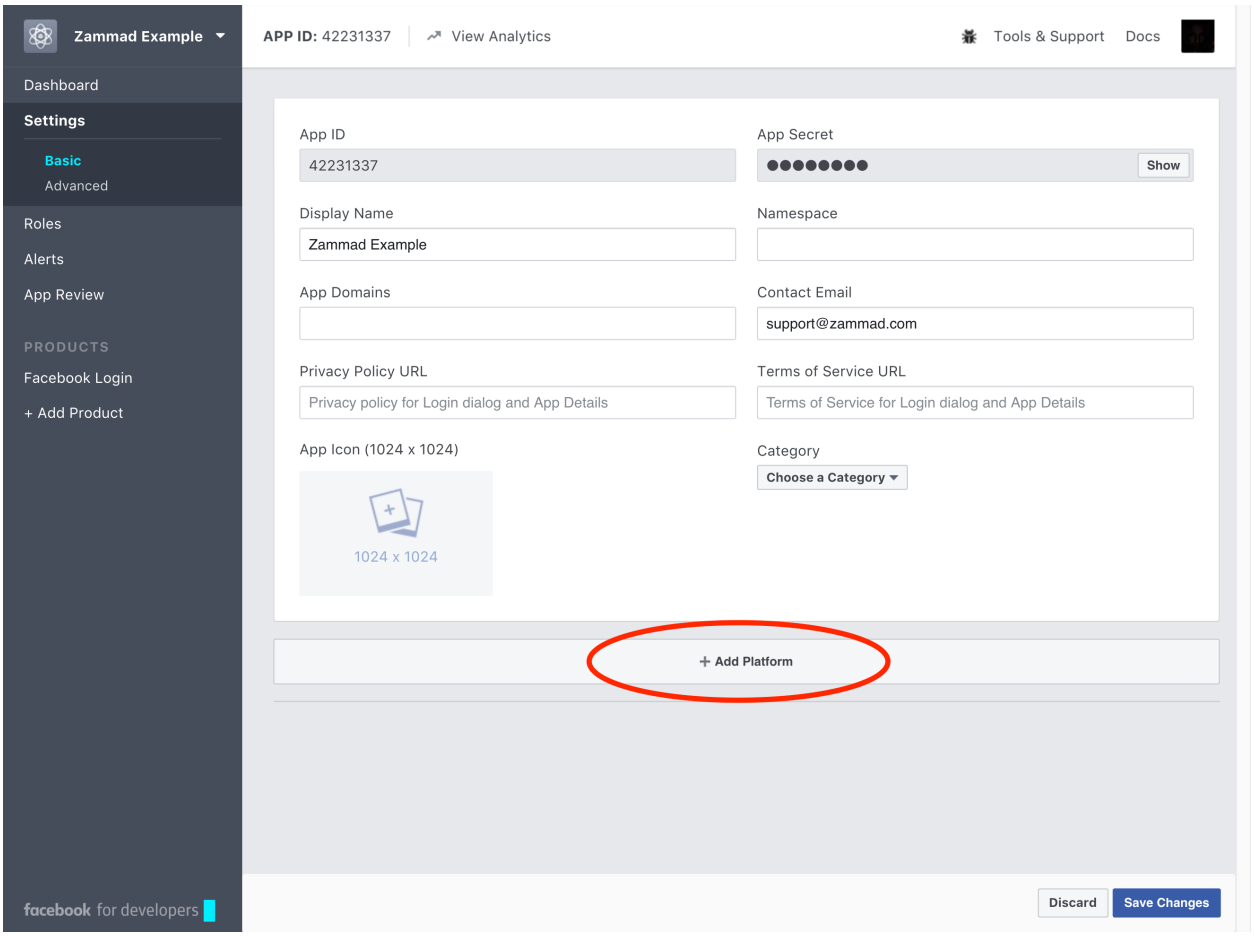

Dashboard
Overviews
TA
Telegram
Settings
Branding
System
Security
Ticket
System
Integrations
API
Objects
Translations
Maintenance
Monitoring
Packages
Sessions
Version
Configure App
Add Account
Connect Facebook App
The tutorial on how to manage a Facebook App is hosted on our online documentation.
Enter your Facebook App Keys
FACEBOOK APP ID *
42231337
FACEBOOK APP SECRET *
42231337
Your callback URL
https://example.zammad.com/api/v1/external_credential
Cancel & Go Back
Submit

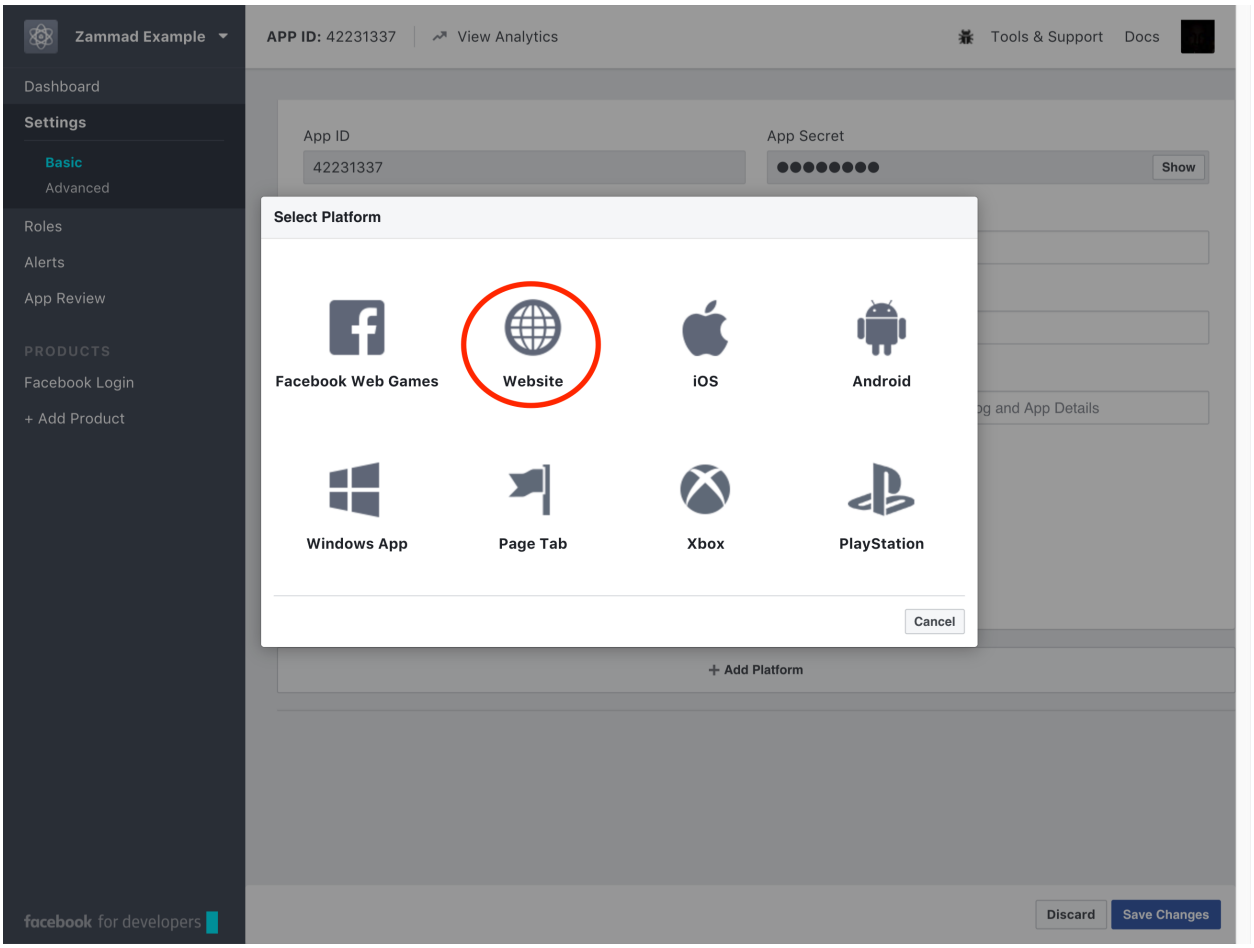

Zammad Example
APP ID: 42231337
View Analytics
Tools & Support
Docs
Dashboard
Settings
Roles
Alerts
App Review
PRODUCTS
+ Add Product
Select a product
Account Kit
Seamless account creation. No more passwords.
Facebook Login
The world's number one social login product.
Audience Network
Monetize your mobile app or website with native ads from 3 million Facebook advertisers.
Analytics
Understand how people engage with your business across apps, devices, platforms and websites.
Messenger
Customize the way you interact with people on Messenger.
Webhooks
Subscribe to changes and receive updates in real time without calling the API.
Marketing API
Integrate Facebook Marketing API with your app.
App Center
Get your game discovered by being listed as a featured game on Facebook.
Web Payments
Accept in-app payments through Facebook's secure payment system.
Instagram
Integrate your app with the Instagram API to let businesses use your app with their Instagram accounts.
Messenger Expression
Let people on Messenger express themselves more creatively through your app.
Looking for something else?
There are many more integration possibilities that we can help you with.
Expand Additional Products
facebook for developers

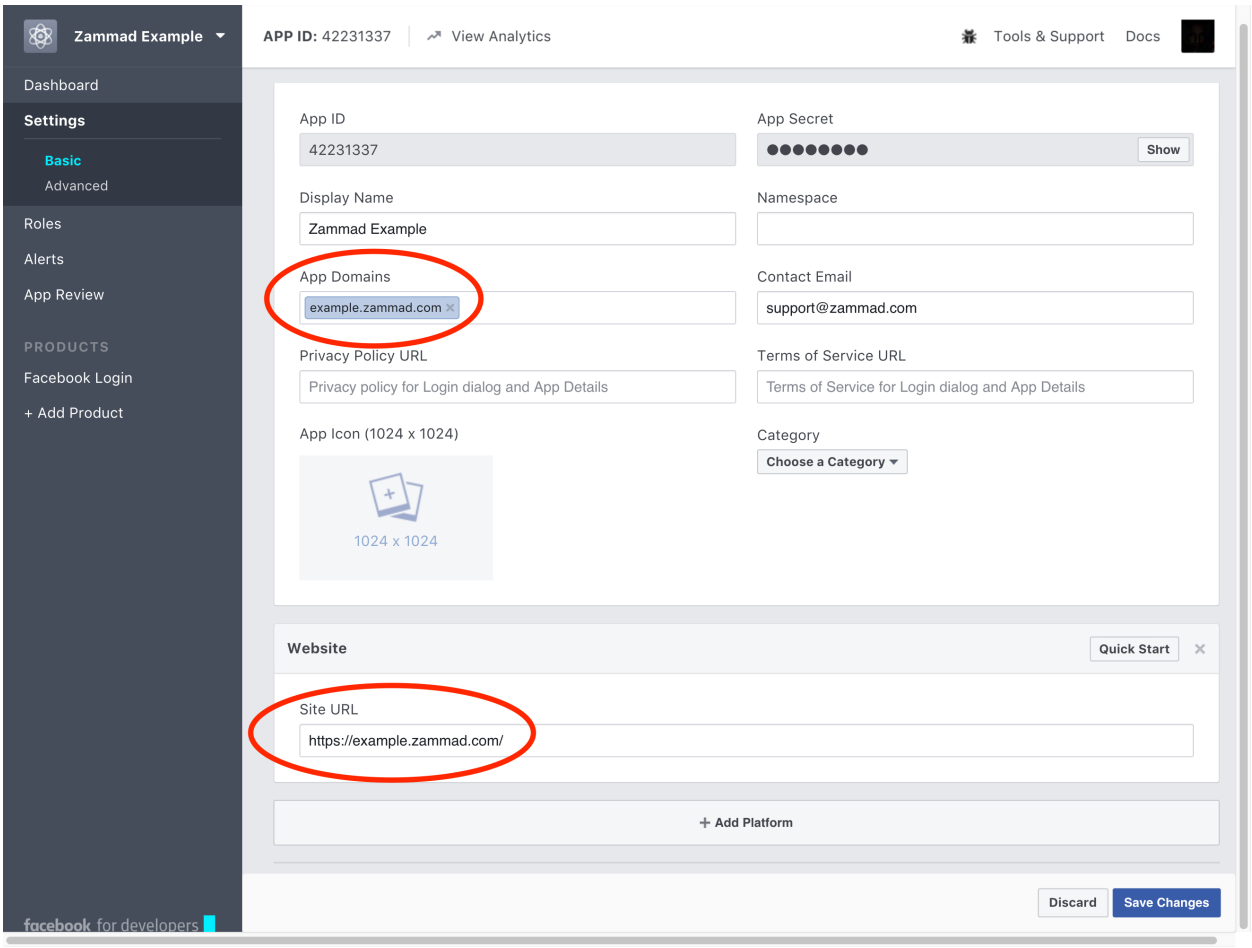

Zammad Example
APP ID: 42231337
View Analytics
Tools & Support
Docs
Dashboard
Settings
Roles
Alerts
App Review
PRODUCTS
+ Add Product
Select a product
Account Kit
Seamless account creation. No more passwords.
Facebook Login
The world's number one social login product.
Read Docs
Set Up
Audience Network
Monetize your mobile app or website with native ads from 3 million Facebook advertisers.
Analytics
Understand how people engage with your business across apps, devices, platforms and websites.
Messenger
Customize the way you interact with people on Messenger.
Webhooks
Subscribe to changes and receive updates in real time without calling the API.
Marketing API
Integrate Facebook Marketing API with your app.
App Center
Get your game discovered by being listed as a featured game on Facebook.
Web Payments
Accept in-app payments through Facebook's secure payment system.
Instagram
Integrate your app with the Instagram API to let businesses use your app with their Instagram accounts.
Messenger Expression
Let people on Messenger express themselves more creatively through your app.
Looking for something else?
There are many more integration possibilities that we can help you with.
Expand Additional Products
facebook for developers


Zammad Example
APP ID: 42231337
View Analytics
Tools & Support
Docs
Dashboard
Settings
Roles
Alerts
App Review
PRODUCTS
Facebook Login
Settings
Quickstart
+ Add Product
Use the Quickstart to add Facebook Login to your app. To get started, select the platform for this app.
iOS
Android
Web
Other
facebook for developers


Zammad Example
APP ID: 42231337
View Analytics
Tools & Support
Docs
Dashboard
Settings
Roles
Alerts
App Review
PRODUCTS
Facebook Login
Settings
Quickstart
+ Add Product
Easily add Facebook Login to your app with our Quickstart
Client OAuth Settings
Client OAuth Login
Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. [?]
Web OAuth Login
Enables web based OAuth client login for building custom login flows. [?]
Force Web OAuth Reauthentication
When on, prompts people to enter their Facebook password in order to log in on the web. [?]
Embedded Browser OAuth Login
Enables browser control redirect uri for OAuth client login. [?]
Use Strict Mode for Redirect URIs
Only allow redirects that use the Facebook SDK or that exactly match the Valid OAuth Redirect URIs. Strongly recommended. [?]
Valid OAuth redirect URIs
https://example.zammad.com/api/v1/external_credentials/facebook/callback
Enables the OAuth client login flow for devices like a smart TV [?]
Deauthorize
Deauthorize Callback URL
What should we ping when a user deauthorizes your app?
Discard
Save Changes
facebook for developers


Zammad Example
APP ID: 42231337
View Analytics
Tools & Support
Docs
Dashboard
Settings
Basic
Advanced
Roles
Alerts
App Review
PRODUCTS
Facebook Login
+ Add Product
App ID
42231337
App Secret
Show
Display Name
Zammad Example
Namespace
App Domains
Contact Email
support@zammad.com
Privacy Policy URL
Privacy policy for Login dialog and App Details
Terms of Service URL
Terms of Service for Login dialog and App Details
App Icon (1024 x 1024)
1024 x 1024
Category
Choose a Category
+ Add Platform
Discard
Save Changes
facebook for developers


Zammad Example
APP ID: 42231337
View Analytics
Tools & Support
Docs
Dashboard
Settings
Basic
Advanced
Roles
Alerts
App Review
PRODUCTS
Facebook Login
+ Add Product
App ID
42231337
App Secret
Show
Select Platform
Facebook Web Games
Website
iOS
Android
Windows App
Page Tab
Xbox
PlayStation
Cancel
+ Add Platform
Discard
Save Changes
facebook for developers



Configure Zammad as Facebook app

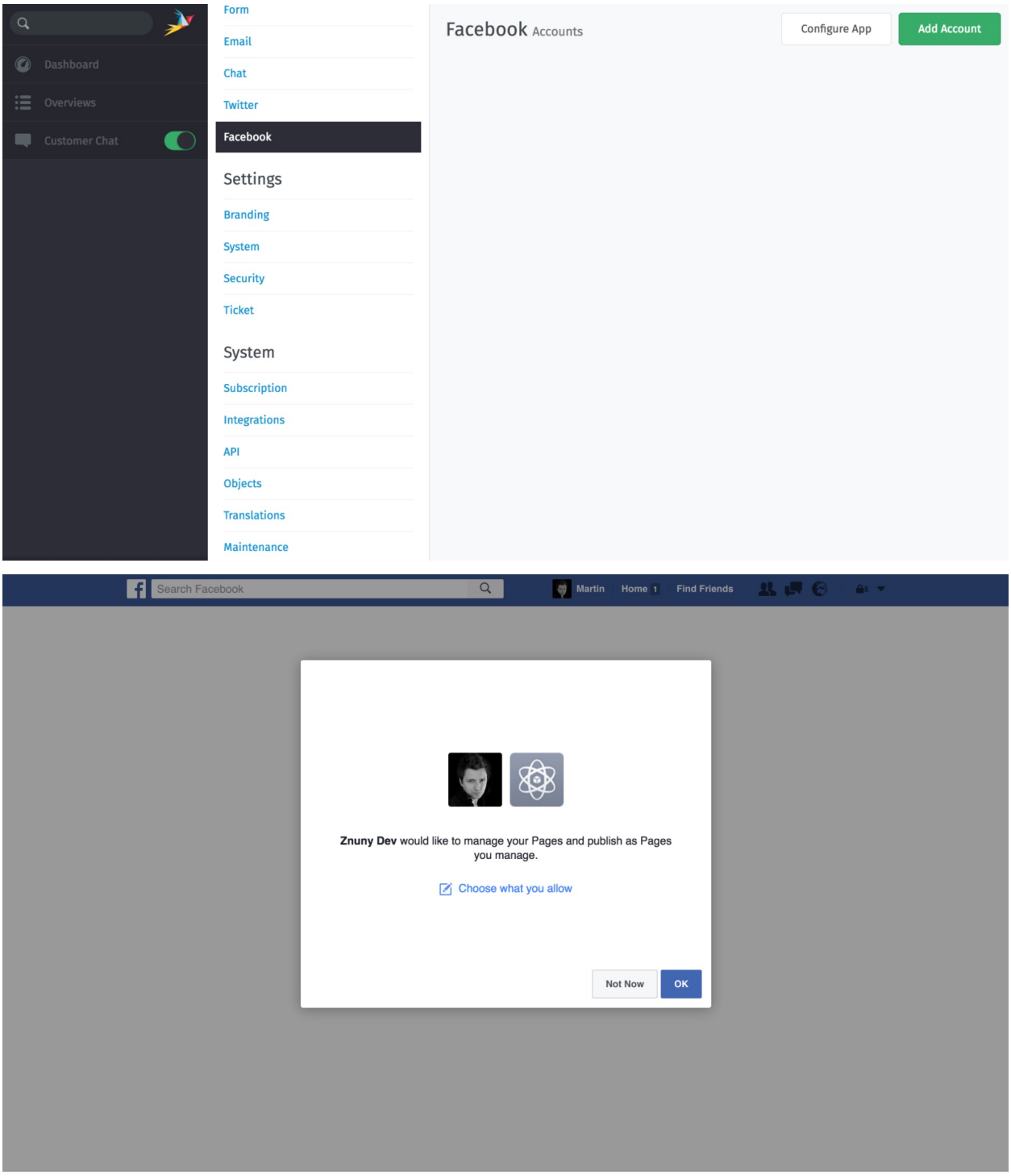
- Go to “Admin -> Channels -> Facebook”
- Click on “Connect Facebook App” and enter your “App ID”, “App Secret” and verify the “Callback URL”.

Done, your Zammad is configured as Facebook App now.

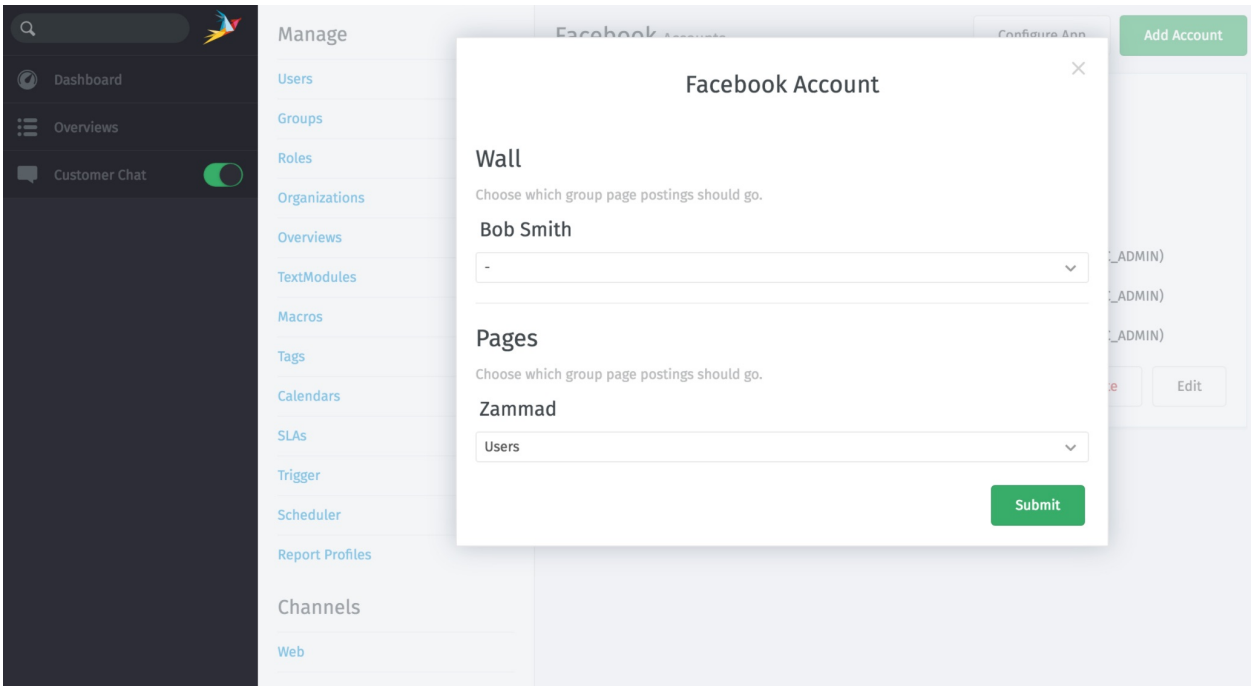
Link your Facebook Page to your Zammad Facebook app

Now you need to link your Facebook Page from which you want to get posts and send out comments.

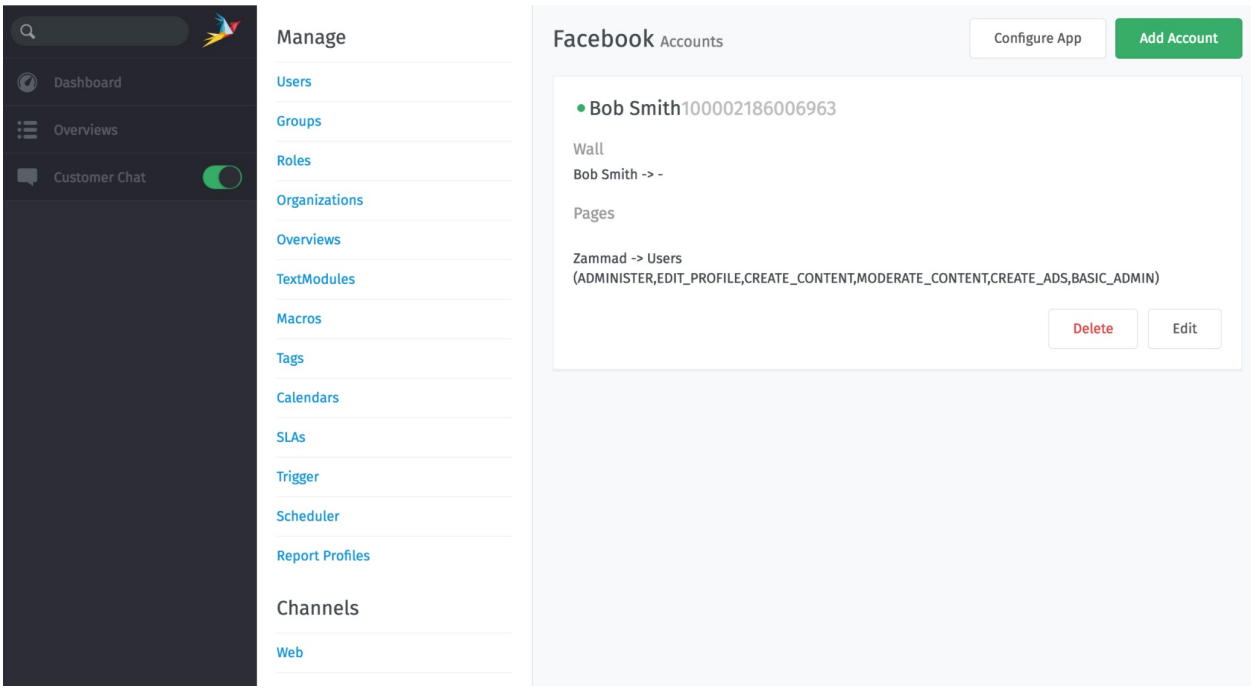
Click on “Add Account”, then you will see the authorize app page of Facebook. Click on “authorize app”.



You will get redirected back to Zammad. Now you need to configure your search keys, where mentions should get routed.



After you are done, you will get an overview of all linked Facebook Accounts.



Start using your new channel

Start and write a post to your page, short time later you will have a new ticket in Zammad.

Just click on the reply button (as you do it for emails) to send a comment.

CHAPTER

TWENTYSEVEN

TELEGRAM

It's possible to put your Telegram bot communication into Zammad. To do so, you need to follow these steps.

Note: Your Zammad instance needs to be publicly available via HTTPS (we use Telegram WebHooks).

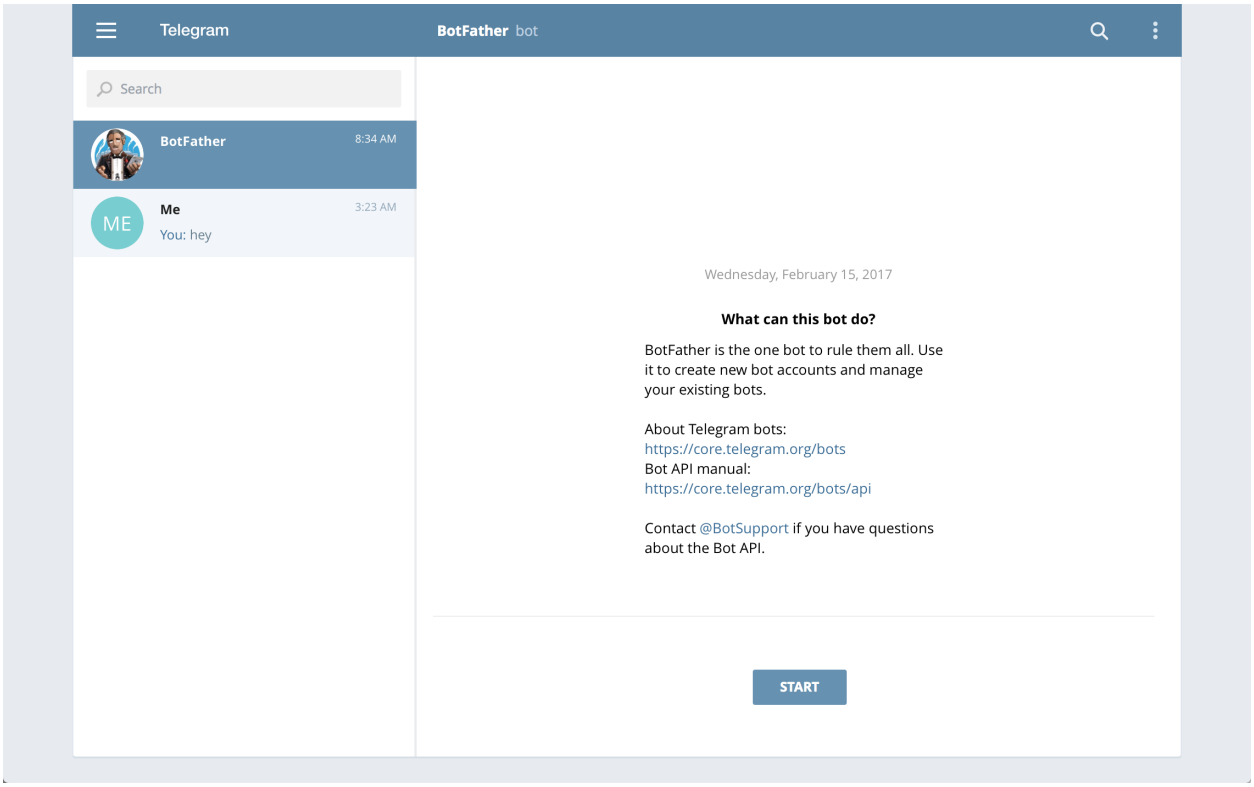
Warning: Zammad **cannot receive file attachments larger than 20MB** in Telegram messages.

This is a hard limit imposed by the Telegram Bot API.

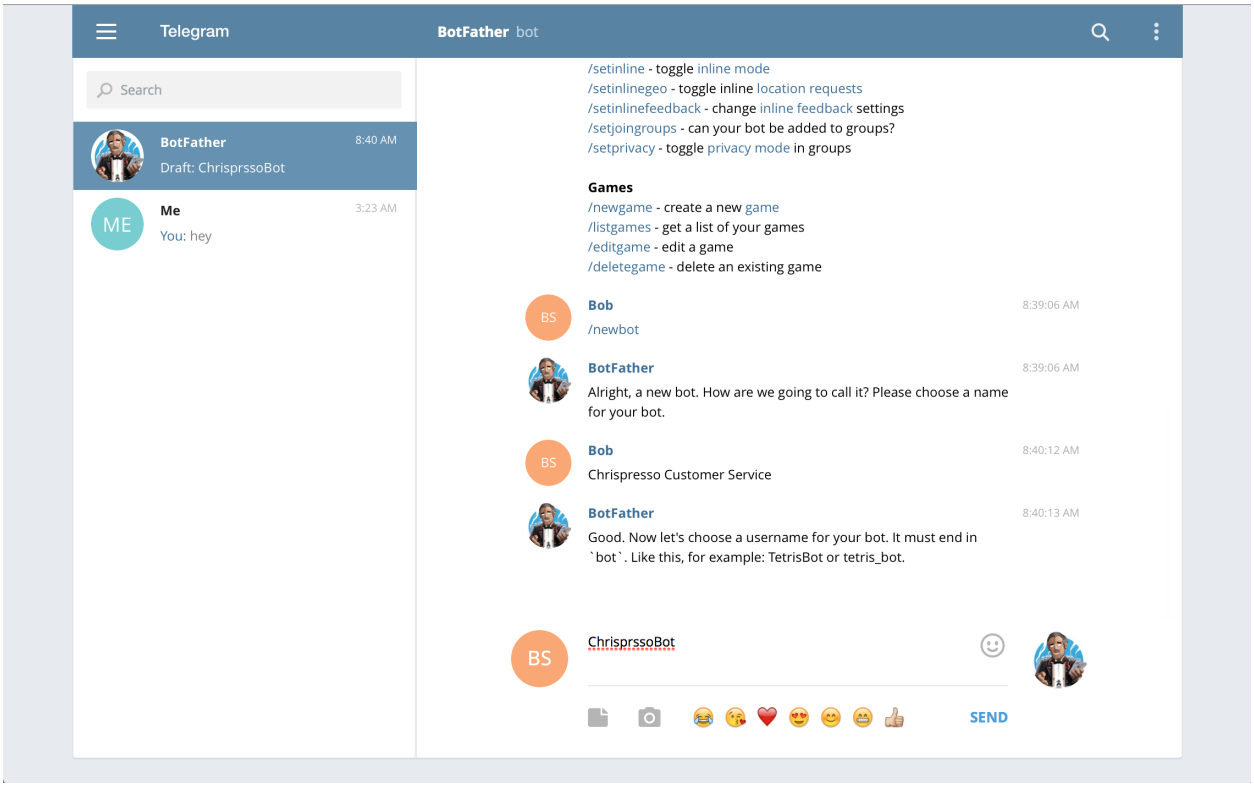
27.1 Register a Telegram bot app

Register your Telegram bot via your Telegram client, see also here: https://core.telegram.org/bots#3-how-do-i-create-a-bot

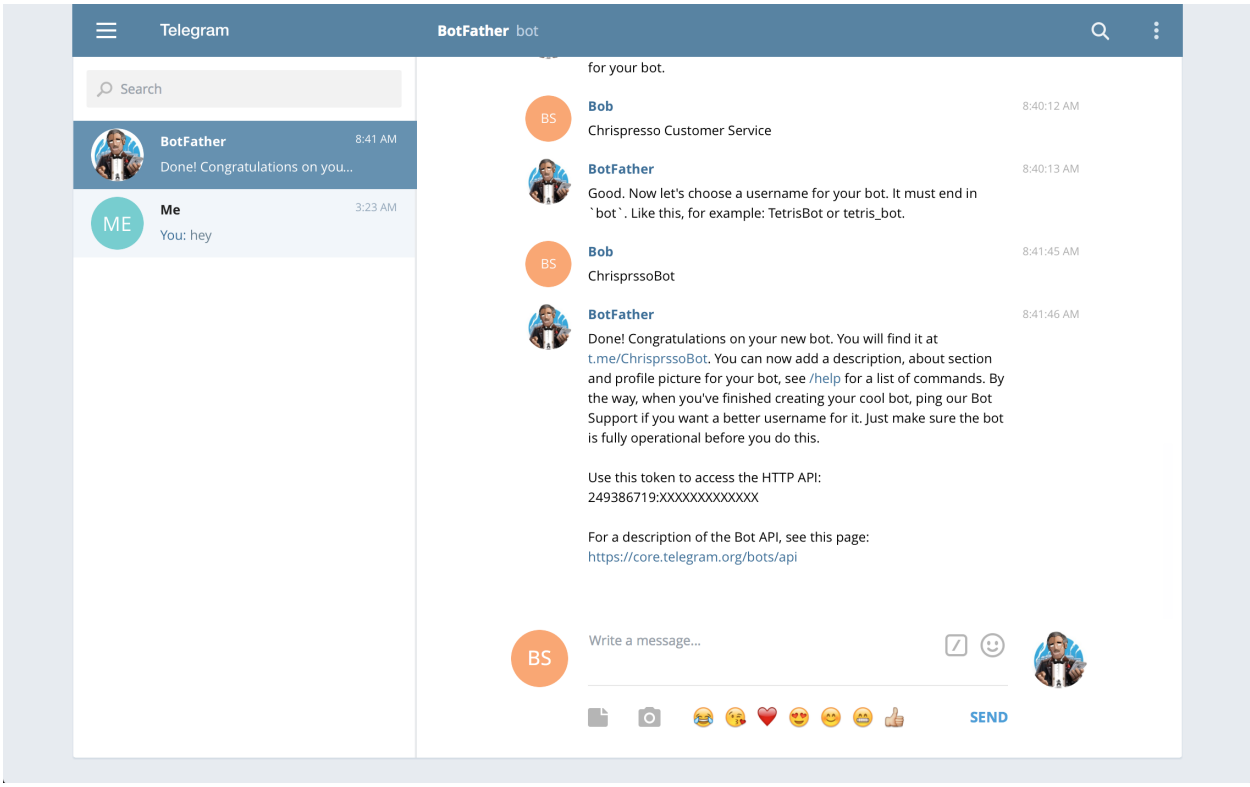
Go to BotFather



Register a new bot by using "/newbot" and define its name and username

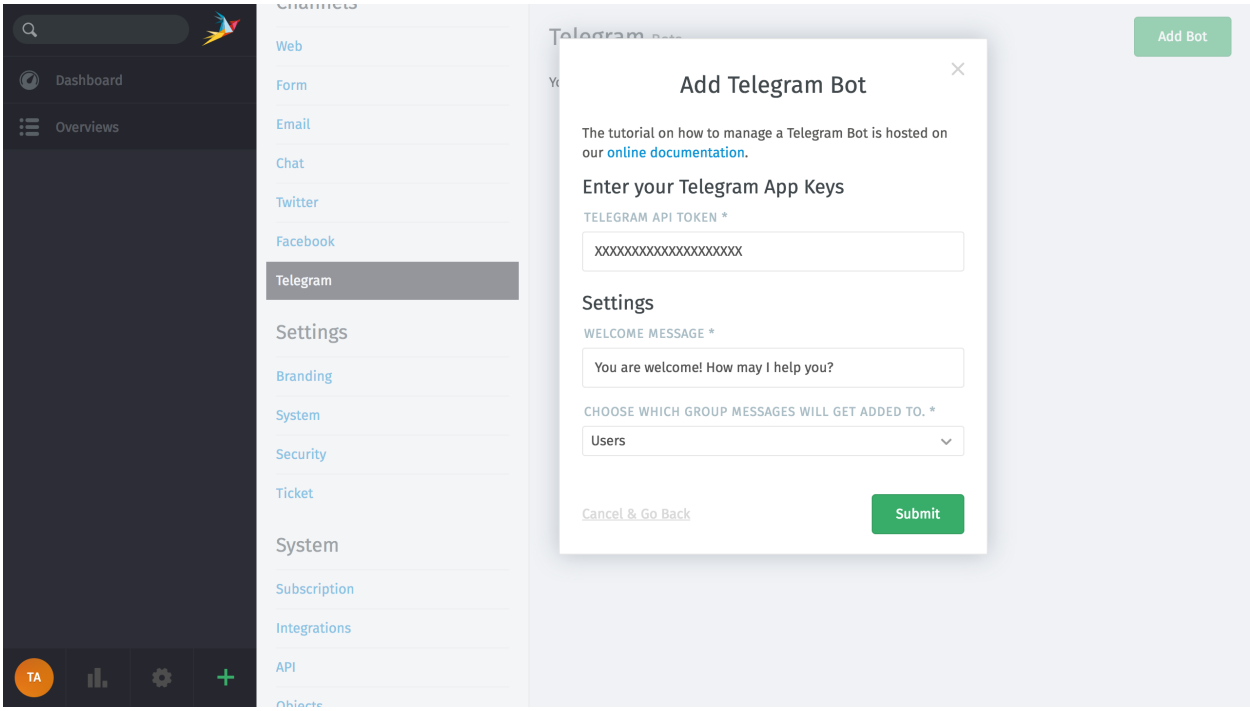


When you're all done, you will get your Telegram bot API token

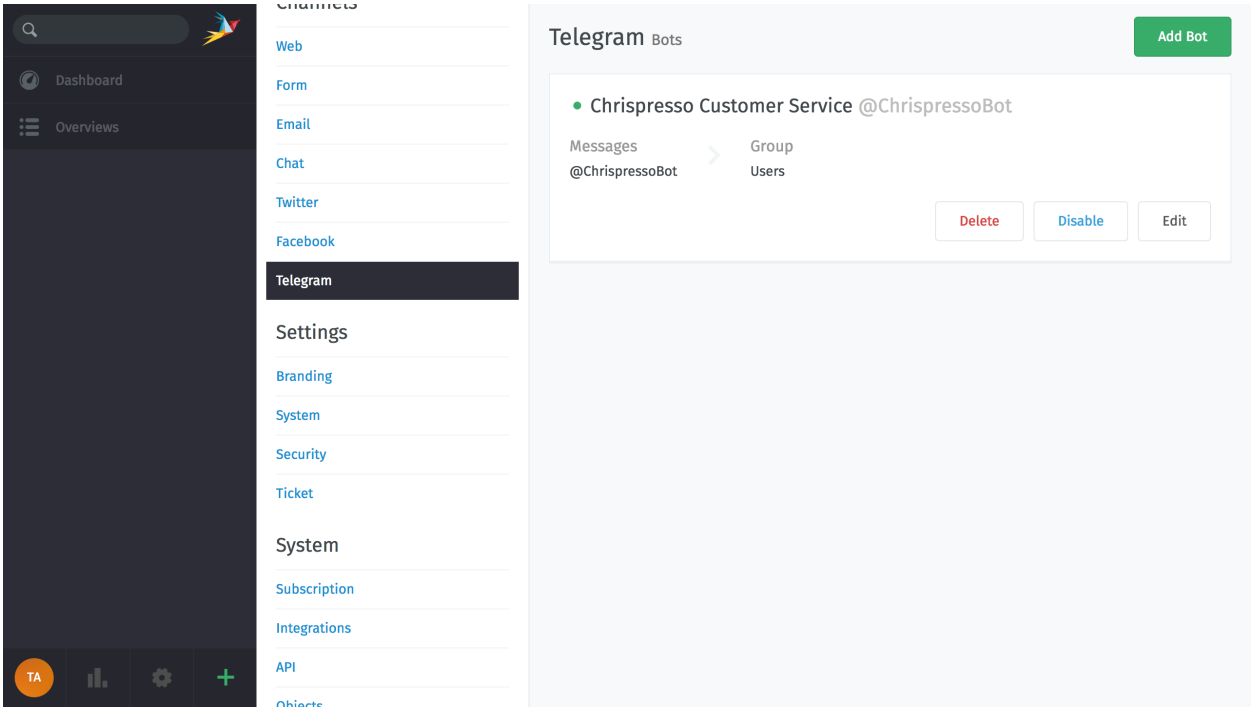


27.2 Configure Zammad as Telegram bot

Go to “Admin -> Channels -> Telegram” and click “Add Bot”



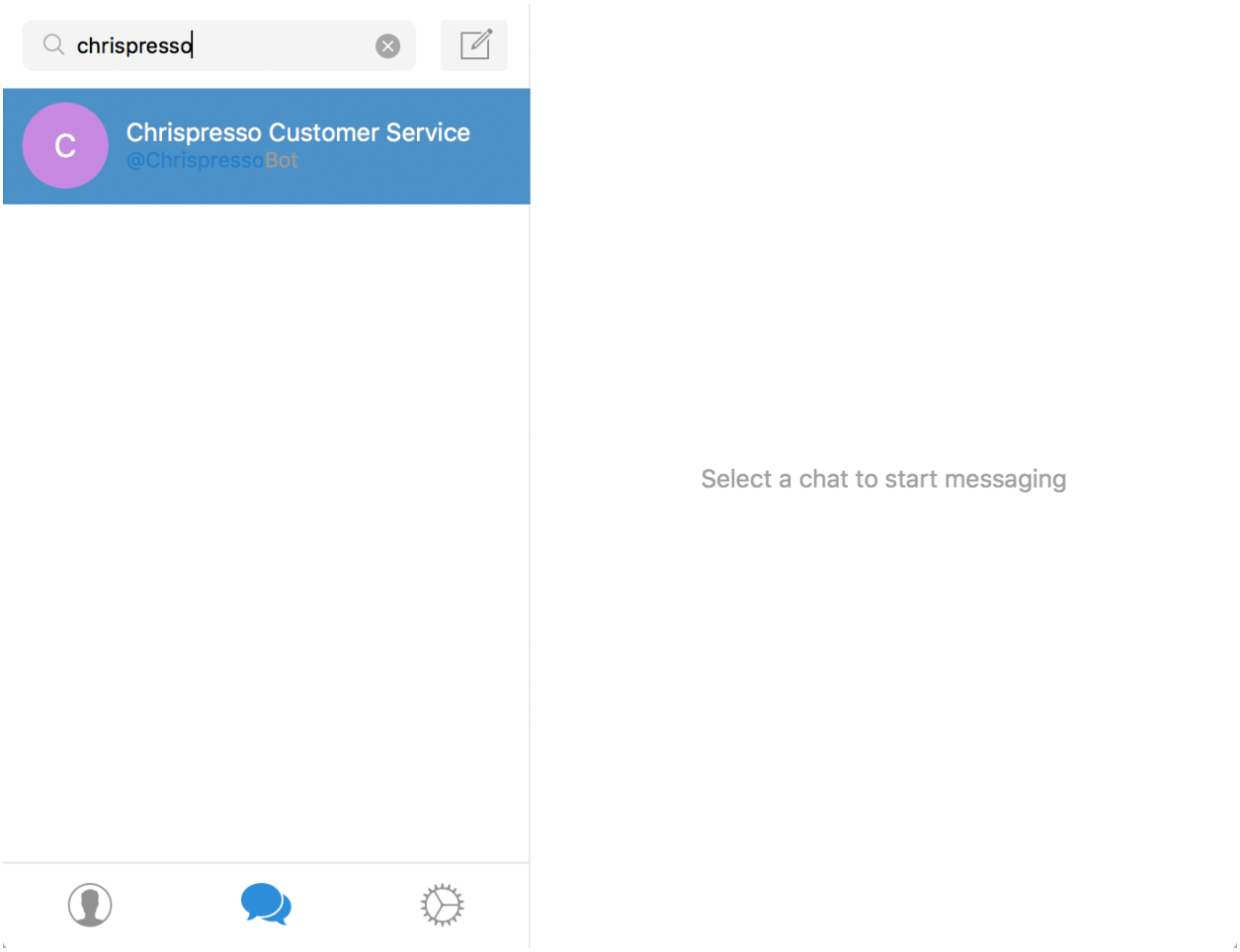
Enter your “API Token”, your “welcome message” and set the incoming group.

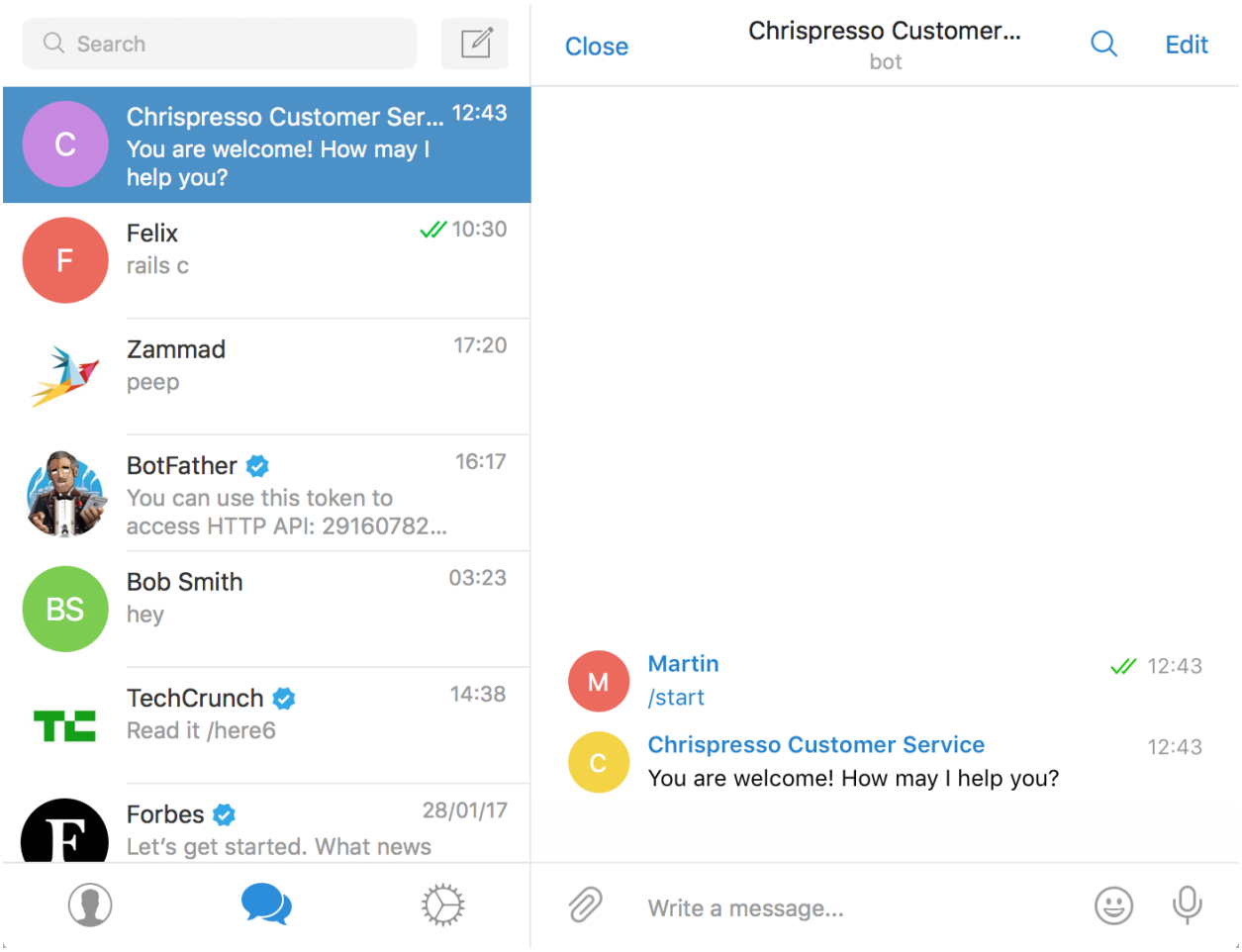


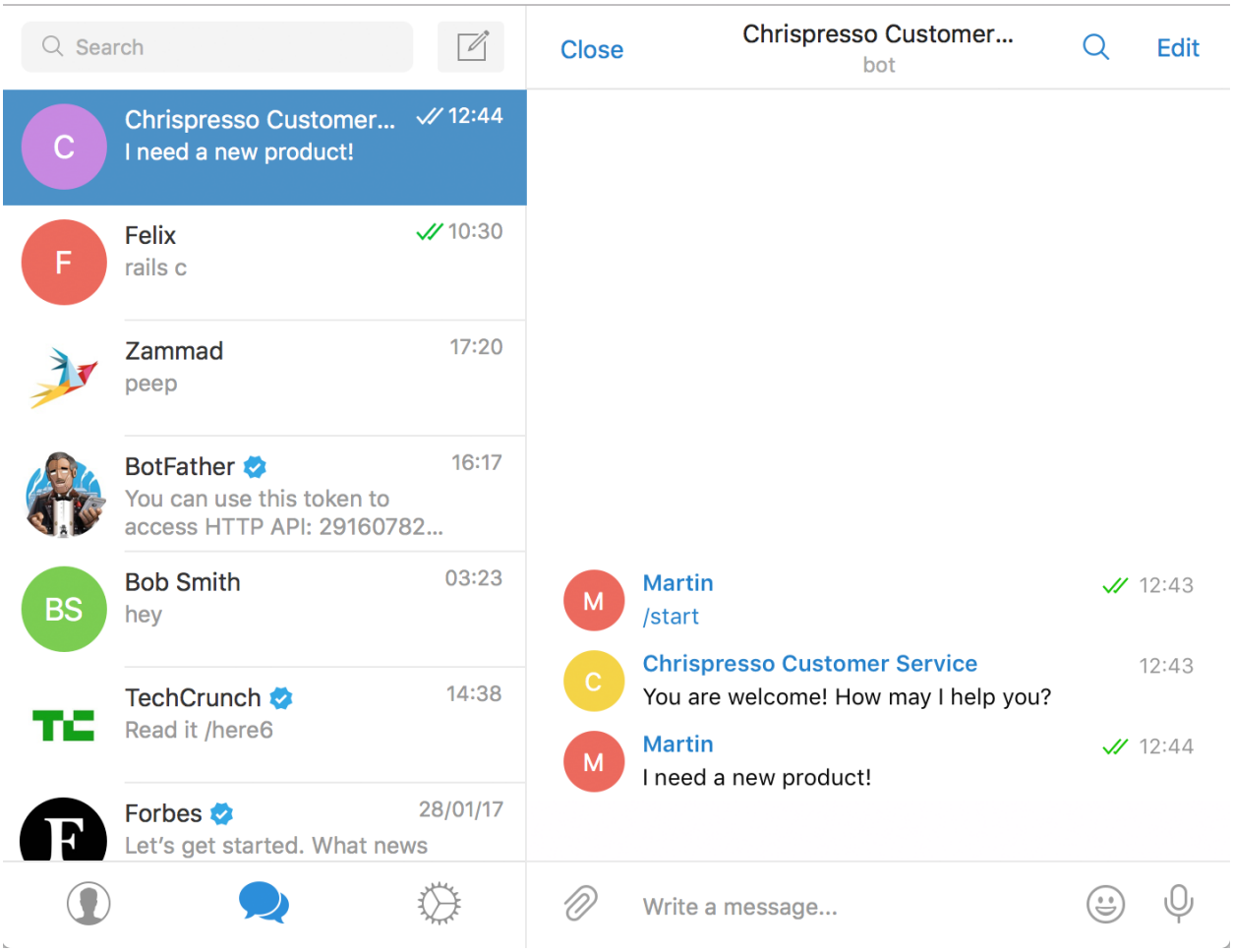
Done, your Zammad is now configured as a Telegram bot.

27.3 Start using your new channel

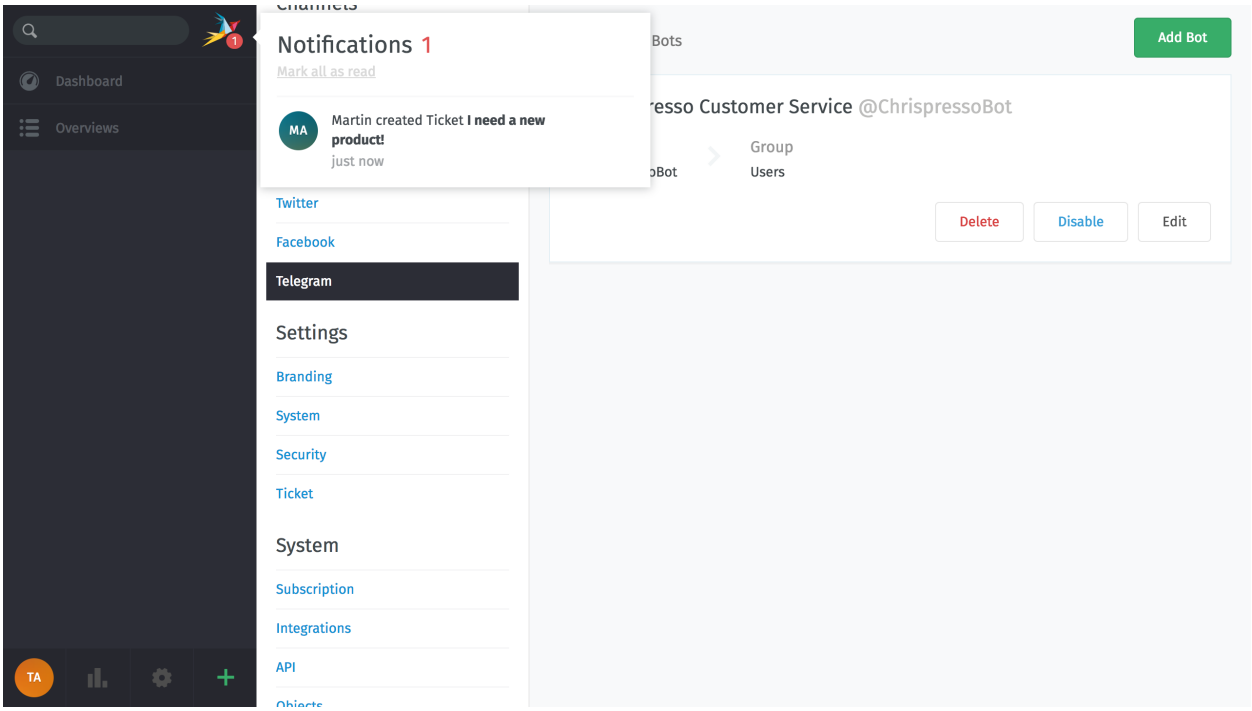
Go to your Telegram client, search for your new Telegram bot and start writing a message.


chrispresso
C
Chrispresso Customer Service
@ChrispressoBot
Select a chat to start messaging

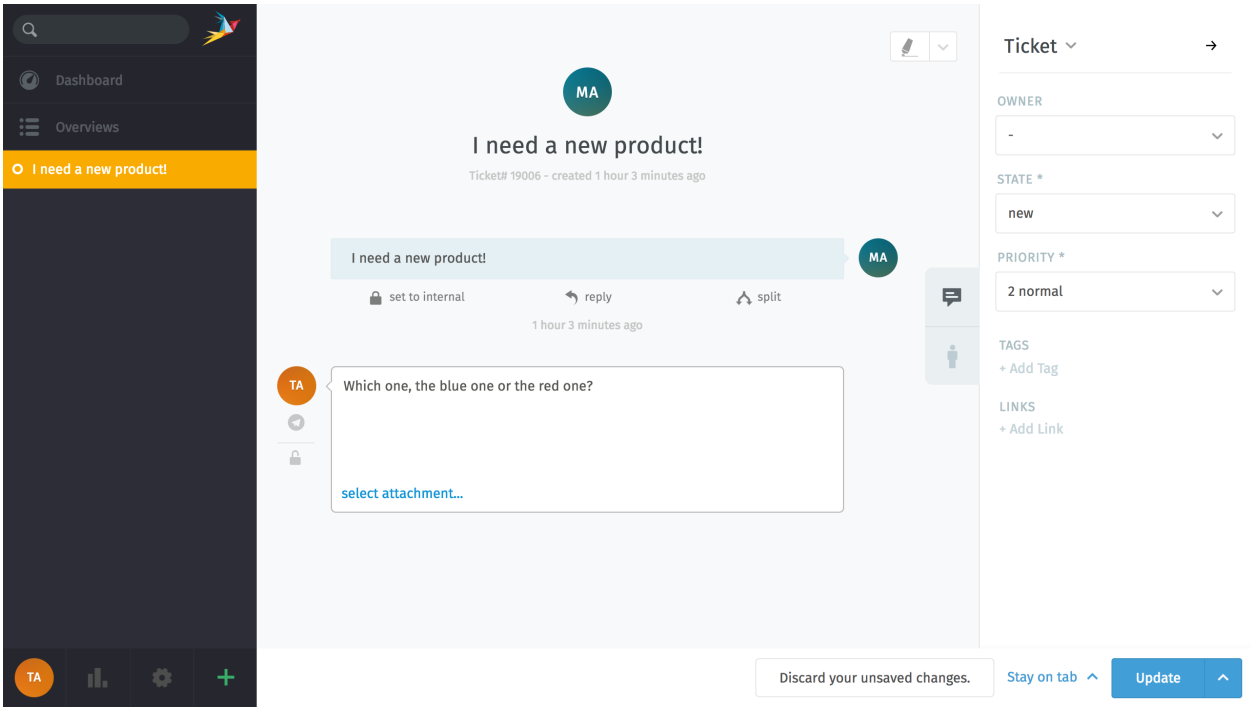

Search
Close
Chrispresso Customer...
bot
Edit
Chrispresso Customer Ser... 12:43
You are welcome! How may I help you?
Felix 10:30
rails c
Zammad 17:20
peep
BotFather 16:17
You can use this token to access HTTP API: 29160782...
Bob Smith 03:23
hey
TechCrunch 14:38
Read it /here6
Forbes 28/01/17
Let's get started. What news
Martin 12:43
/start
Chrispresso Customer Service 12:43
You are welcome! How may I help you?
Write a message...



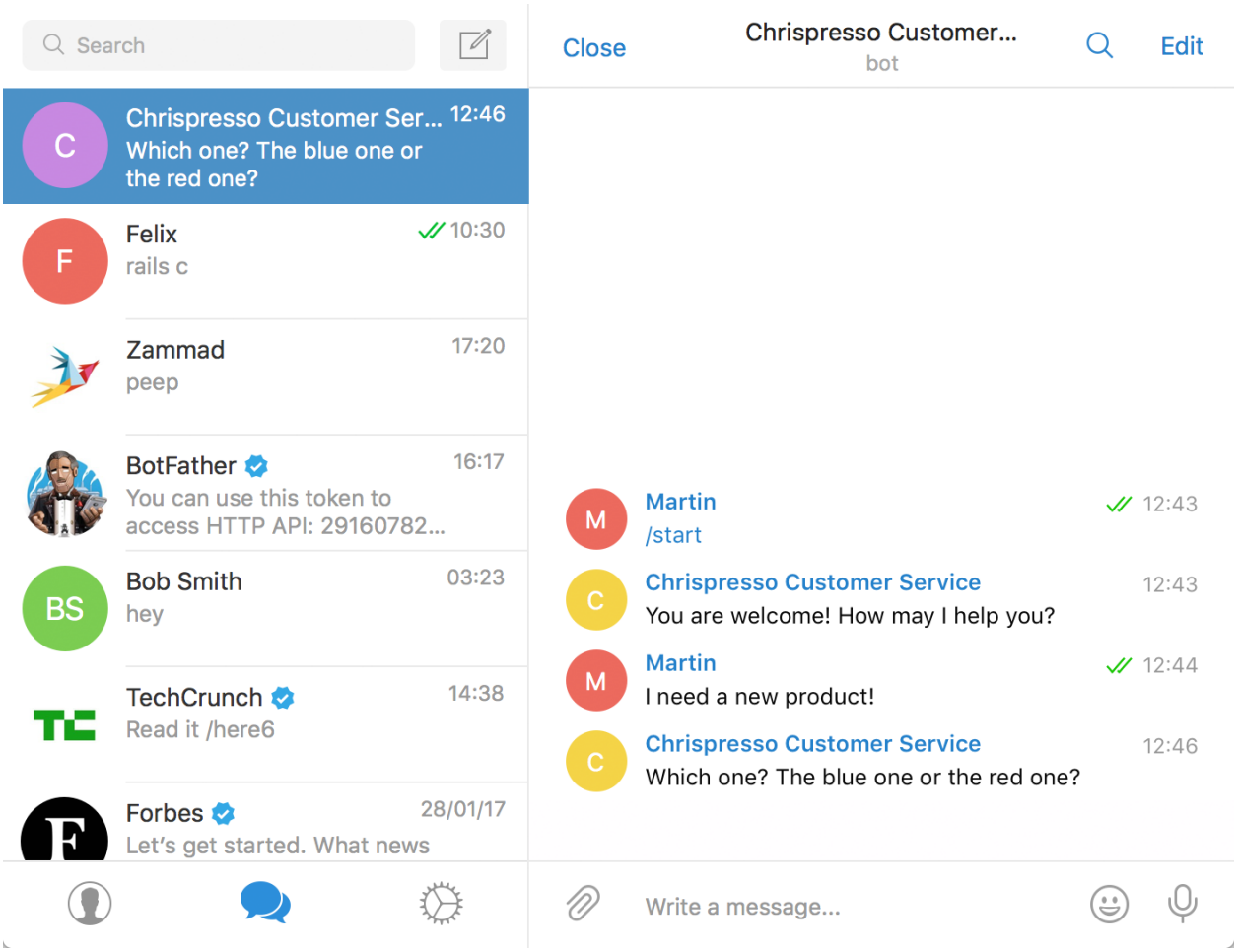
After a few seconds a new message in Zammad appears.



Just click on reply button (as you do it for emails) to send a reply.



The message will appear in your Telegram client.


Search
Close
Chrispresso Customer...
bot
Edit
Chrispresso Customer Ser... 12:46
Which one? The blue one or the red one?
Felix 10:30
rails c
Zammad 17:20
peep
BotFather 16:17
You can use this token to access HTTP API: 29160782...
Bob Smith 03:23
hey
TechCrunch 14:38
Read it /here6
Forbes 28/01/17
Let's get started. What news
Martin 12:43
/start
Chrispresso Customer Service 12:43
You are welcome! How may I help you?
Martin 12:44
I need a new product!
Chrispresso Customer Service 12:46
Which one? The blue one or the red one?
Write a message...

CHAPTER

TWENTYEIGHT

BRANDING

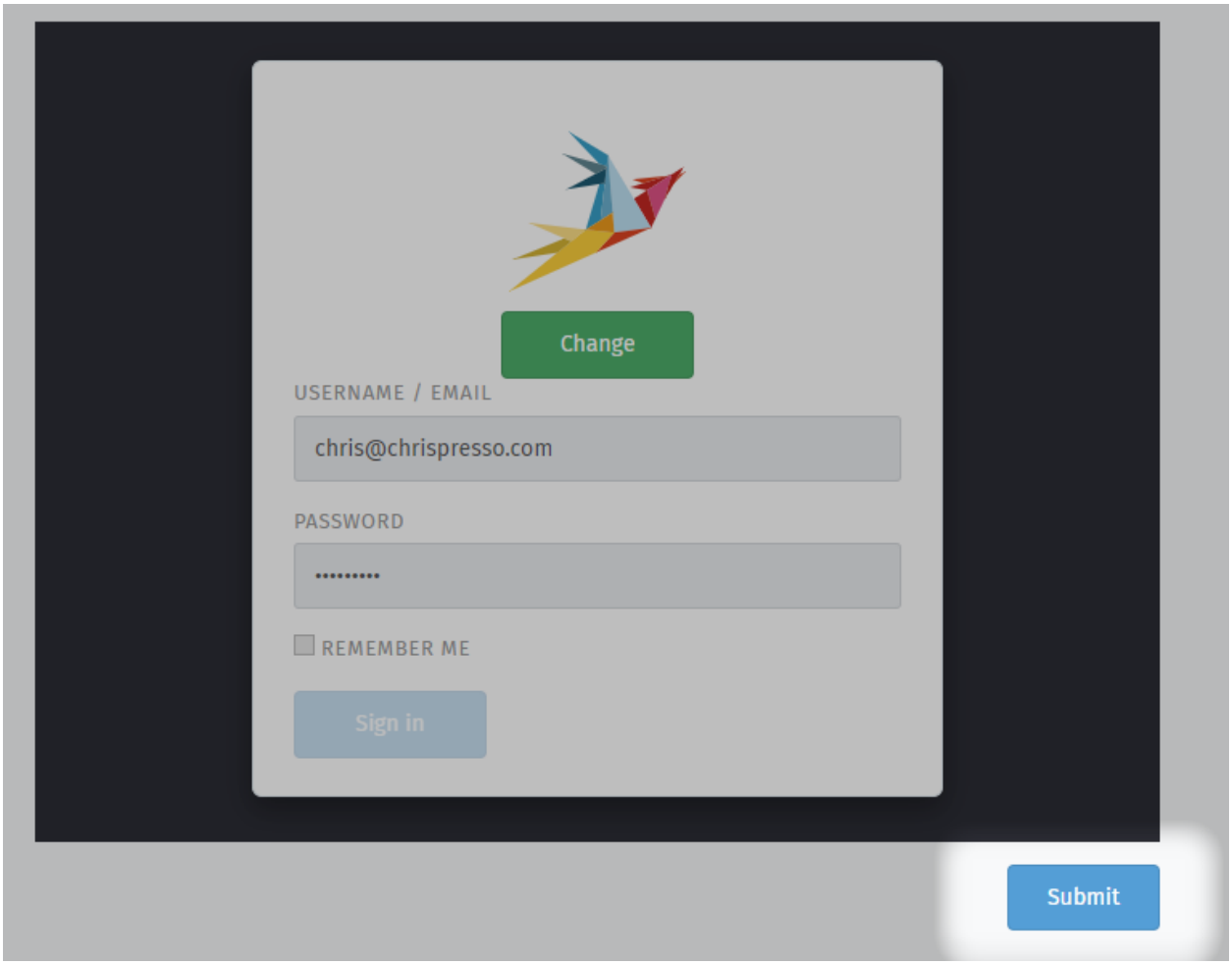
Product Name Defines the name of the application, shown in the web interface, tabs and title bar of the web browser.

Default value: `Zammad Helpdesk`

Organization Will be shown in the app and is included in email footers.

Logo Defines the logo of the application, shown in the login page of Zammad.

Note: Ensure to hit the "Submit" button after uploading the logo. Other wise your change will not be saved.



Locale Allows to set the default language of the Zammad instance. The here defined locale mostly acts as a fallback for:

- user preferences (if Zammad can't detect the users locale)
- CSV output (reporting, time accounting)
- notifications

Timezone Define the timezone of your Zammad installation.

Note: This does not have any effect on timings for your agents or how Zammad stores date and time values.

Warning: Changing this value has direct consequences on the following areas:

- *Scheduler* tasks
- search indexing (and thus reporting)
- notifications
- calendar subscriptions
- browser printing

Please note that some of above are fallbacks in case Zammad could not detect the agents timezone correctly.

Pretty Date This setting allows you to define how Zammad should display time stamps within the interface to all users.

Note: This does not have any effect on how Zammad returns e.g. time values via variables.

Choose in between the following options:

relative This timestamp is the one that changes the most. Over the time it will transition like so:

- just now
- 5 minutes ago
- 3 days 1 hour ago
- `03/04/2022`

Tip: Hovering the timestamp helps, you'll always get a clean timestamp if you do.

absolute This timestamp makes Zammad to always include the week day. For one week after creation the timestamp will not contain the date itself: Thursday 18:35.

After a week it will switch to this: Friday 4. Mar 16:00.

timestamp This will cause Zammad to show a complete timestamp according to your locale defaults. For English this would mean: e.g. `2022/12/03 2:40 pm` or for German e.g. `12.03.2022 14:40`.

Tip: This is the most consequent timestamp in Zammad as it does not change it's form and looks the same even over long time periods.

Default setting: `relative`.

CHAPTER

TWENTYNINE

SYSTEM

For your overview we split each tab within system settings into its own page:

29.1 Base

Note: Self Hosted only

Below settings are only available to self hosted users. In hosted environments we're handling these settings for you to ensure service stability.

Fully Qualified Domain Name The URL of your Zammad installation.

This setting is used within *Variables* and notifications.

Note: This setting is automatically set by the Getting Started wizard.

Warning:

- Changing this setting also changes callback URLs for your channels etc.
- This setting can have negative affects on being able to login.

HTTP type The HTTP type tells your installation how your clients connect. This is relevant for authentication and cookie security.

This setting is used within *Variables* and notifications.

Note: This setting is automatically set by the Getting Started wizard.

Warning:

- Changing this setting also changes callback URLs for your channels etc.
- This setting can have negative affects on being able to login.

SystemID This ID is being used within your ticket number. In case you're communicating with another ticket system with similar ticket number schemes this ID greatly reduces the risk of false follow ups.

The SystemID is randomly selected upon installation of Zammad (`1-99`).

> **Warning:** **Do not** change this setting in a productive system! Your Zammad installation may no longer recognize old ticket number based follow ups upon change!

29.2 Services

Image Service Defines the backend for user and organization image lookups.

Default: `Zammad Image Service` (active)

> **Hint:** On premise installations behind restrictive firewalls require `HTTPS` access to `images.zammad.com`.

Geo Calendar Service Defines the backend for geo calendar lookups. Used for initial calendar succession.

Default: `Zammad GeoCalendar Service` (active)

> **Hint:** On premise installations behind restrictive firewalls require `HTTPS` access to `geo.zammad.com`.

Geo IP Service Defines the backend for geo IP lookups. Shows also location of an IP address if an IP address is shown.

Default: `Zammad GeoIP Service` (active)

> **Hint:** On premise installations behind restrictive firewalls require `HTTPS` access to `geo.zammad.com`.

Geo Location Service Defines the backend for geo location lookups to store geo locations for addresses.

Default: `Geo Location Service` (active)

Hint: You can find a detailed privacy information on what we store for how long on our Privacy Appendix inside of our System-Administrator-Documentation.

29.3 Storage

Note: Self Hosted only

Below settings are only available to self hosted users. In hosted environments we're handling these settings for you to ensure service stability.

Storage Mechanism This tells Zammad where to store attachments for tickets and knowledge base.

By default we're writing to the `Database` - you can switch to `Filesystem` at any time. If you chose filesystem, your files are written to `/opt/zammad/storage/`

Note: We strongly encourage you to use filesystem storage on busy instances. This will greatly improve system performance (de-crease database load and size).

Tip: Moving attachments from Database to Filesystem can be run during production use.

29.4 Network

Note: Self Hosted only

Below settings are only available to self hosted users. In hosted environments we're handling these settings for you to ensure service stability.

Proxy Settings

Note: The proxy settings can only be saved after successfully testing the proxy connection.

Proxy Address. Allows you to provide a proxy server if needed. This is relevant for network communication by Zammad.

It **does not** affect the update process or Elasticsearch.

Username for proxy connection. If your proxy server requires authentication, provide the username here.

Password for proxy connection. If your proxy server requires authentication, provide the password here.

No proxy for the following hosts. Exception list for hosts you can't or don't want to reach via proxy server.

Default: `localhost,127.0.0.0,::1`

29.5 Frontend

Core Workflow Ajax Mode This setting allows administrators to enforce *Core Workflows* to use Ajax-Calls instead of web sockets. You'll usually only need this if you experience serious issues as noted below.

Hint: Possible (technical) reasons

In some cases, your network structure (e.g. firewalls, proxies) may disconnect long web socket connections. This leads to select fields staying empty (e.g. owner selection *after* selecting your group) or fields not shown / hidden (e.g. when switching to or from pending states, the "pending till" field not showing / hiding).

Please keep in mind that the Ajax fallback may cause serious pressure on your application server. If you have the choice stick to web sockets.

Default: `no` (inactive)

Show calendar weeks in the picker of date/datetime fields With this setting you can instruct Zammad to provide week number display globally for all users. Calendar with week number display are usually used in business sectors and may not be relevant for everyone.

This setting affects *all areas* - you'll also see week numbers in the admin panel for e.g. triggers and macros.

Default: `no` (not shown)

Here's the difference:

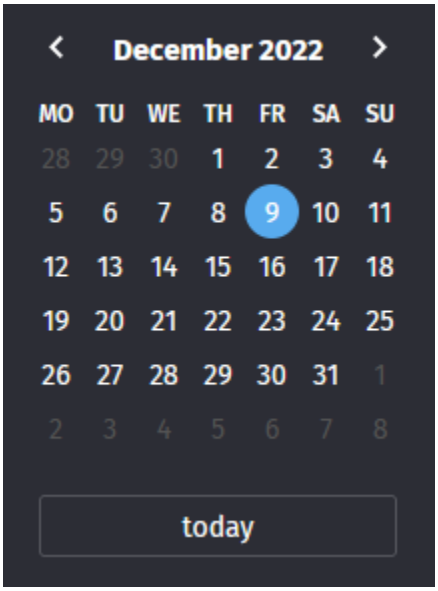


Fig. 1: Set to `no`

X

29.6 Console based settings

There's some console based settings we didn't put into the UI. However, you may find them helpful - you can find them in our hidden settings section.

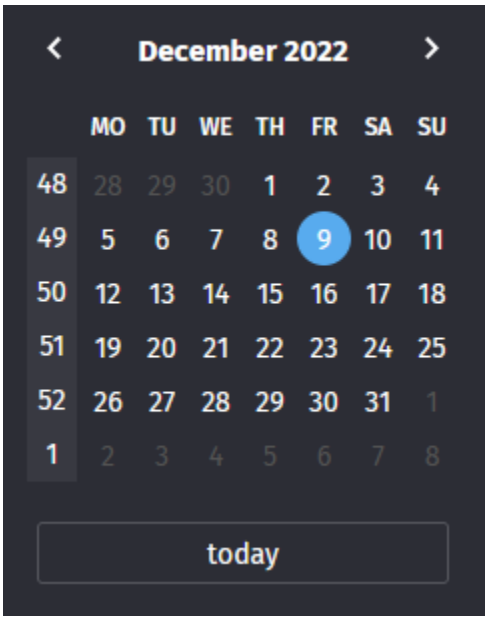


Fig. 2: Set to yes

CHAPTER

THIRTY

SECURITY

For your overview we split each tab within security settings into its own page:

30.1 Base

30.1.1 New User Accounts

Activates the *register as a new customer* function on the login page. If set to `no` only administrators or agents can create new accounts manually.

Default setting: `yes`

> **Warning: This setting may be confusing**
>
> Deactivation of above function **does not** deactivate automatic account creation! This means: If a user writes e.g. an email to Zammad and has no account yet, Zammad will automatically create the account.
>
> User accounts are a direct dependency of tickets and thus technically mandatory.

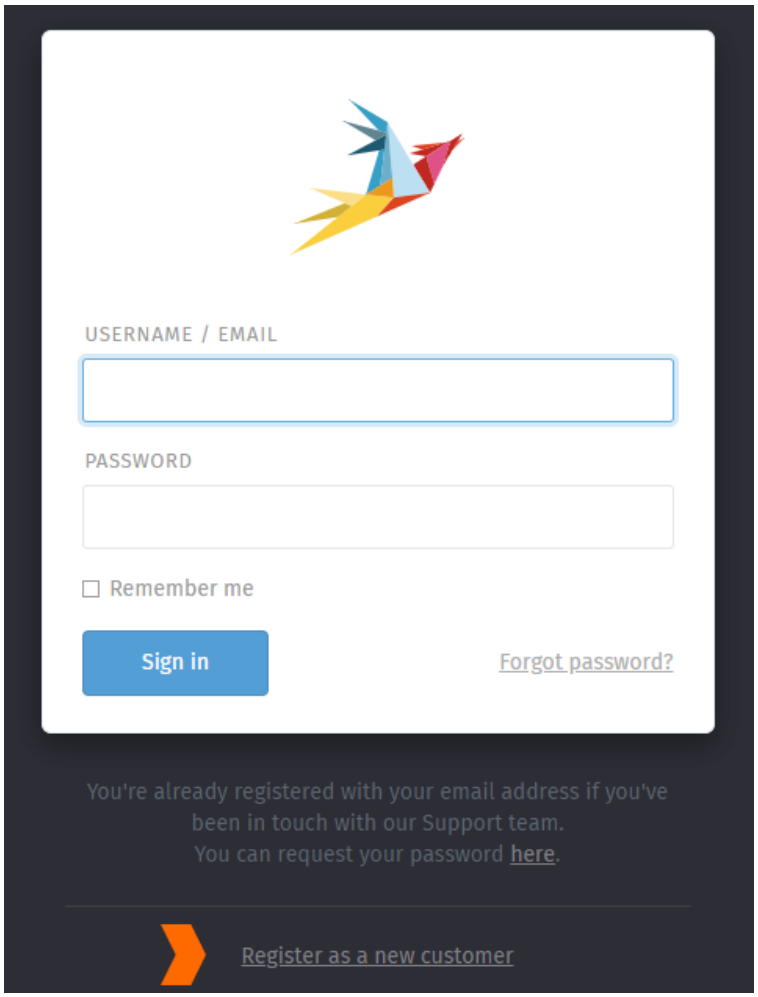
30.1.2 Lost Password

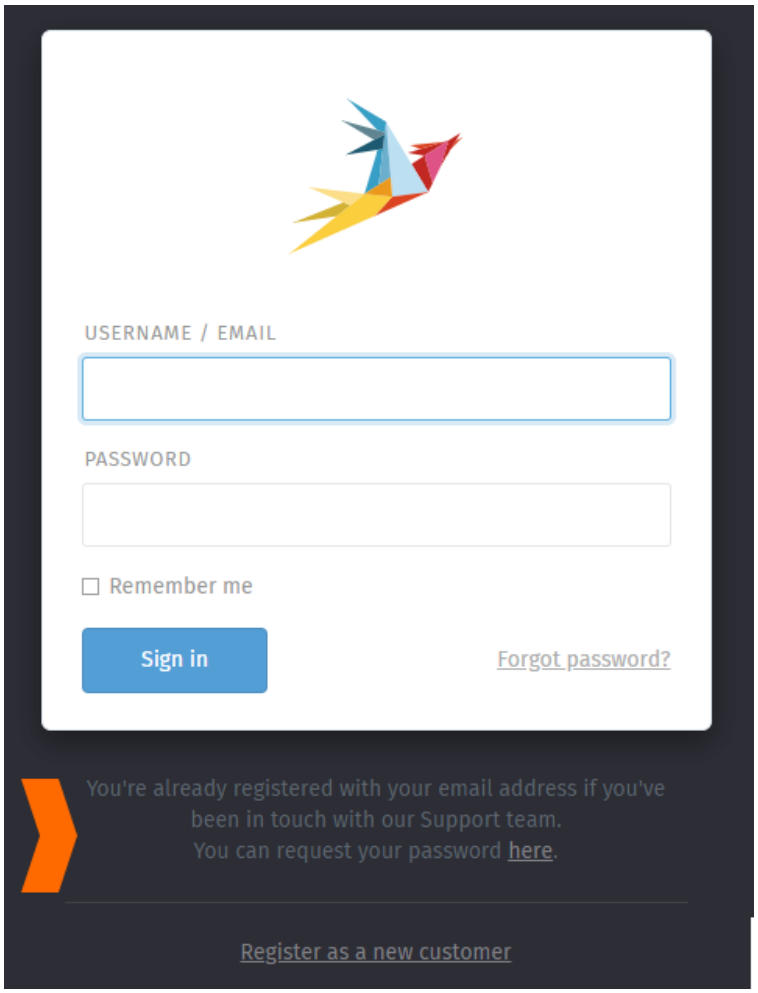
Activates the *lost password* function on the login page. If set to `no` only administrators may change the user's password - users may update their own password if they're still logged in and they have the required *permission*.

Default setting: `yes`

Tip: This function may be confusing

With third party authentications – but especially LDAP – you may want to disable this function. Zammad **will not** change third party login passwords and instead set or change the local password!


USERNAME / EMAIL
PASSWORD
Remember me
Sign in
Forgot password?
You're already registered with your email address if you've been in touch with our Support team.
You can request your password here.
Register as a new customer

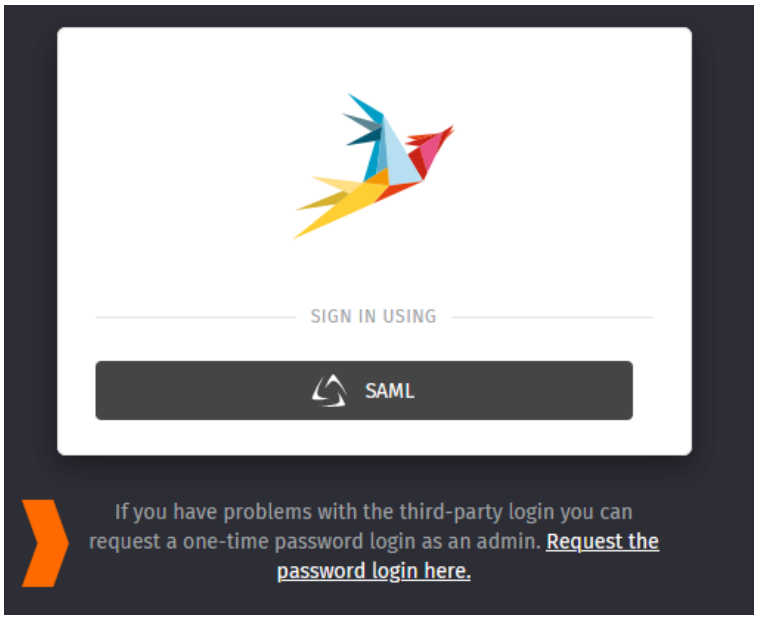

USERNAME / EMAIL
PASSWORD
Remember me
Sign in
Forgot password?
You're already registered with your email address if you've been in touch with our Support team.
You can request your password here.
Register as a new customer

30.1.3 Password Login

Activates the username & password login by default and if no third-party login is activated.

See *Third-Party Applications* for supported third-party logins.

Default setting: `yes`



Hint: To help administrators to overcome "login issues", Zammad automatically offers a password request for administrator users. This allows you to adjust Third-Party applications even if the login does no longer work!

Note: This function may be confusing

Disabling password login on the Zammad login page only takes effect if you enable any *Third-Party Applications*.

30.1.4 Session Timeout

All settings below by default are set to `4 weeks`. Session Timeout defines the life time of a users session. As soon as it's reached, Zammad will automatically log off the session in question.

Zammad takes the highest value set assigned for the user based on the permissions.

1. `admin`
2. `ticket.agent`
3. `ticket.customer`
4. `default` (fallback if user doesn't have above permissions set)

All settings act independently from each other allowing you to disable the timeouts for e.g. admins, but not agents.

Note: An example

Let's suppose you configured the following session timeouts

- default: 3 weeks
- admin: 2 weeks
- ticket.agent: 4 weeks
- ticket.customer: 1 week

This results in the following situations

- a user with `admin` permission will have a timeout of 2 weeks
- a user with `admin` and `ticket.agent` permissions will have a timeout of 2 weeks
- a user with `ticket.customer` permission will have a timeout of 1 week
- a user with neither `admin`, `ticket.agent` nor `ticket.customer` permissions will have a timeout of 3 weeks

---

30.2 Password

This section allows you to define password requirements for the local user accounts.

Note: Zammad does **not** allow you to change your LDAP password, instead, it will set a password in its local database which might confuse your users. This will be addressed in the future by #1169.

Warning: Exception for strong passwords

Please note that below password policies *do not* affect administrators setting passwords on user accounts. While this seems strange and not safe we believe that an administrator knowing an user's password is insecure as well.

The suggested workflow is either:

- to use third party logins to not require local passwords at all **- or -**
- to require your *user to reset the password* upon first login.

This way administrators are not required to set a user's password at all!

30.2.1 Maximum failed logins

You can choose a value between `4` and `20`. This defines how often a login to a user account may fail until Zammad will lock it. Your users can always use the "forgot password" function to change their password and unlock their account.

The default value is `10`.

Note: Beside changing the user's password, you can also unlock accounts via

- *user management list*
- console
- api

Hint: Failed logins via LDAP no longer lock accounts.

30.2.2 2 lower case and 2 upper case characters

You can add complexity to passwords by enforcing at least 2 upper and lower case characters.

The default value is `no`.

30.2.3 Minimum length

This defines the minimum password length required for users to provide (from `4` to `20`).

The default value is `6`.

30.2.4 Digit required

This enforces your users to use at least one digit within the password.

The default value is `yes`.

30.2.5 Special character required

This setting allows you to improve your password quality even more by enforcing the password to contain at least one special character.

The default value is `no`.

30.3 Third-Party Applications

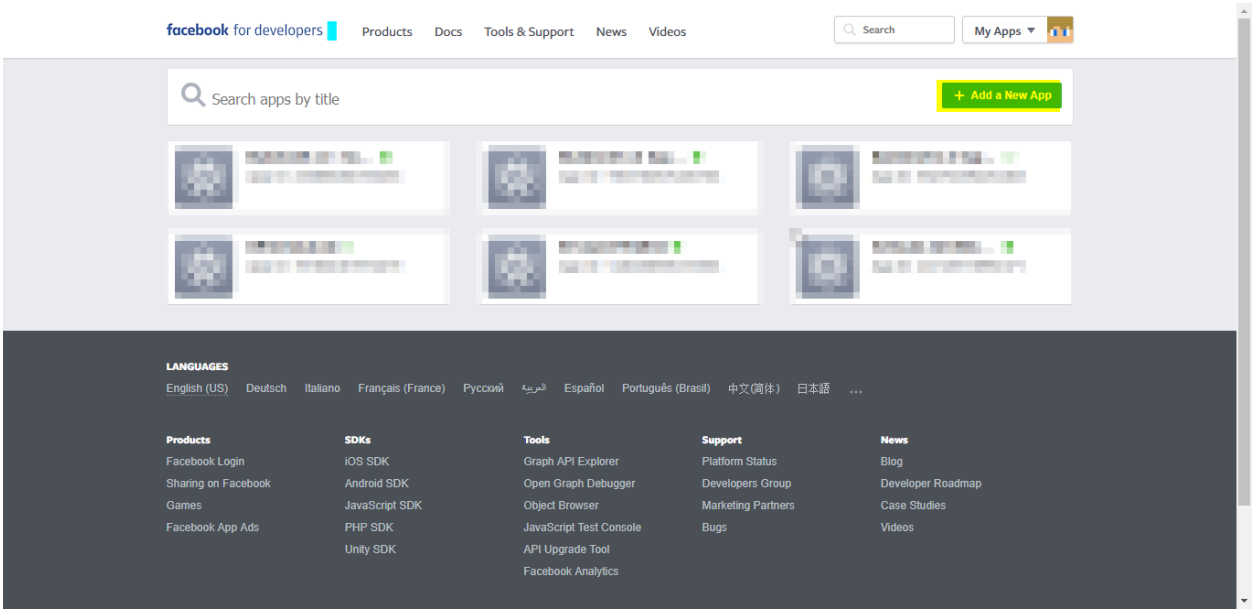
Third party authentication is a great way to help your users to login to Zammad more easily. If the account is yet unknown, Zammad will create a new user automatically, without the user needed to interact (e.g. type in his name). Another big advantage of this feature is that your user doesn't need to remember another password.

30.3.1 Facebook

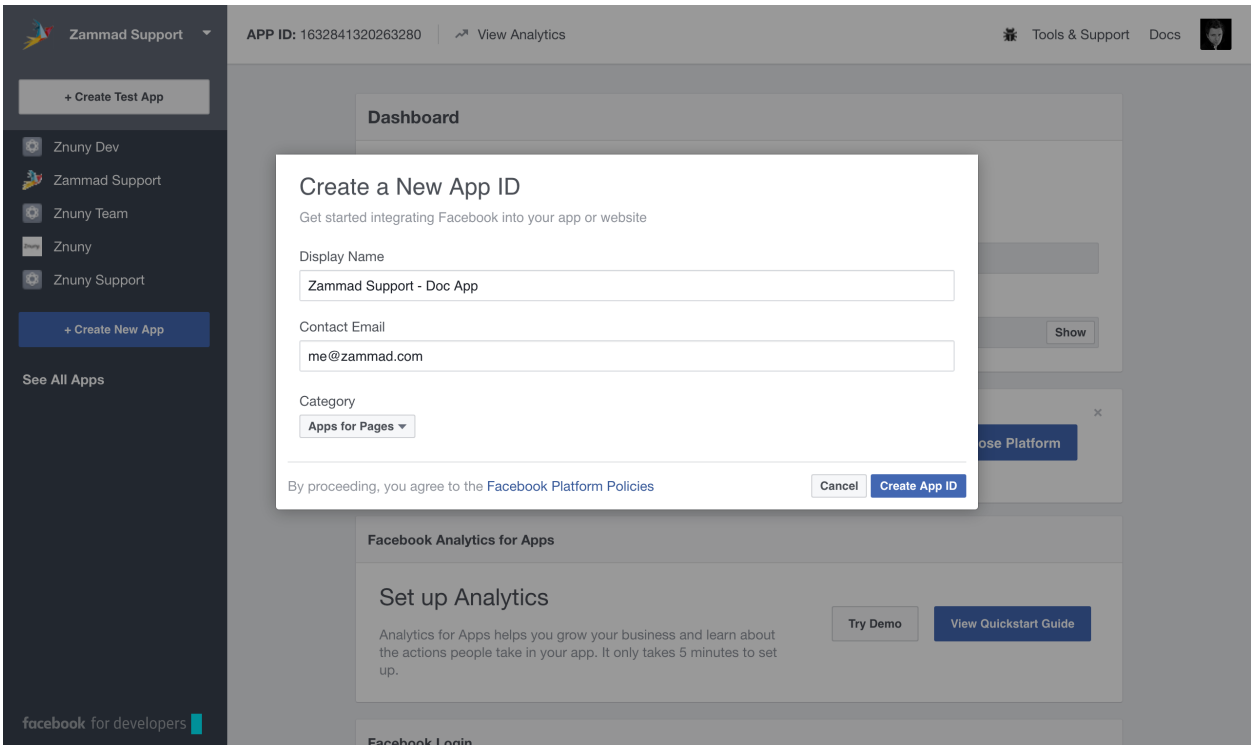
It is possible to create a quick login for your helpdesk via Facebook To do so, you need to follow these steps:

Register Facebook app

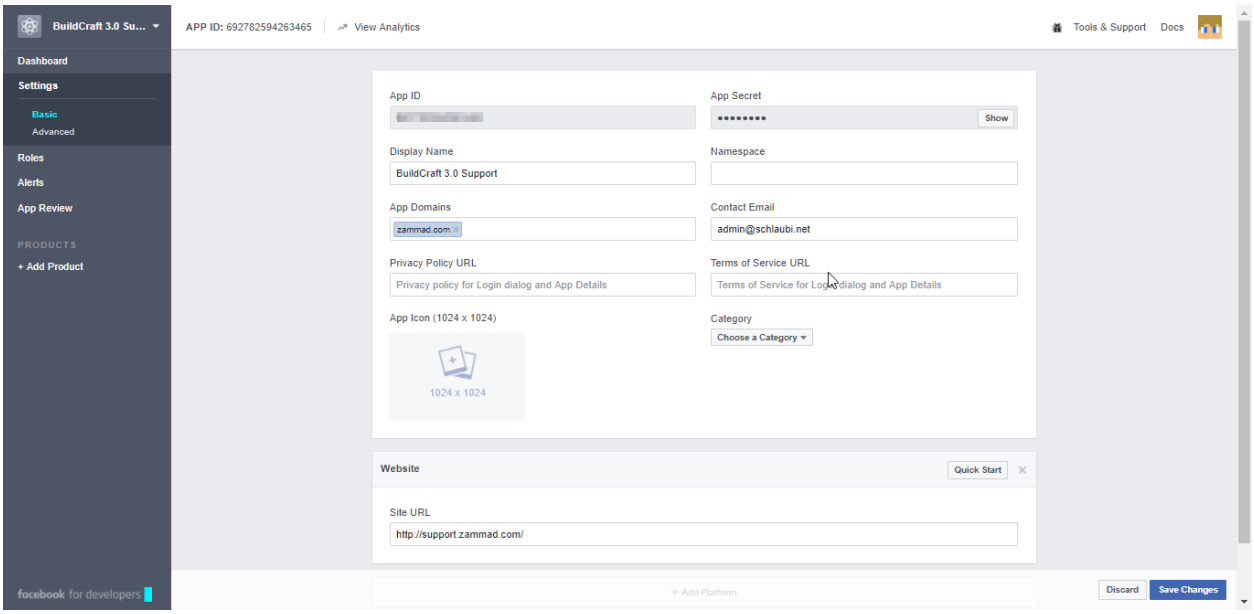
Visit [https://developers.facebook.com/apps/] and click on “Add a new App”



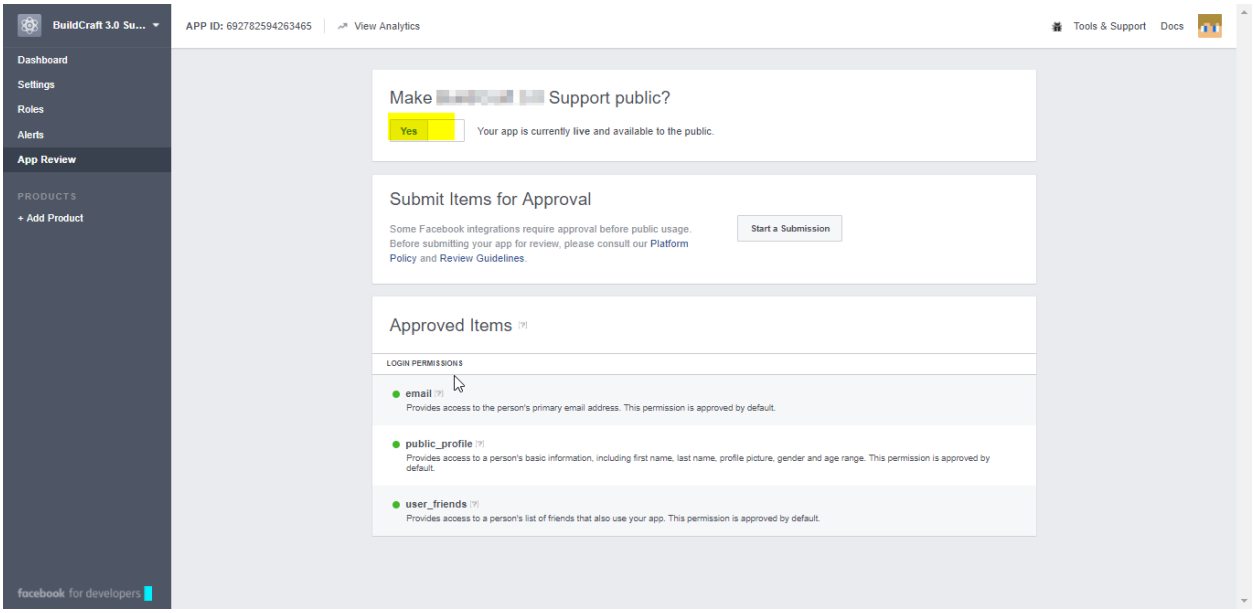
After that enter the app settings



Navigate to “Settings” and fill in this infromation

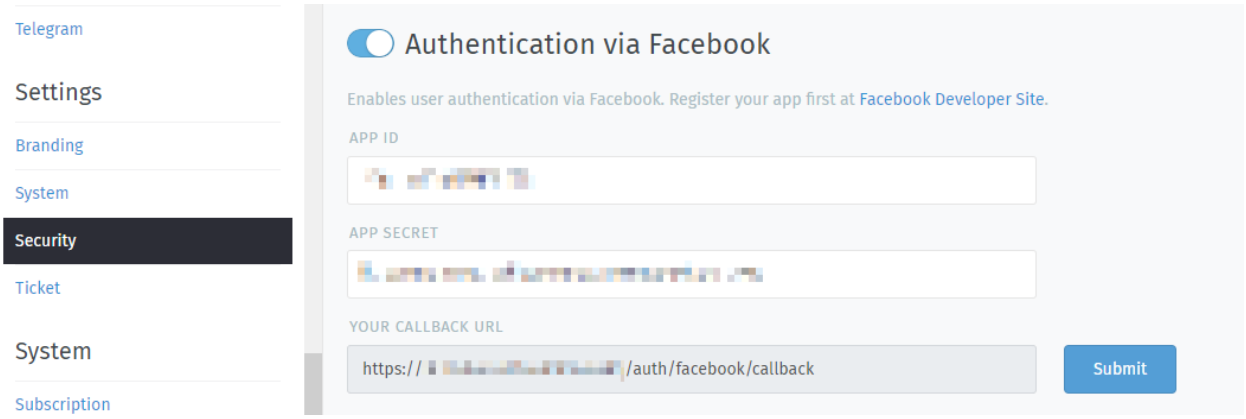


Navigate to app review and Check “Make [appname] public?”



Configure Zammad as Facebook app

Navigate to “Admin -> Security -> Third Party Applications” and enter the App ID and the App Secret. You can find this Key in the Dashbard of your Facebok app.



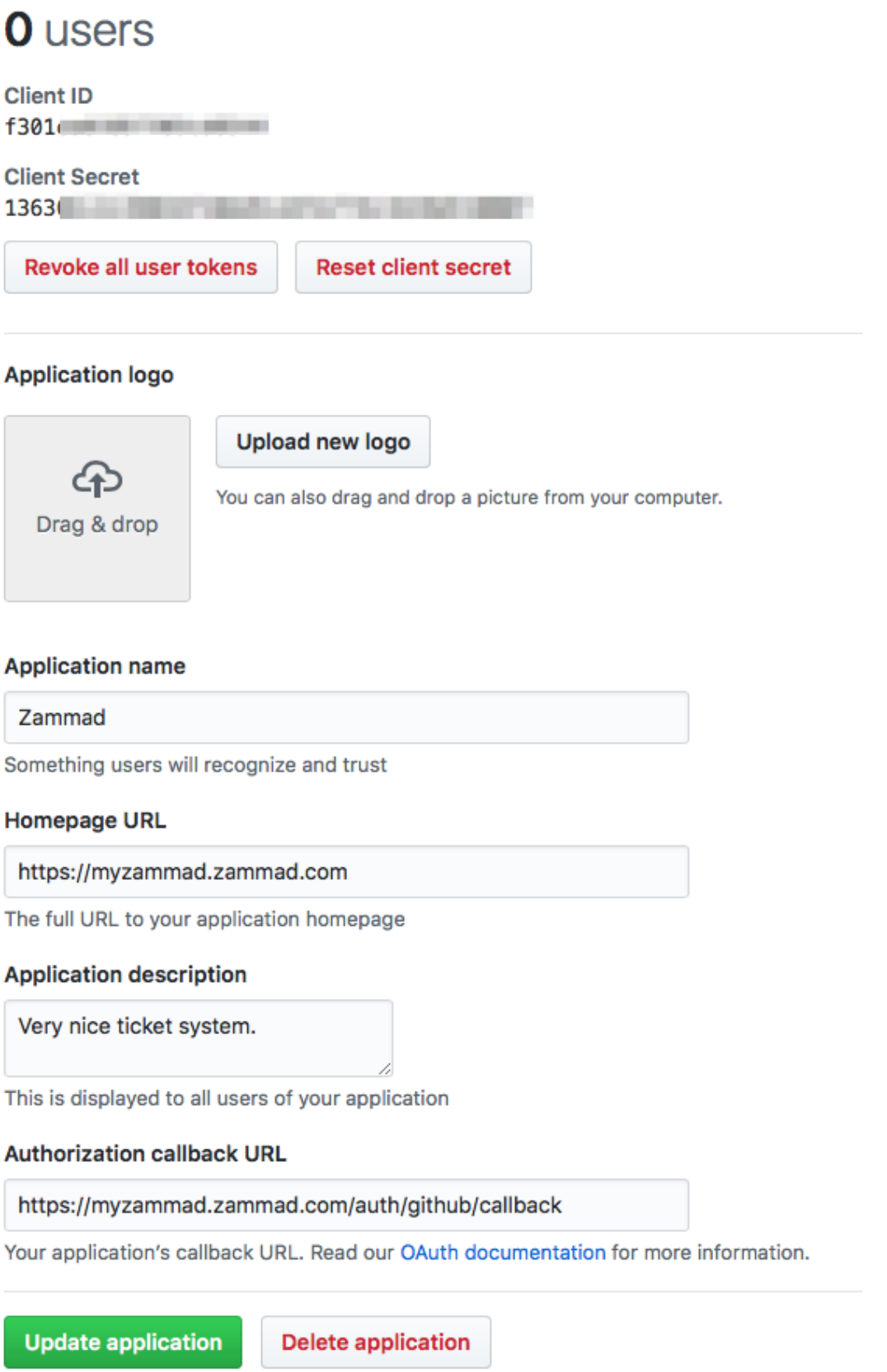
Now you can link accounts via “Avatar -> Profile -> Link Accounts” or login via Zammad login page.

30.3.2 GitHub

It is possible to create a quick login for your helpdesk via GitHub. To activate the quick login you need to enable OAuth for GitHub.

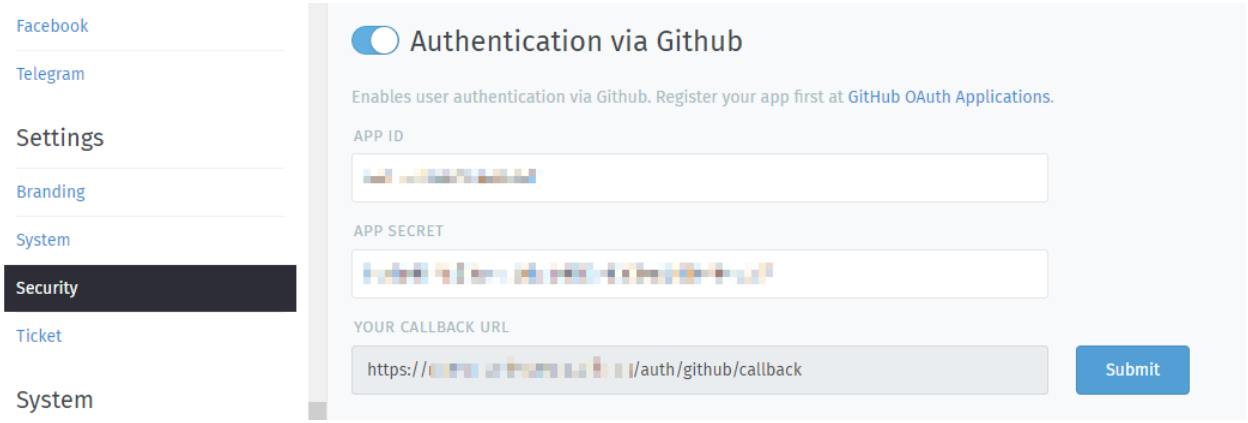
Register GitHub app

Visit https://www.github.com/settings/applications/new and enter the app settings. As callback URL enter “https://zammad_host/auth/github/callback” where zammad_host has to be replaced with your Zammad FQDN

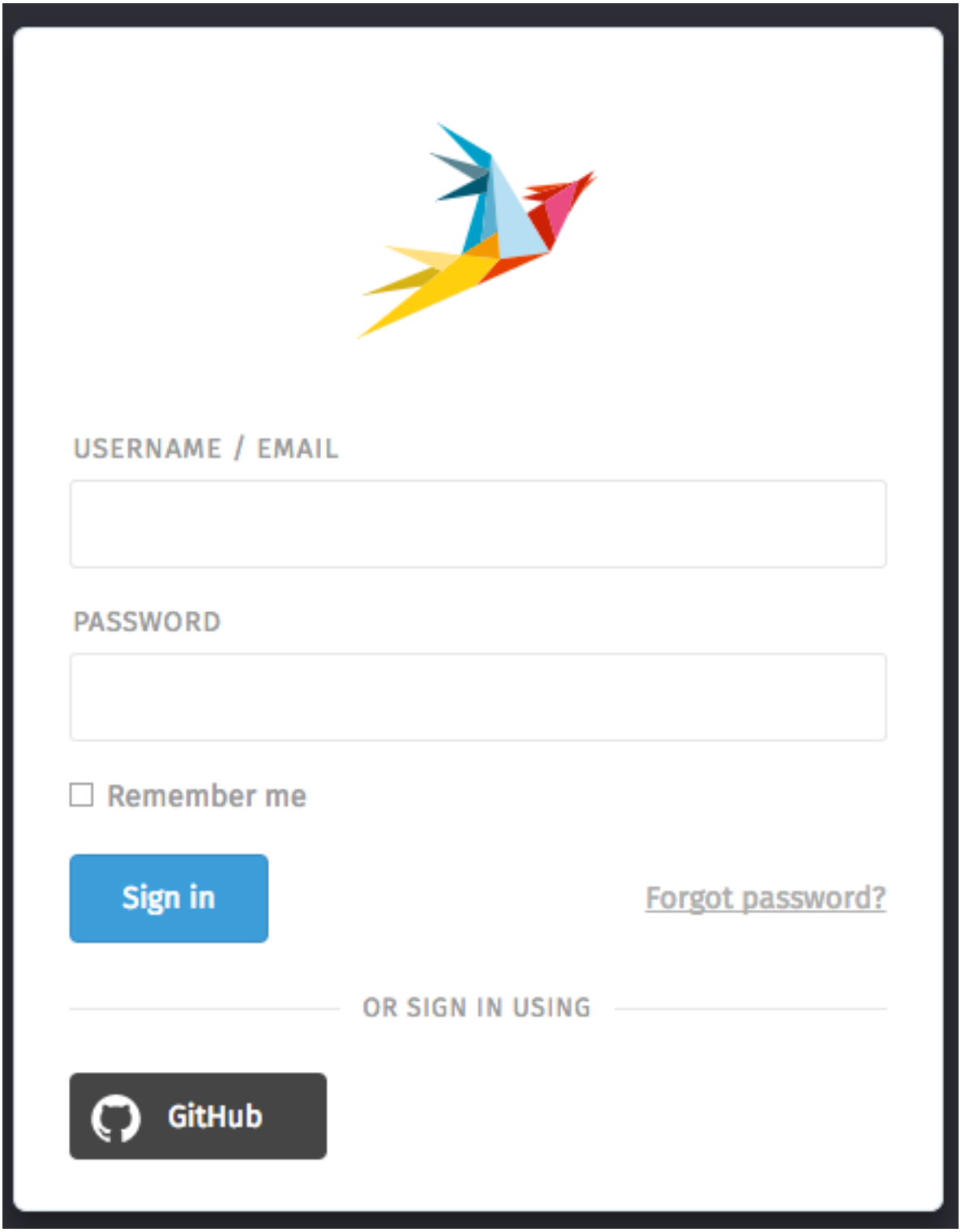

0 users
Client ID
f301
Client Secret
1363
Revoke all user tokens
Reset client secret
Application logo
Drag & drop
Upload new logo
You can also drag and drop a picture from your computer.
Application name
Zammad
Something users will recognize and trust
Homepage URL
https://myzammad.zammad.com
The full URL to your application homepage
Application description
Very nice ticket system.
This is displayed to all users of your application
Authorization callback URL
https://myzammad.zammad.com/auth/github/callback
Your application's callback URL. Read our OAuth documentation for more information.
Update application
Delete application

Configure Zammad as GitHub app

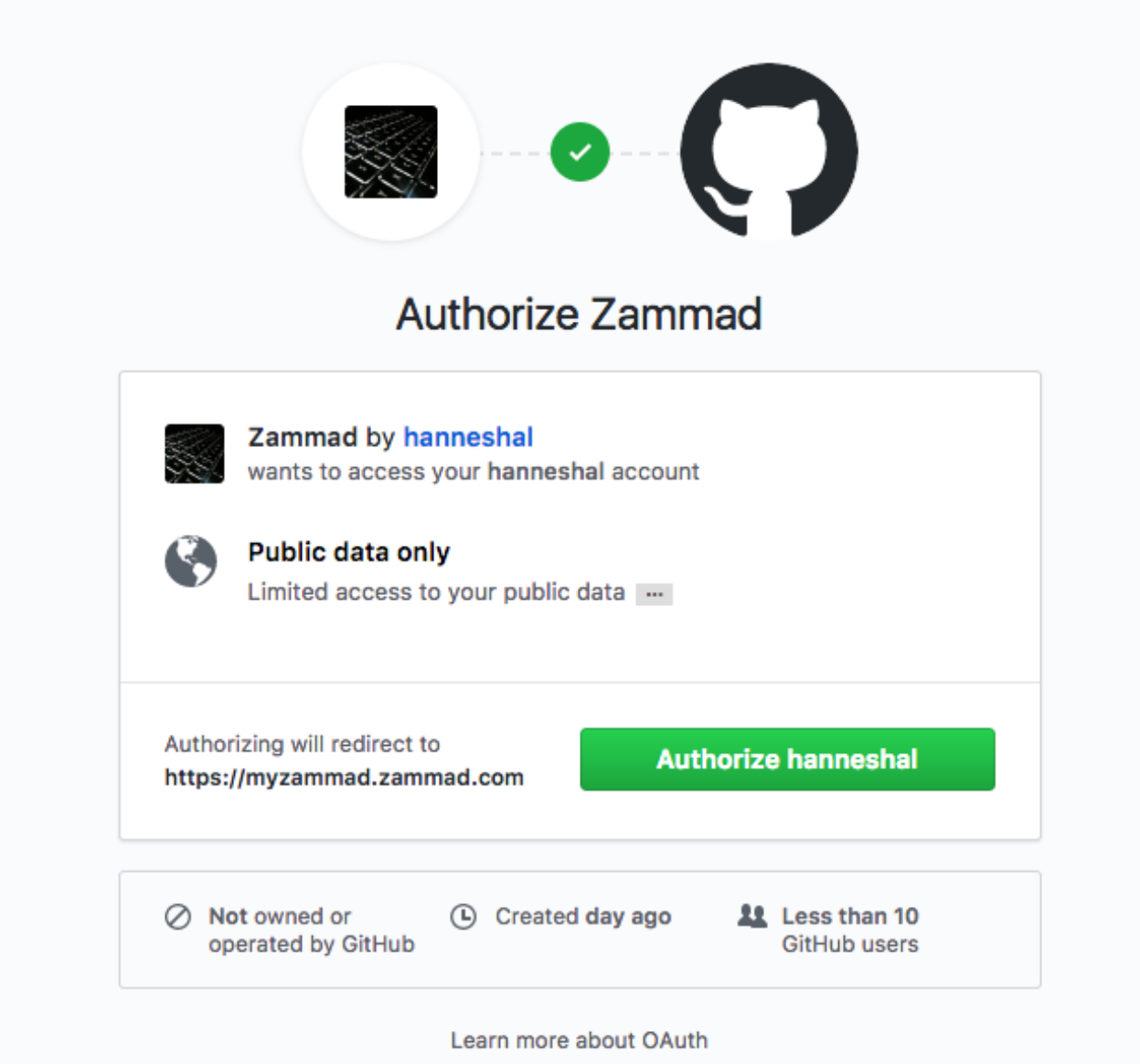
Enter the “APP ID” and the “APP SECRET” from the GitHub OAUTH Applications Dashboard



After you configured the GitHub credentials and activated the login method, you should see a new icon on the login page.



If you click on the icon you will be redirected to GitHub and see something similar to this:



When you grant the access you will be redirected to your Zammad instance and logged in as a customer.

Now you can link accounts via “Avatar -> Profile -> Link Accounts” or login via Zammad login page.

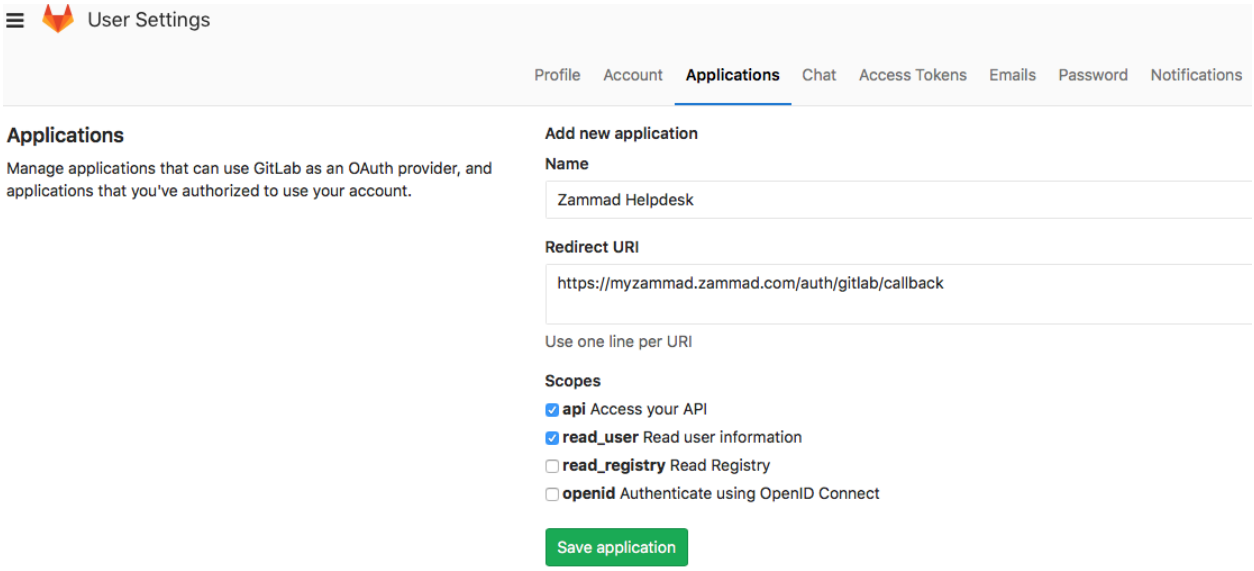
30.3.3 Gitlab

It is possible to create a quick login for your helpdesk via Gitlab. To activate the quick login you need to enable OAuth for Gitlab.

Register Gitlab app

To register an app in Gitlab open your profile and select applications.

As callback URL enter "https://zammad_host/auth/gitlab/callback" where zammad_host has to be replaced with your Zammad FQDN

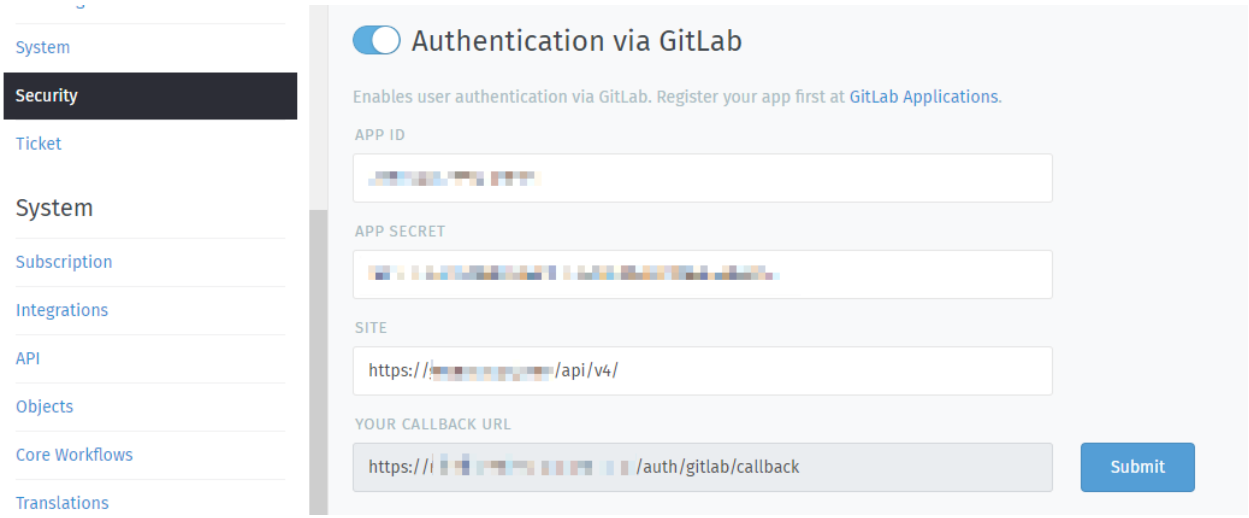


At the moment we need the "api" scope. This is caused due a bug in Gitlab: https://gitlab.com/gitlab-org/gitlab-ce/issues/33022

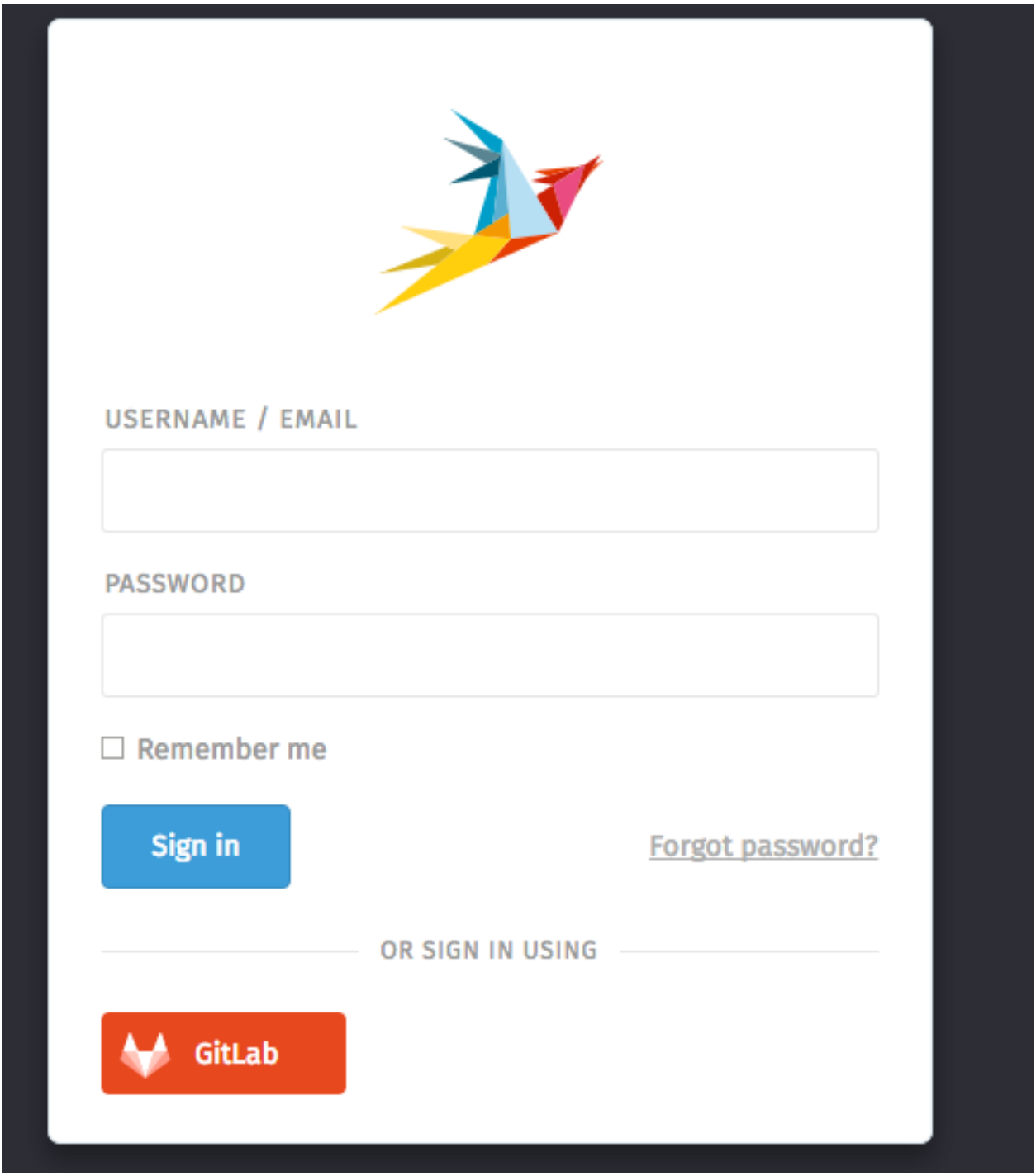
Configure Zammad as Gitlab app

Enter the "APP ID" and the "APP SECRET" from the Gitlab OAUTH Applications Dashboard.

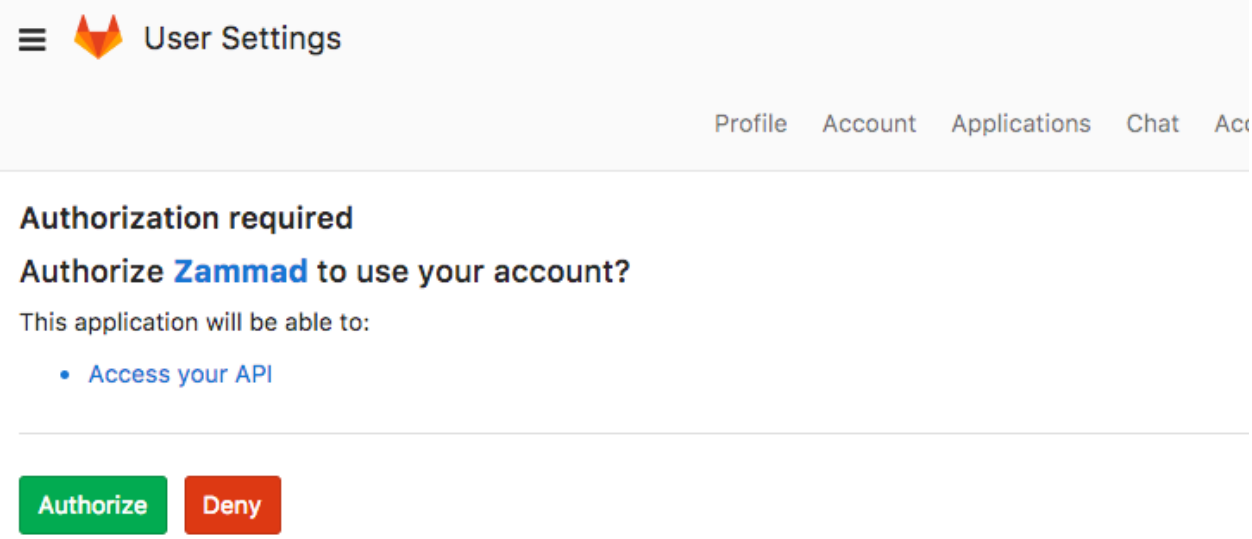
Note: Please ensure to use `https://{git_host}/api/v4/` for site.



After you configured the Gitlab credentials and activated the login method, you should see a new icon on the login page.



If you click on the icon you will be redirected to Gitlab and see something similar to this:



When you grant the access you will be redirected to your Zammad instance and logged in as a customer.

Now you can link accounts via “Avatar -> Profile -> Link Accounts” or login via Zammad login page.

30.3.4 Google

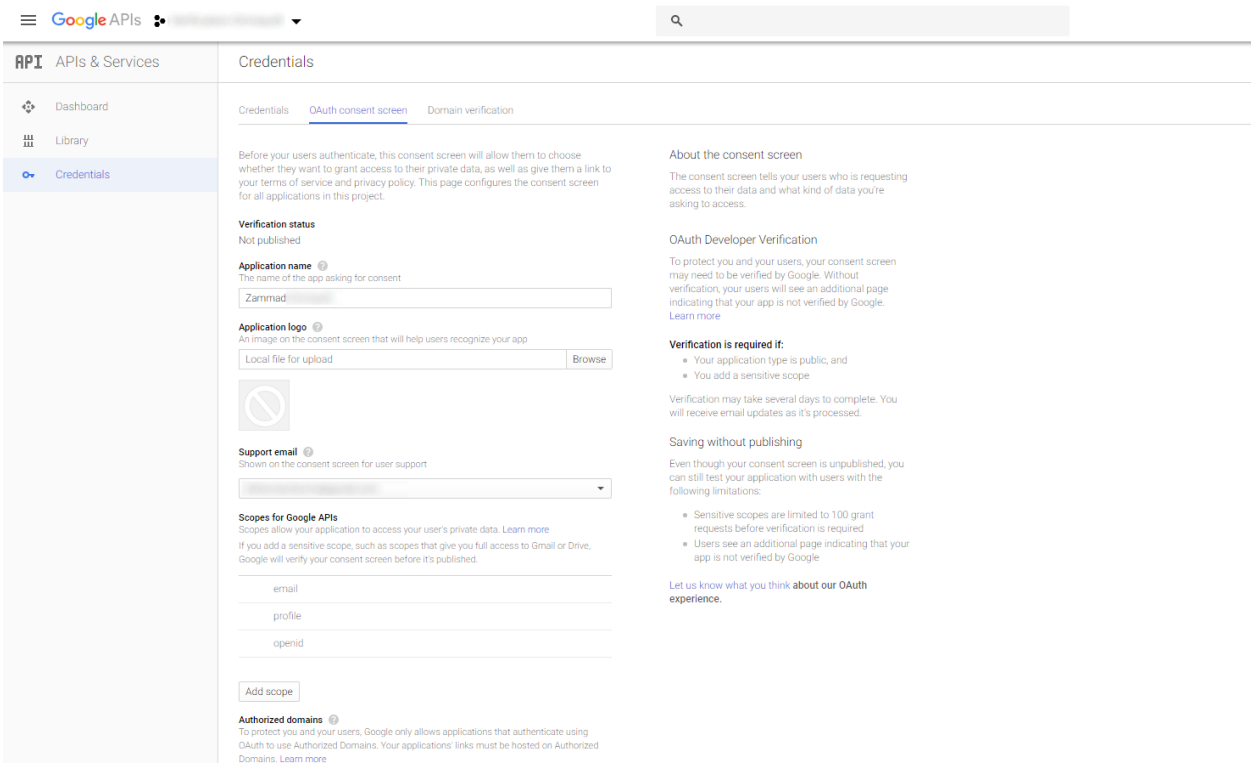
With some easy and fast steps, you can enable Zammad to authenticate your users via Google.

Register a Google app

First of all, we need to create a new project - you can skip this step if you already have one.

Hint: Use this link to create a new project: `https://console.cloud.google.com/projectcreate`

Now expand the menu, expand “APIs & Services” and select “Credentials”. Go to the tab “OAuth consent screen” first and ensure to fill in the requested information - without doing so you can’t create credentials!



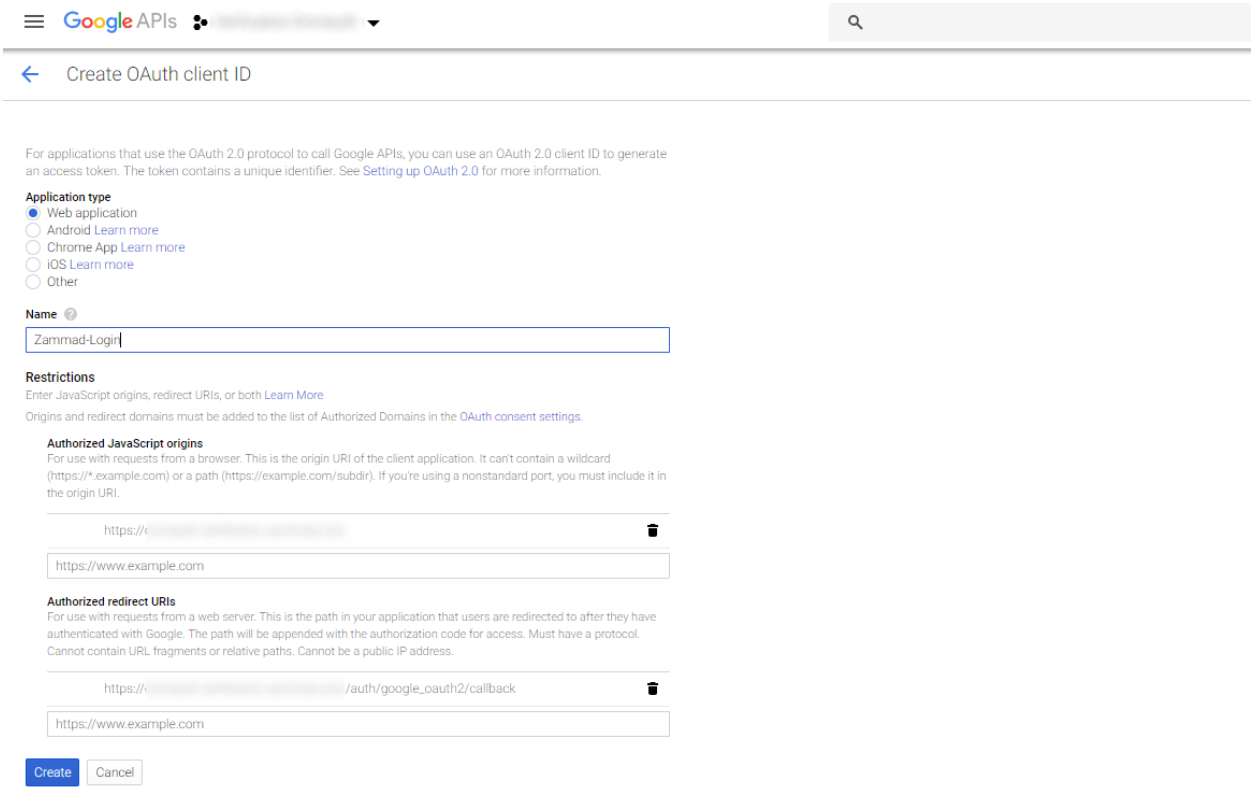
After filling in and savingthe consent screen information, you can change to “Credentials” tab and create new “OAuth client ID”-Credentials.

Fill in the neceassary information, for restrictions you need the following (replace `zammad_host` with your FQDN):

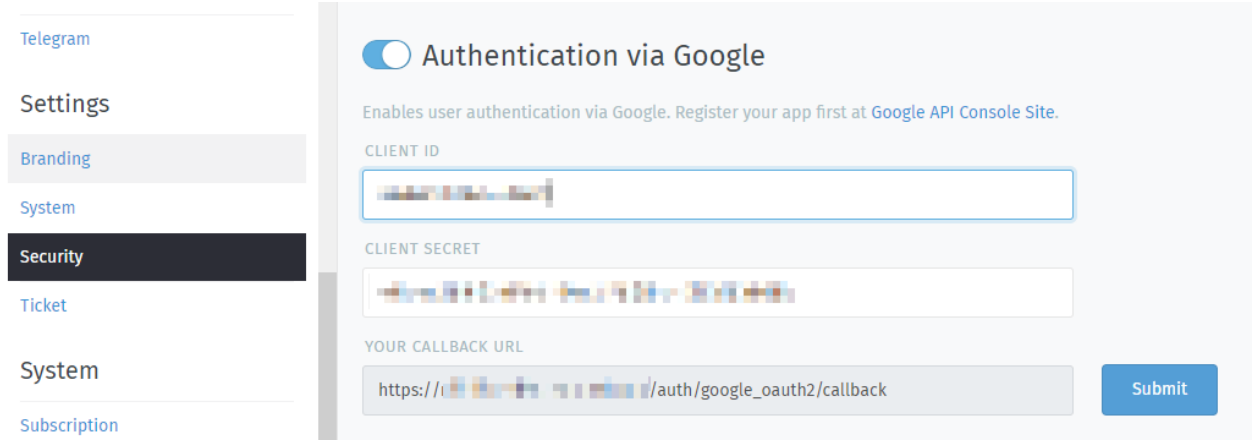
Aplication type `[x] Web application`

Authorized JavaScript origins `https://zammad_host/`

Authorized redirect URIs `https://zammad_host/auth/google_oauth2/callback`



After creating the credentials, go to your Zammad installation and navigate to “Admin -> Security -> Third Party Applications” -> Google. Enter your Client ID and Client secret here.



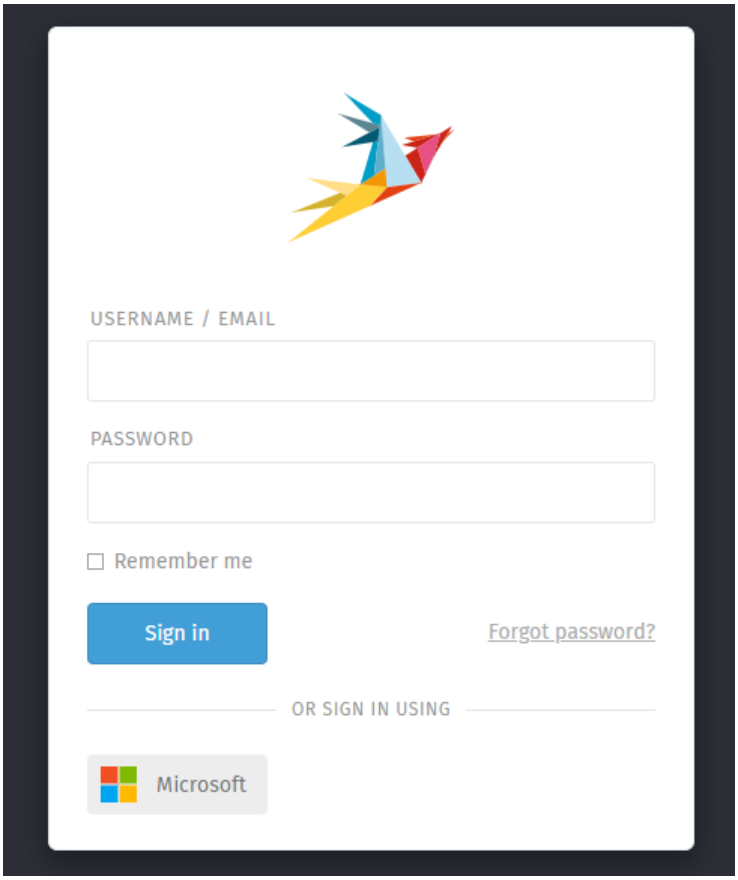
After submitting, the login via Google can be used.

30.3.5 Microsoft

Zammad's Microsoft connection allows your users with Microsoft accounts to login. This works for Azure users as well and can be an alternative to *LDAP / Active Directory*.

Hint: This login function was called "Office 365" prior Zammad 5.1.

Note: This documentation part does not cover our *Microsoft 365 email channel*.



Limitations

Supported account types: Please note that Zammad only supports these account types (App dependent):

- Accounts in this organizational directory only (Default Directory only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

Avatars of signing in users: Zammad currently ignores user avatars. If the user is registered with e.g. Gravatar, Zammad will fetch the avatar from there if enabled. See *Zammad Image Service* for more.

Step 1 - Register a Microsoft app for Zammad

Login to the Microsoft Azure Portal and navigate to App registrations to create a new app. Provide the requested information as follows and register your app.

Name: Any meaningful name fitting, this name will be displayed to users trying to authenticate with this app.

Supported account types: Choose one of the above mentioned account types (see Limitations).

Tip: The correct account type depends on your use case. If you want to use the authentication internal only, choose the first option. If you're unsure, use the "Help me choose…" link.

Redirect URI (optional): Select web and provide your callback url. The callback url looks like this: `https://zammad.domain.tld/auth/microsoft_office365/callback`

Within API permissions add the following permissions:

OpenId permissions

- `openid`

User

- `User.Read`

Contacts

- `Contacts.Read`

You can find these permissions within *Microsoft Graph → Delegated permissions*.

Within **Certificates & secrets** create a new client secret. Note down the returned secret **value** for later. **Do not** use the secret ID!

From **Overview** copy your apps *Application (client) ID*. If you're using a single tenant app, please also copy *Directory (tenant) ID*. You now have all required information for Zammad.

Step 2 - Add app credentials to Zammad

Navigate to Security → Third-party Applications (Tab) within Zammad's admin settings. Scroll down to the section *Authentication via Microsoft* and fill in the required information.

App ID: This is your *Application (client) ID*.

App secret: This is your *client secret* (value).

App Tenant ID: **optional** only required for apps that use account type *Accounts in this organizational directory only (Default Directory only - Single tenant).*

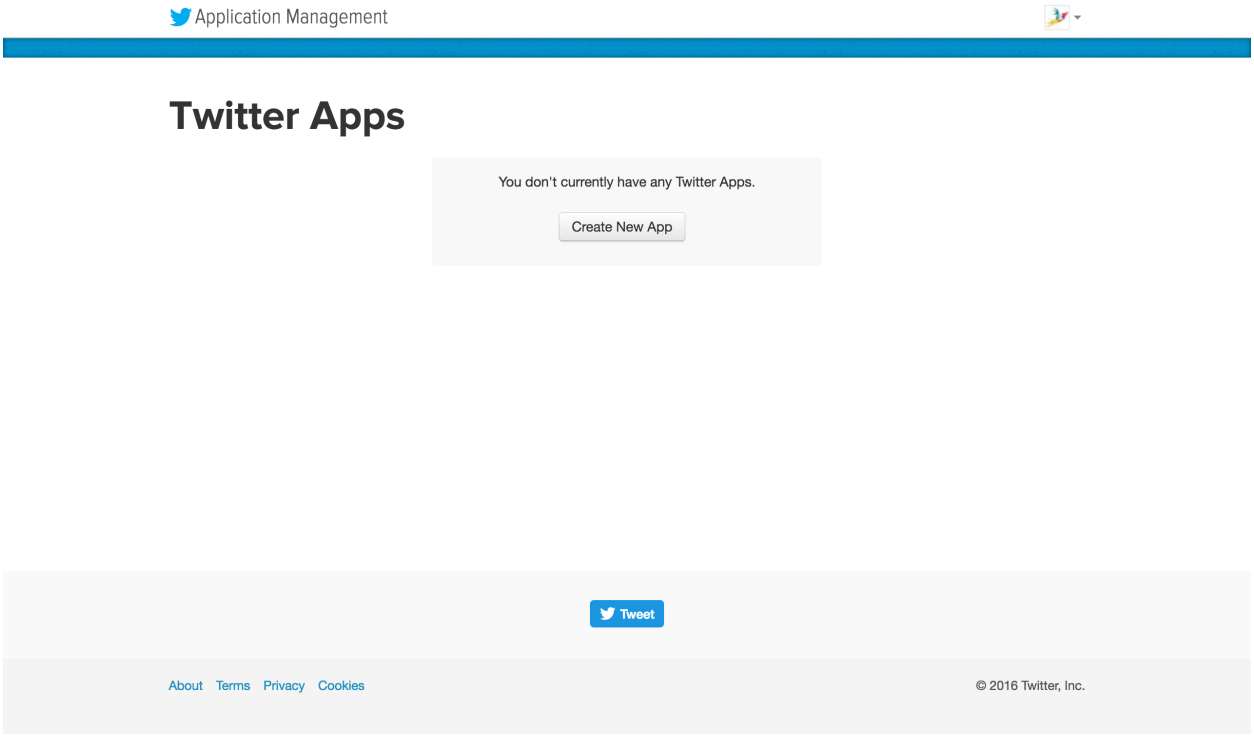
Apply your settings by pressing submit and activate *Authentication via Microsoft*.

30.3.6 Twitter

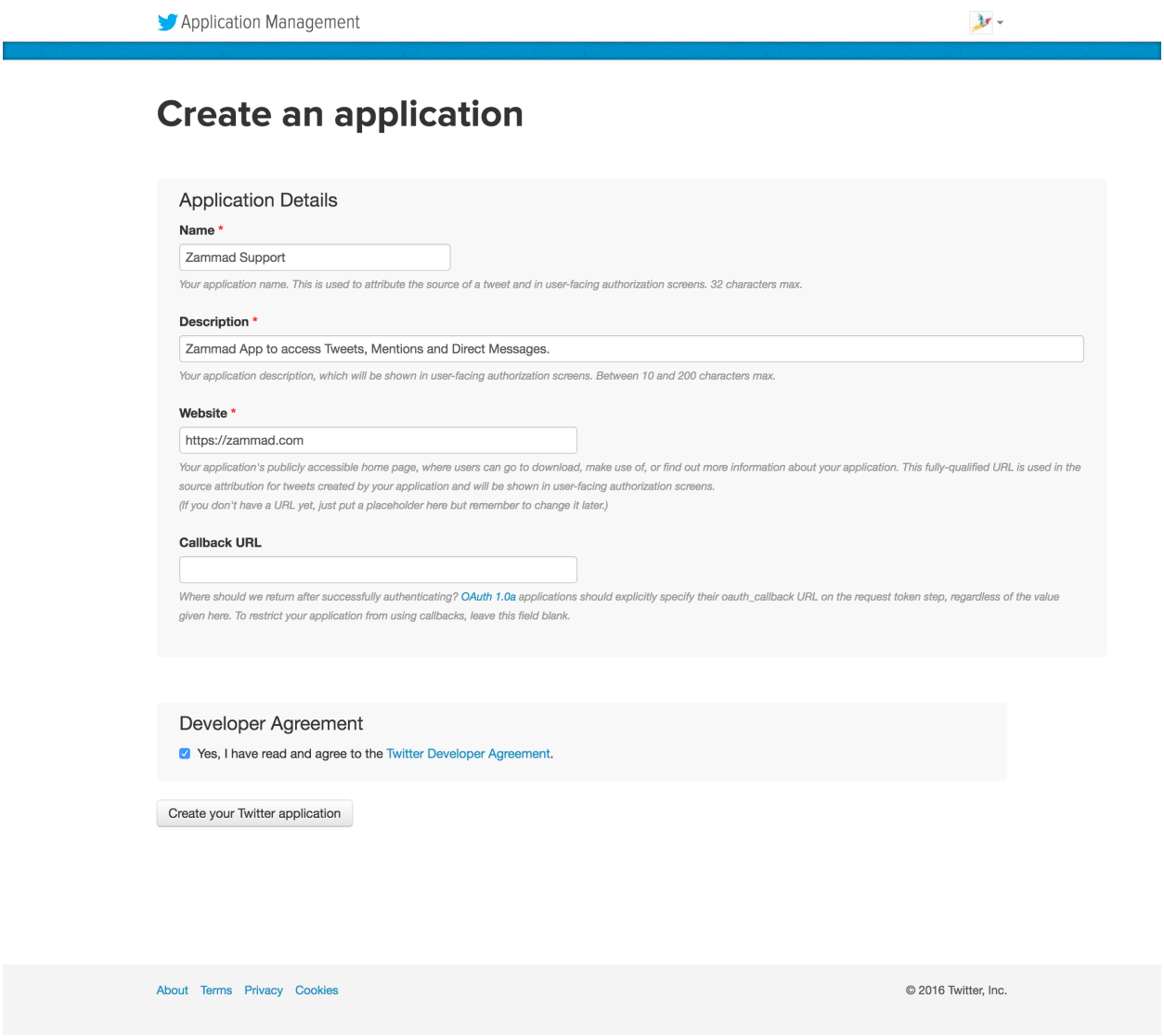
It is possible to create a quick login for your helpdesk via Twitter to do so, you need to follow these steps:

Register Twitter app

Go to https://dev.twitter.com/apps and login with your account.

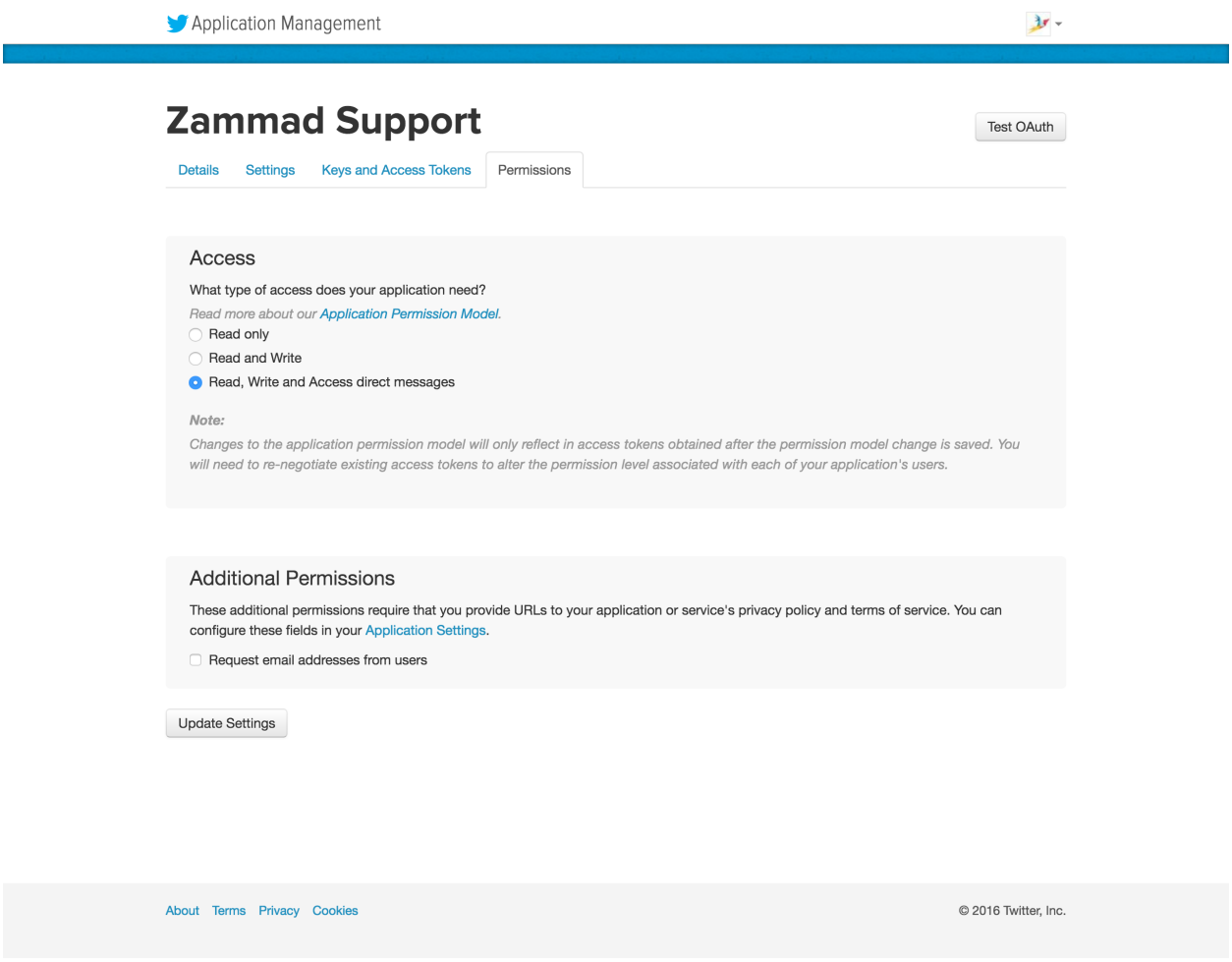


Click on “Create App”

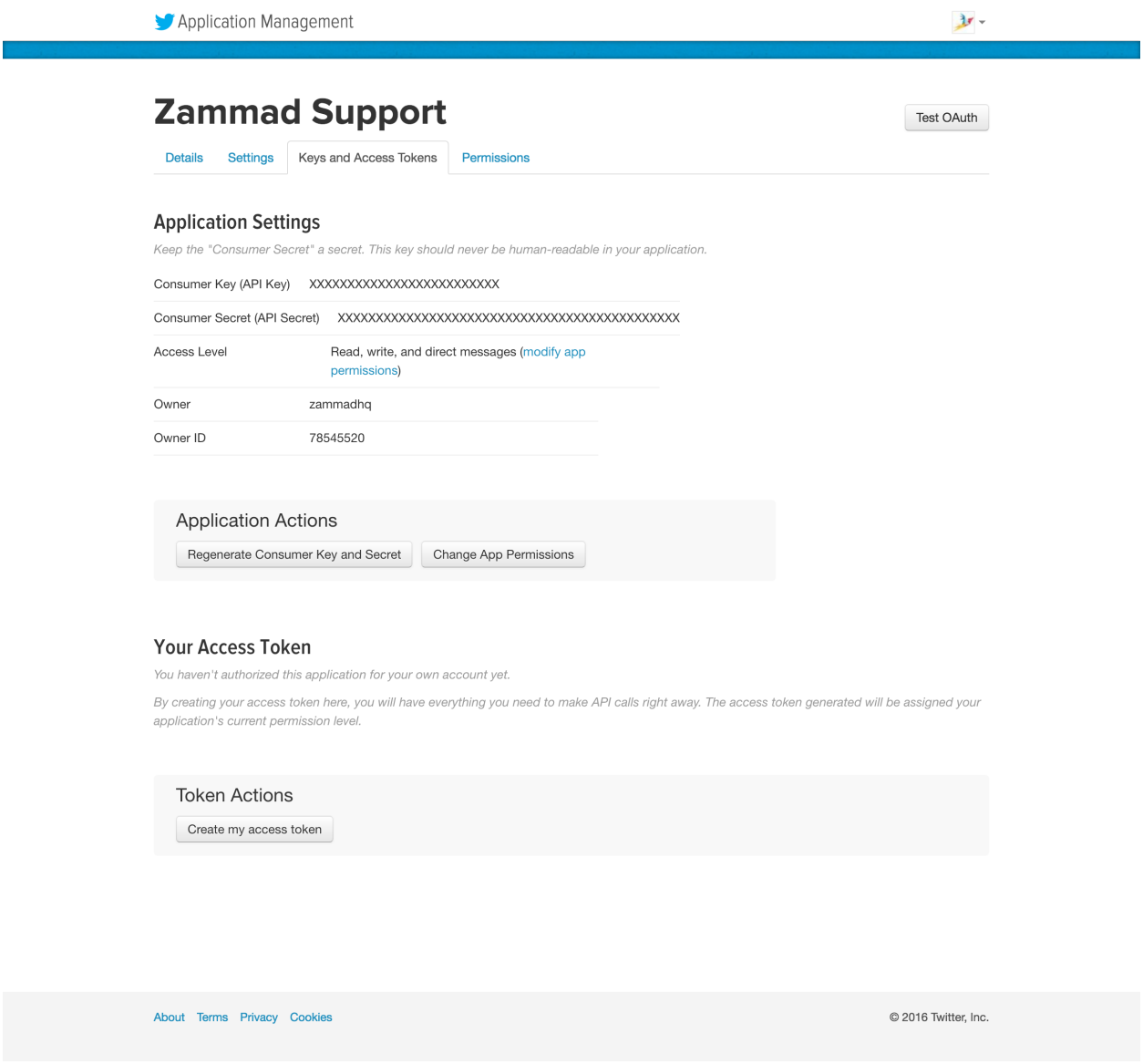


Enter app settings. As "Callback URL" you need to enter "https://zammad_host/api/v1/external_credentials/twitter/callback"

After the app has been created, update the application icon and organization attributes.



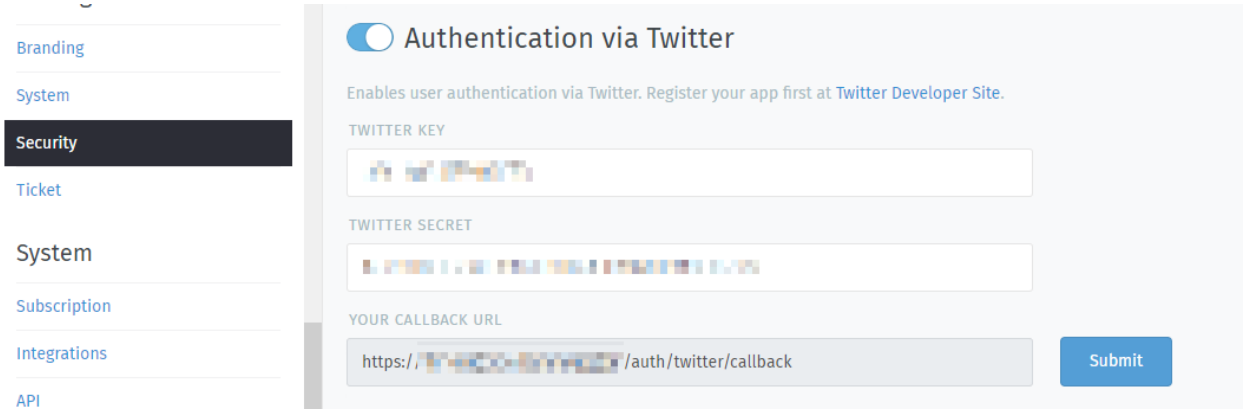
Next we need to set `read, write and access direct messages permissions` for the app.



Go to “Keys and Access Token” tab and note the “Consumer Key” and “Consumer Secret”.

Configure Zammad as Twitter app

Go to “Admin -> Security -> Twitter -> Third Party Applications -> Twitter Section”



Fill in the "Twitter Key" and the "Twitter Secret" and click the "Submit" button.

Now you can link accounts via "Avatar -> Profile -> Link Accounts" or login via Zammad login page.

30.3.7 SAML

Connect your SAML identity provider as a single sign-on (SSO) method.

Note: What is SAML?

SAML is an open standard for SSO authentication (among other things). Sign-ins are shared across multiple **service providers** and managed by a central **identity provider** (IdP).

In this case, the service provider is Zammad, and the IdP is a software service that you either host or subscribe to (*e.g.,* Keycloak, Redhat SSO Server, ADFS, or Okta).

This guide assumes you are already using SAML within your organization (i.e., that your IdP is fully set up).

Step 1: Configure Your IdP

Add Zammad as a client/app

Import Zammad into your IdP using the XML configuration found at `https://your.zammad.domain/auth/saml/metadata`.

Note: What if my IdP doesn't support XML import?

You will have to configure Zammad as a new client/app manually using the above XML metadata file for reference. For instance, when you see this tag:

```
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"␣
↪Location="http://your.zammad.domain/auth/saml/callback" index="0" isDefault="true"/>
```

Set the **Assertion Consumer Service Binding URL** (sometimes also listed as **Valid Redirect URIs**) to `http://your.zammad.domain/auth/saml/callback`.

Set up user attribute mapping

Zammad requests the following user attributes (or "properties") from the IdP:

- Email address (`email`)
- Full name (`name`)
- Given name (`first_name`)
- Family name (`last_name`)

You may need to set up "mappers" (or "mappings") to tell your IdP how user attributes in SAML correspond to those in Zammad. For a more detailed breakdown, refer to the XML metadata file referenced in the previous section.

Per-IdP Instructions

Keycloak

- **To add Zammad as a client,** save the XML configuration to disk (`https://your.zammad.domain/auth/saml/metadata`) and use **Clients > Create > Import** in the Keycloak admin panel.
- **To help Zammad match its own user accounts to Keycloak users,** create a user attribute (or "property") mapper:

Table 1: **Clients > https://your.zammad.domain/auth/saml/metadata > Mappers > Create**

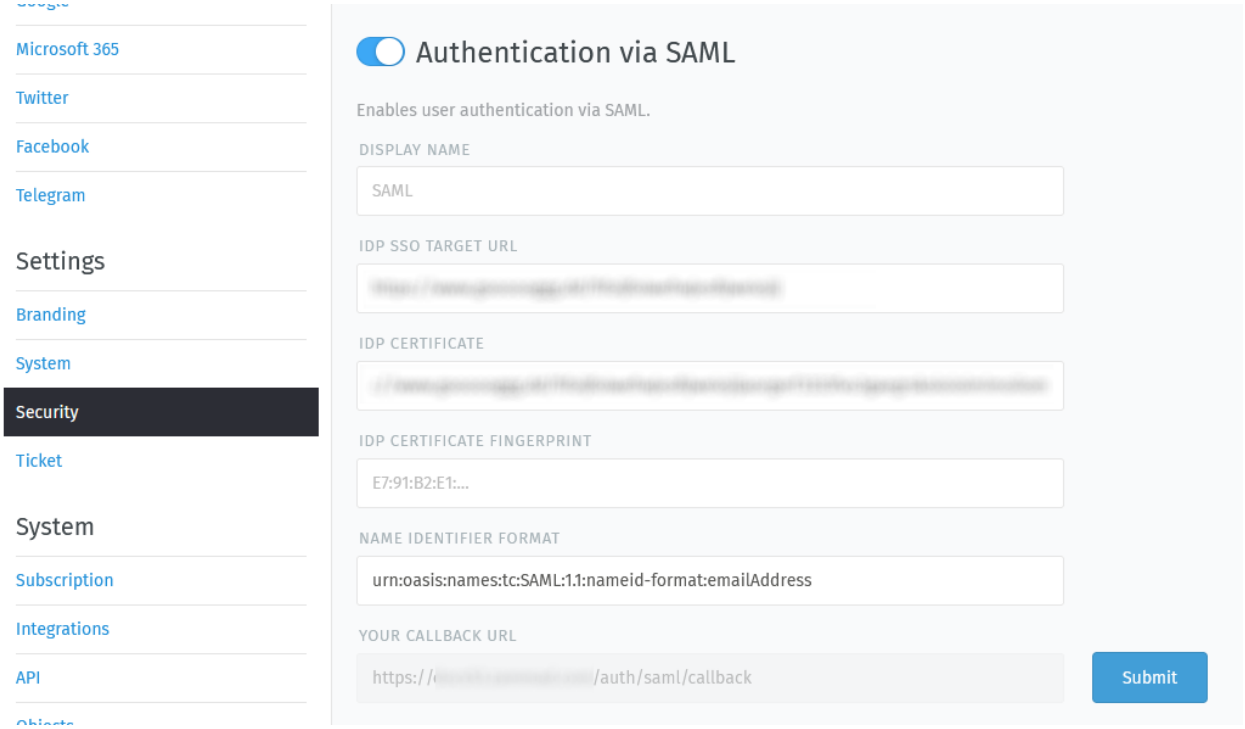
Name	`emailAddress`
Mapper Type	`User Property`
Property	`emailAddress`
SAML Attribute Name	`email`
SAML Attribute NameFormat	`basic`

In the example above, we're telling Zammad that whenever it receives a SAML login request, it should take the `emailAddress` property from Keycloak, look for a Zammad user with the same `email` attribute, and create a new session for that user.

If your Keycloak users' email addresses are stored on another property (*e.g.,* `username`), adjust accordingly.

Step 2: Configure Zammad

Enable SAML and enter your IdP's details in the Admin Panel under **Settings > Security > Third Party Applications > Authentication via SAML**:



Display name Allows you to define a custom button name for SAML. This helps your users to understand better what the button on the login page does.

Defaults to `SAML`.

IDP SSO Target URL This is the target URL Zammad shall redirect to when the user presses the SAML button.

IDP Certificate The public certificate of your IDP for Zammad to verify during the callback phase.

IDP Certificate fingerprint The fingerprint of your IDPs public certificate to verify during callback phase.

Note: For the IdP certificate / certificate fingerprint:

Provide **only one or the other**—do not provide both! (Between the two, we recommend the signing certificate itself: fingerprints use SHA-1, which has been broken for a while now.)

Keycloak users: Find your certificate in the Keycloak admin panel under **Realm Settings > Keys > RSA > Certificate**.

Name Identifier format This is the unique identifiers field type. Usually should be `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress` in any case.

> **Warning:** Zammad expects an email address as unique identifier!

Your callback URL This URL is needed for your IDP configuration so it knows where to redirect to after successful authentication.

See *automatic account linking* for details on how to link existing Zammad accounts to IdP accounts.

Troubleshooting

Automatic account linking doesn't work Have you double-checked your IdP's user attribute mapping configuration?

Note: We're currently missing documentation for the following login providers:

- LinkedIn
- Weibo

Tip: You can deactivate logging in via *Password Login* if any of above mentioned authentication providers are enabled in your instance.

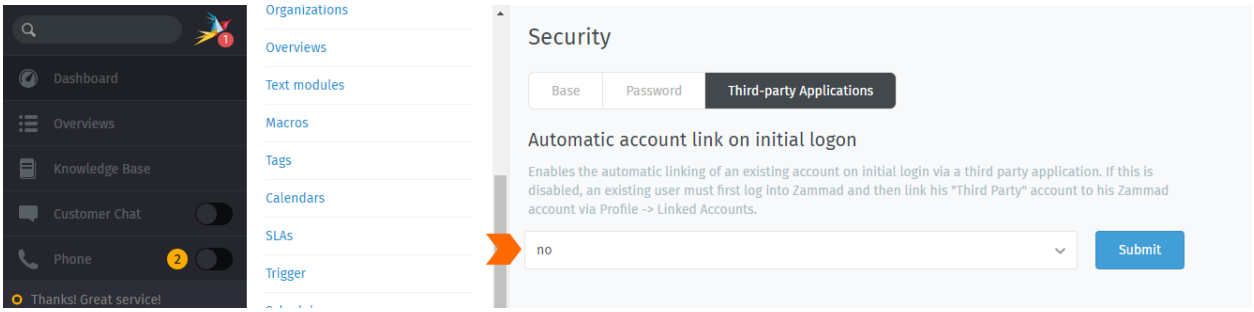
30.3.8 Automatic account link on initial logon

In general there's two possible options for Zammad on how to deal with already known users as they try to authenticate against a third-party application. By default, Zammad will not automatically link "unknown" authentication providers to existing accounts.

This means that the user has to manually link authentication providers to their accounts (for more about this consult the user documentation).

Sometimes this doesn't come in handy as this also means you'll receive error messages about "email address being in use already" for (yet) unknown third-party authentication methods.

If you want to allow your users to always be able to log in, no matter what, you may want to enable `Automatic account link on initial logon`.



CHAPTER

THIRTYONE

TICKET

Note: Additional settings for the ticket composer interface can be found in the *Composer Settings*.

31.1 Base

Ticket Hook (default: `Ticket#`) The identifier for a ticket; *e.g.,* Ticket#, Call#, MyTicket#.

Ticket Hook Position (default: `right`) With this setting you can decide (if) where to insert the ticket number.

Right This setting will add the ticket reference on the right site of the subject.

Example: `Some Subject [Ticket#12345]`

Left This setting will add the ticket reference on the left site of the subject.

Example: `[Ticket#12345] Some Subject`

None This will completely remove ticket references from the subject.

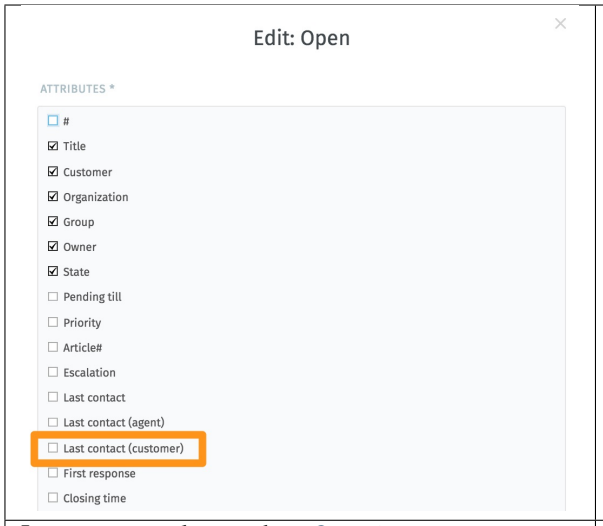
> **Warning:** Please ensure to take a look at *Settings* within the email channel to ensure you have at least one reference that helps Zammad to assign follow-ups correctly.
>
> Disabling this and not setting up any further follow up search will lead to unexpected results!

Ticket Last Contact Behaviour (default: `Use the start time of the last customer thread (which may consist of mu`
This setting changes the way Zammad updates the `Last Contact` value of a ticket. This is fairly important for overviews and what you expect upon ticket updates of a customer.

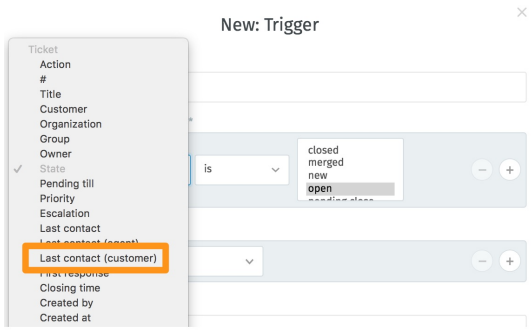
Use the start time of the last customer thread (which may consist of multiple articles). If this option is chosen, Zammad will update the last updated value just once if a customer replies. After that, Zammad will wait for an agent to react.

This comes in handy if you want to work tickets in the order of their first update. This will not change ticket orders if customers decide to "bump" the ticket affected.

Use the time of the very last customer article. When setting this option, Zammad will always update the last updated value. This means that if ordering by `Last Update`, ticket orders will change if customers "bump" the ticket.



Last contact value used on *Overviews*



Last contact value used on *Triggers*

31.2 Number

Ticket Number Format (default: `Increment (SystemID.Counter)`) This setting defines the way Zammad's ticket number will look. Changing this setting in production will cause follow up detection to fail.

Increment (SystemID.Counter) The increment value contains the SystemID (automatically generated during installation) and a increment which counts up with every new Ticket. The increment will automatically get another digit if needed.

Examples: `1010138`, `1010139`

Date (Year.Month.Day.SystemID.Counter) This version of the ticket number is fairly long. However, because it contains the date of ticket creation, it also allows you to see when the ticket has been created. You may feel familiar with this number format if you've worked with OTRS already.

Examples: `201206231010138`, `201206231010139`

OPTIONS The below options allow you to further customize your ticket number. The availability of options differ depending of the chosen ticket number format above.

Checksum (default: `no`) If set to yes, we'll built a checksum to further "unique" the ticket number of your system. If you're dealing with many other ticket systems this might help to prevent false positive follow up detection on tickets.

This will also add another digit to the ticket number.

Min. size of Number (default: `5`) This setting defines the minimum length of your ticket number. If your counter is not yet high enough, Zammad will fill the number with zeros to ensure the wanted length.

Please note that the minimum length is 3 (SystemID + at least one digit if below 10). Also, if the length is no longer long enough, the number will automatically get one digit longer. Ticket-Numbers **do not overflow**, thus it's save to keep it at 5 or set it lower if wanted.

Note: This option is only available if ticket number format is set to `Increment`!

31.3 Auto-Assignment

In a larger Zammad environment, it happens that several agents open the same ticket at the same time. Although collision detection is then effective, the efficiency of processing can be increased by means of the automatic assignment of tickets when a ticket is opened.

Note: Auto Assignment only kicks in if the ticket has no owner yet. By default the agent can always reset the ticket owner to - if needed.

The automatic assignment of tickets can be activated and configured in the admin area under within Settings -> Ticket -> Auto assignment.

If you want to use this function for only specific tickets, you can configure the conditions accordingly to meet your requirement. By default the condition affects all tickets with the state `open`.

If you need to exclude users (e.g. a group leader), you can search and select the desired agents in the `Exception Users` list.

Note: The search function in this area is only effective, if you have too many agents to display at once to help you with the configuration.

CHAPTER

THIRTYTWO

SUBSCRIPTION (SAAS)

The subscription settings page allows you to configure your instances package and number of agents required.

> **Warning: Hosted environment specific**
>
> This setting section is only available for Hosted setups. If you're looking for on premise support contracts, please see the Zammad pricing page.

32.1 Plan

Subscription On the top of the subscription summary you'll find the number of used and available (booked) agent seats. By using the *see more* link you can also expand an agent list to see the currently configured, active agents.

> **Tip:** You can learn more on how to manage your agents within *user management*.
>
> This list *does not* count accounts with admin permissions only.

Plan This section gives you an overview of the available plans and their functions / limits. If you require a more detailed table, check our detailed pricing table for more.

Your selection in this step will decide on the base agent price and e.g. agent limits that might apply in the summary step.

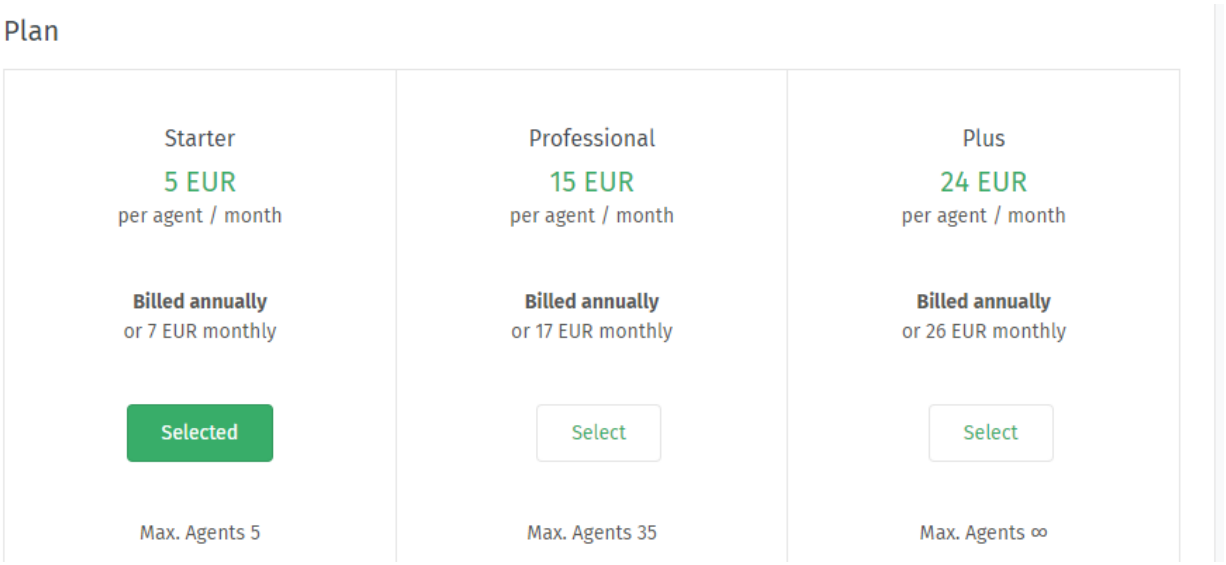
The green *Selected* button will also tell you what plan you're currently in.

> **Hint:** Trial instances are running at Professional with 3 agents. The plan cannot be changed during the trial for technical reasons, if you require more agents for testing, please contact our sales team with a reason for the raise request.

Summary In this section you can adjust the settings of the previous selected plan.

Billing cycle You can choose between either monthly or yearly billing. The price per agent will be cheaper if you decide for yearly billing.

> **Tip:** If you're still trying out things and e.g. are unsure of the right package for your instance, you may want to choose monthly first and then upgrade to yearly when you're sure.



Note: Note that upgrading (more agents, higher package) is always possible, however, downgrading will only happen when your billing period renews.

Upgrading resets your billing period and existing credit will be deducted from the new total.

Plan: (Starter|Professional|Plus) - Agents Choose the number of agents you require for your instance.

Keep in mind that some packages may have agent limits. Depending on the previous chosen package, you won't be able to exceed specific agent numbers. Also note that setting less agents than currently configured is not possible.

Total This will be the net total you're going to be billed.

By using the *Update subscription* button, the instance package will be adjusted immediately. If you didn't provide a payment method yet, Zammad will request you to provide one at this point.

> **Warning:** In case you're still within the trial period, this means that the trial will be over immediately!
>
> Down- or upgrading during the trial period *is not* possible.

Payment method You can pay via credit card or SEPA mandate.

Credit card Simply follow the dialogue by clicking *authorize* and confirm -if required by your bank- your consent to Zammad using your card for the payments.

SEPA Provide the requested information of the bank account holder and update if you're ready. You'll then receive an email with the SEPA mandate.

Technically there's also the possibility to pay via invoice, however - this method only applies to a fairly low number of instances. Please contact our sales team for more information.

32.2 Billing

Within the billing tab you can control all billing relevant information like invoices, billing address and the option to cancel your subscription.

Billing information All adjusted billing information below only affect future invoices. If your invoice was issued wrong, please contact our sales team.

Billing address Provide your company address here, make sure to include the companies name in case required. This address will be referenced on your invoice.

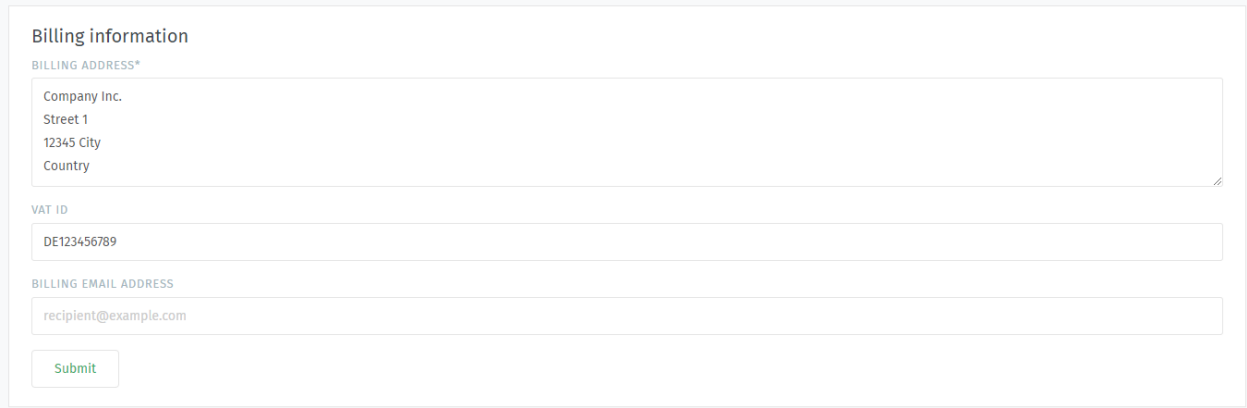
VAT ID Provide your VAT ID here. If applicable your invoice will not contain german VAT. Please make sure to pay the VAT in your country as required.

Billing Email Address Usually the instance owner (the person that registered the instance) will receive all emails including the invoices. In case you have your own billing department, you can provide their email address here.

All billing relevant information will then be sent to this email address.

Hint: Invoices are sent as attachment (PDF) to this email address.

Don't forget to press the *Submit* button after you changed above settings.



Payment history The payment history will display a history of all paid invoices. At this point you can also download these in case you need them again.

Note: You will only find *paid* invoices here. Invoices that are to be billed are not available before the payment is completed.

Date Date of the invoice creation.

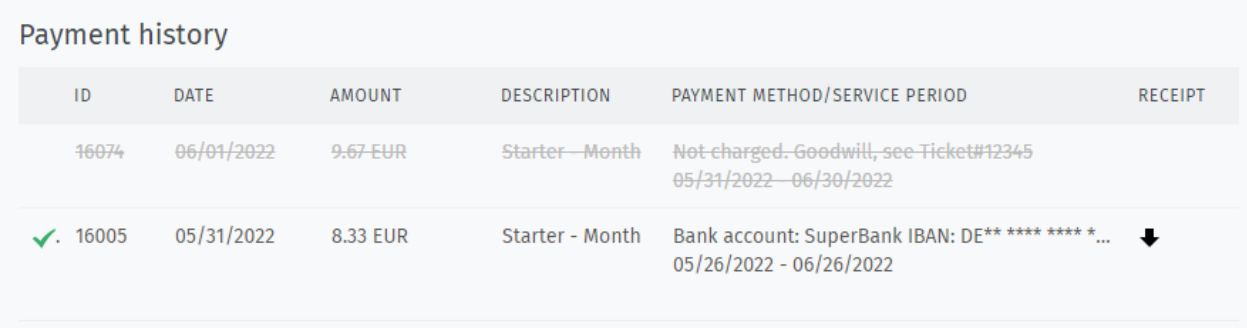
Amount Invoice total including VAT if applicable.

Description Contains contract period (monthly or yearly) and hosted plan for the subscription period in question.

Payment method / Service period Used bank account or credit card as well as the subscription period the invoice is about.

Note: It might happen that the invoice date and subscription period differ. This is nothing to worry about, the subscription periods will be accounted later on.

Receipt Use the arrow to download the invoice in question. You can download all available invoices any time you need to here!



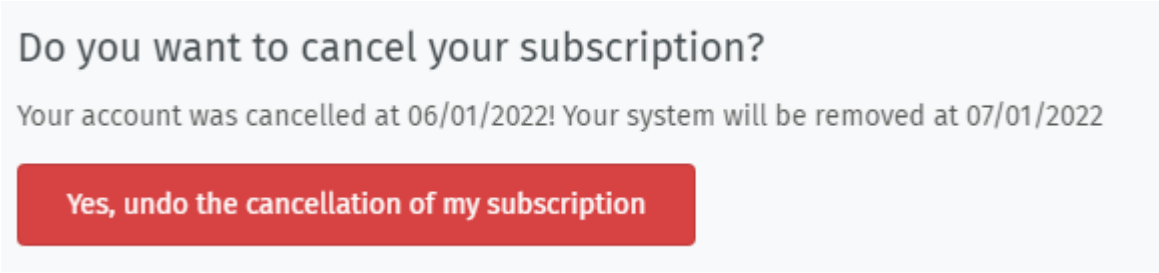
Do you want to cancel your subscription? In case you no longer want to use Zammad's SaaS, you can cancel your subscription by using the red *Yes, please cancel my subscription* button.

Your subscription will end the day after your trial or billing period ends.

Warning: We will remind you about your cancelled subscription several times up front. After the subscription ended all instance data will be removed. A restore *is not* possible after that point!

Hint: Cancelled by accident?

You can always undo the cancellation via the *Yes, undo the cancellation of my subscription* button up to the last subscription day!



The subscription section currently consists of two tabs: Plan & Billing. For your overview we've divided those two tabs into independent sub pages:

Subscription plan Everything affecting your instance subscription functions like number of agents, package and payment method.

Subscription billing Everything regarding billing address, invoices and account cancellation.

32.3 FAQ

I set up a trial account but am missing functions to test Sorry. The trial instance is running within the professional package allowing up to three agents.

Can I change package within the trial? No. As soon as the package is changed the subscription begins.

What happens to my trial instance after the trial period ended? Your instance will automatically be cancelled for you. Please see *What happens to my instance after it has been cancelled?* for more.

What happens to my instance after it has been cancelled? That depends slightly on your instance state:

Trial instance If you're still on trial, we will ice your instance and remind you once again about your instance. We then will wait some days and remove your instance from our system. This is the last time we will inform you by Email.

Paid instance If you're a paying customer and cancelled your subscription, the instance removal will take place the day after your last subscription day.

Can removed instances be restored? No. Once your instance has been removed, a restoration is not possible.

CHAPTER

THIRTYTHREE

INTEGRATIONS

Zammad offers numerous integrations that add rich features to your instance.

Note: We're still working on this part of our documentation, stay tight!

33.1 Integrations for phone systems

Hint: Your VoIP provider or telephony system is not listed? Possibly your provider supports Zammad by using the generic CTI - if you're unsure ask your provider.

Provider does not support Zammad? Consider creating a feature request in the Zammad Community.

33.1.1 CTI (generic)

This integration enables Zammad to provide a caller log to your agents. With this your agents can greet your customers by their name and are supported by Zammad with automatically opening a new ticket dialogue or the user profile.

If you want to learn more on how your agents can use this function, please refer the user documentation.

> **Note:** Automatically opening new ticket dialogues or user profiles requires agent to extension mapping - see more below.

Limitations

Please note the following limitations to reduce confusion later on:

- CTI integrations provide caller log functions only.
- This integration does not provide any VoIP functionalities; you can't make phone calls from within Zammad.
- If your browser supports `tel`-URLs, you can click on phone numbers for automatic dialing. This requires additional software / configuration on your agent's computers.

Requirements

Please provide the following requirements:

- A telephone system that supports webhooks (outbound) (best option are those that allow customizing the calls).
- A unique Call-ID during the call session.
- Call event data (both numbers and direction).
- Your Zammad instance must be reachable for your telephony system.

If you want to learn more on what kind of requests are supported by Zammad and what it expects, please consult our CTI-API documentation.

Available settings

Hint: Click the button next to the `CTI (generic)` heading to activate or deactivate this function.

Endpoint Settings Zammad will list your generic CTI endpoint here. It contains a unique token so ensure to keep this URL save.

You'll need this endpoint for your PBX to talk to Zammad, see CTI-API documentation.

Important: All following options **do not save automatically**. Always use the Save button on the lower end of the integration page!

Call Settings

Inbound This option allows you to block specific incoming caller IDs. It allows you to temporarily reject e.g. spam callers without having to contact providers or PBX administrators.

Caller ID to block Provide caller IDs to block in `E.164` format.

Note Provide a meaningful note for your fellow administrators to remind yourself why you've chosen to block the number.

Note: Your telephony system has to support this function. Zammad will send a `reject` response which will cause your telephony system to hang up the call.

To callers this usually will feel like the line is busy.

Outbound In many cases you may want to use a different caller ID depending on the destination you're calling. This may apply due to specific connection rates to other countries or because you want your customer to feel you're calling from the same country.

Note: This option expects `E.164` number formats.

Destination caller ID The caller ID or number you're trying to call.

Tip: You can use wildcards for e.g. country specific outbound numbers like:

- `49*` for Germany

- `4930*` for Berlin / Germany landlines
- `33*` for France

Set Outbound caller ID The outbound caller ID to set (the number your customer sees on his display) in `E.164` number format.

Note Provide a short description for fellow administrators.

Note: This option requires your PBX to send a specific request to Zammad *before dialing*. Please consult the CTI API in case you're not sure.

Other Settings Below you can find all available additional settings for this CTI integration. For your overview we're sorting them by appearance and reference their description first.

Default caller ID for outbound calls In many cases you may want to use a different caller ID depending on the destination you're calling. This may apply due to specific connection rates to other countries or because you want your customer to feel you're calling from the same country.

Note: This option expects `E.164` number formats.

Note: This option requires your PBX to send a specific request to Zammad *before dialing*. Please consult the CTI API in case you're not sure.

Shown records in caller log Allows you to set the number of shown caller log entries for all users. You can choose from the following values:

- 60 (default)
- 120
- 180
- 240
- 300

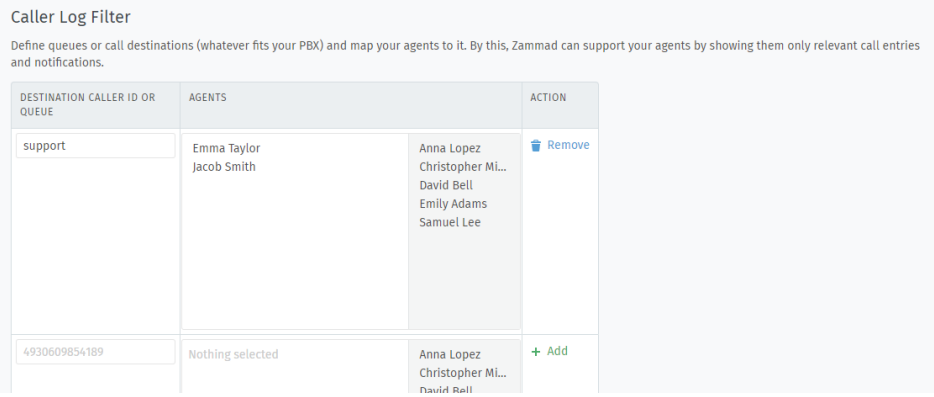
> **Warning: Potential performance issue**
>
> Setting this setting higher than 60 may cause serious performance issues on very busy instances. Keep in mind that this setting causes Zammad to poll and send up to 300 records to *all active agent sessions* in very short time periods.

Caller Log Filter This function allows you to provide call information based on e.g. queues only to agents that really need the information.

Why? If you have a team for several countries or departments, you don't want to bug your agents from other departments. Leaving these options empty will fallback to showing everything to everyone.

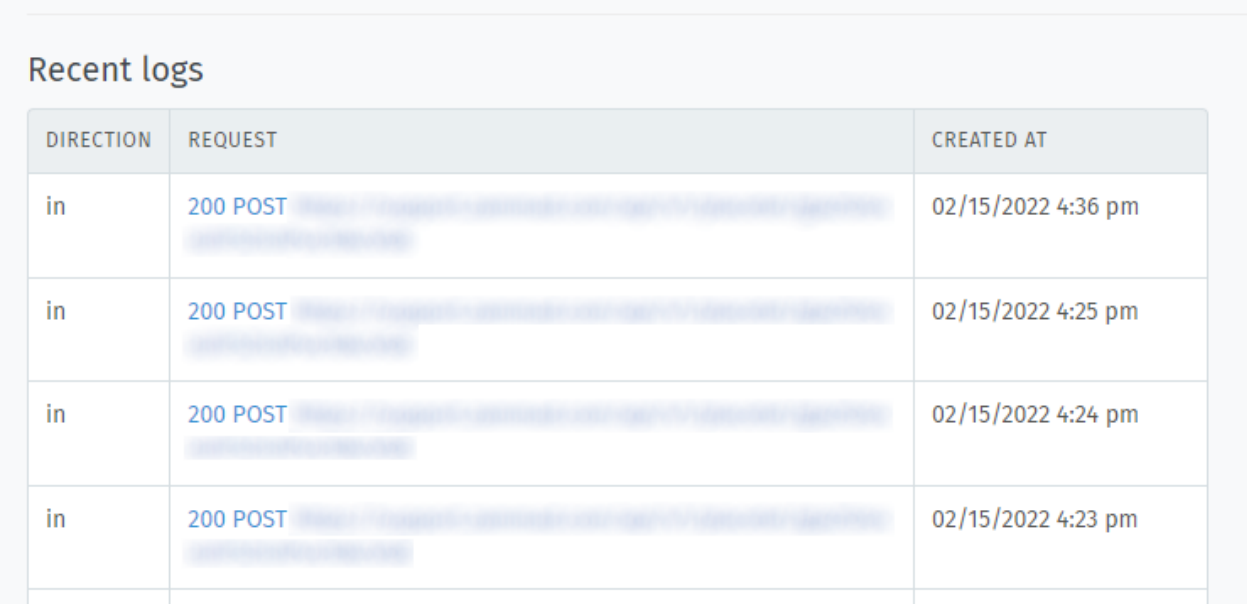
Destination caller ID or Queue This depends on your PBX and usually is either a queue ID, phone number or extension.

Agents Select the agents that are responsible for the group. These agents will then see caller log entries and call notifications fitting to said queue.



Recent Logs

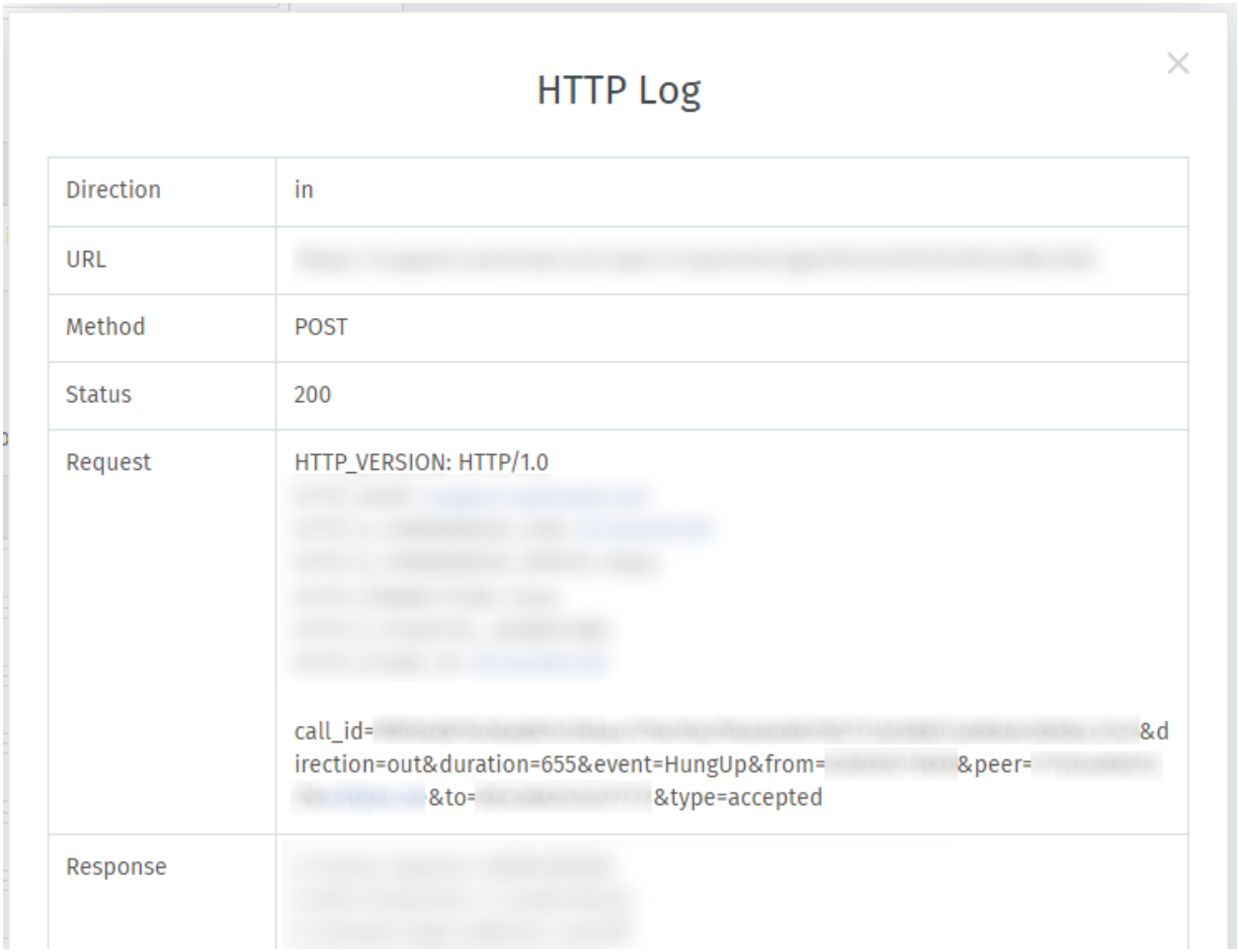
With recent logs Zammad allows you to view the latest calls for the CTI functionality. This usually comes in handy, when you're looking for errors.



I'm just here to clear floats up.

By clicking on the entry of interest, Zammad will provide more details on the call in question. You'll see the payload it received and also the response that was sent.

x


HTTP Log
Direction
in
URL
Method
POST
Status
200
Request
HTTP_VERSION: HTTP/1.0
call_id=
&direction=out&duration=655&event=HungUp&from=
&peer=
&to=
&type=accepted
Response

Troubleshooting

My Phone page stays blank, signalling does work… If you've made sure that signalling works (check Recent logs) and your Caller Log still stays empty, please ensure that you only configured one CTI integration version. Specifically defining more than one agent mapping on the different integration variants will be the issue.

Clear the not needed agent mapping and reload your browser page.

33.1.2 Placetel CTI

Setup Placetel connection for Zammad

Note: This configuration step requires a *full* administrative Placetel account. You may receive forbidden error messages with Placetel in case your permissions are not high enough.

The following actions can be configured via the Placetel web interface.

Step 1: Activate Zammad integration Within Integrations, scroll down to Partner integrations and select Zammad.

You can alternatively filter by "Ticket Systems" to reduce the number of entries on the page. You'll still want to look for Partner integrations.

Within the Zammad integration now press "Activate". A new tab `API` becomes available - open this tab.

Now tick "Enable Call Control / Notify API" and paste the Placetel endpoint from your Zammad instance into the field "URL of your API endpoint". Save to apply the settings

Step 2: Generate API Token for Placetel Go back to the integrations page and scroll down to "Web API". Generate a new API token by using the "Create a new API token" button.

Note: If you already generated a token either use your existing token or reset it by using above mentioned button. Placetel will ask you to conform this reset.

Please keep in mind that existing API scripts may no longer work due to token resets!

Copy the provided API token and insert it into the "API Token" field within Zammad's Placetel integration.

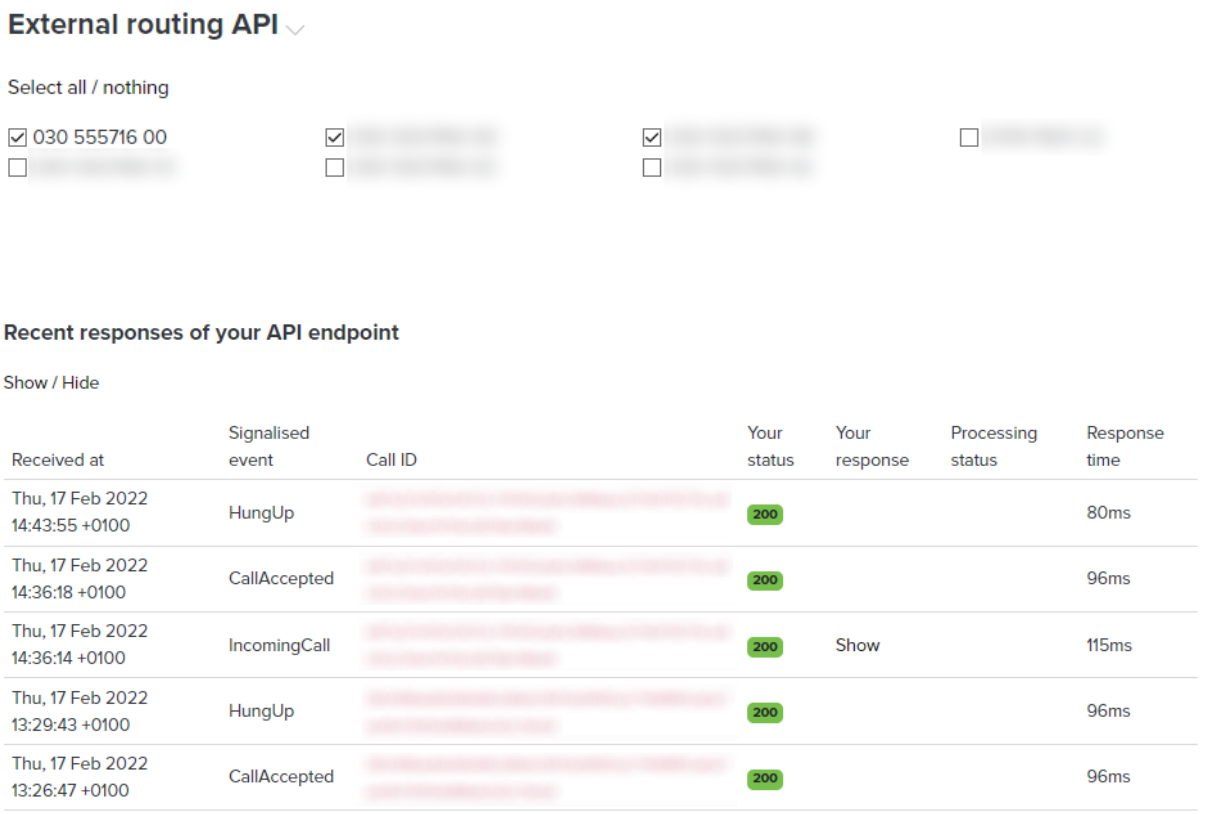
Apply your changes by using the "Save" button on the bottom of the Placetel integration page and activate the Placetel integration.

Step 3: Restrict the numbers to notify on Having a lot of numbers that shouldn't be used for notifying Zammad? Within the the Integrations page of the Placetel web interface, go to "Notify API".

Lower on the page Placetel allows you to restrict the numbers to notify on. You'll find this within the "External routing API" part.

Hint: This menu point also provides a API request log from Placetel view. Just open "Recent responses of your API endpoint" to learn more.

If you want to see Zammad's perspective, use the "Recent Logs" part from within the Placetel integration page.



Step 4 (optional): Further configurations for Placetel If needed, you can now configure Zammad's Placetel integration in more detail. You can learn more about your options here: *Placetel integration settings*.

This integration enables Zammad to provide a caller log to your agents. With this your agents can greet your customers by their name and are supported by Zammad with automatically opening a new ticket dialogue or the user profile.

If you want to learn more on how your agents can use this function, please refer the user documentation.

Note: Automatically opening new ticket dialogues or user profiles requires agent to extension mapping - see more below.

Limitations

Please note the following limitations to reduce confusion later on:

- CTI integrations provide caller log functions only.
- This integration does not provide any VoIP functionalities; you can't make phone calls from within Zammad.
- If your browser supports `tel`-URLs, you can click on phone numbers for automatic dialing. This requires additional software / configuration on your agent's computers.

Requirements

Please provide the following requirements:

- You need an administrative Placetel account for your organization.
- Your Zammad instance must be allowed to communicate to external services.
- Placetel must be able to reach your Zammad instance.

Setup Placetel connection for Zammad Learn how to configure Placetel to enable Zammad and Placetel to communicate with each other.

Available settings

Hint: Click the button next to the `Placetel` heading to activate or deactivate this function.

Important: All following options **do not save automatically**. Always use the Save button on the lower end of the integration page!

Endpoint Settings The here listed endpoint settings are relevant for the *integration configuration with Placetel*.

Endpoint This endpoint will be required for the Zammad integration within the Placetel web interface.

API Token You'll receive this token within the `Web API` menu. Make sure to copy this value, it's only shown once!

Call Settings

Inbound This option allows you to block specific incoming caller IDs. It allows you to temporarily reject e.g. spam callers without having to contact providers or PBX administrators.

Caller ID to block Provide caller IDs to block in `E.164` format.

Note Provide a meaningful note for your fellow administrators to remind yourself why you've chosen to block the number.

Other Settings Below you can find all available additional settings for this CTI integration. For your overview we're sorting them by appearance and reference their description first.

Shown records in caller log Allows you to set the number of shown caller log entries for all users. You can choose from the following values:

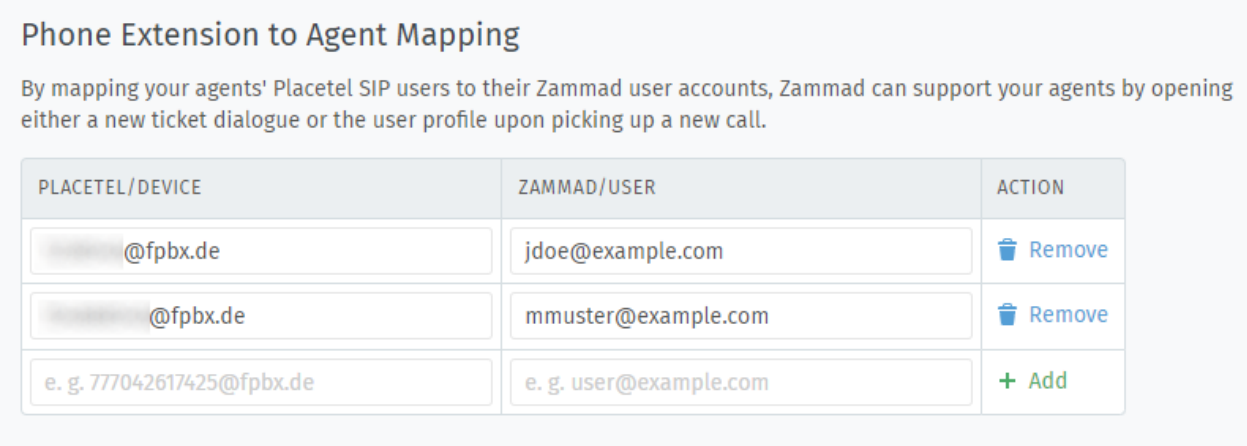
- 60 (default)
- 120
- 180
- 240
- 300

> **Warning: Potential performance issue**
>
> Setting this setting higher than 60 may cause serious performance issues on very busy instances. Keep in mind that this setting causes Zammad to poll and send up to 300 records to *all active agent sessions* in very short time periods.

Phone Extension to Agent Mapping By mapping your agents extension to their existing Zammad users, Zammad can provide a new ticket dialogue or open the user profile for the agent that picks up the call.

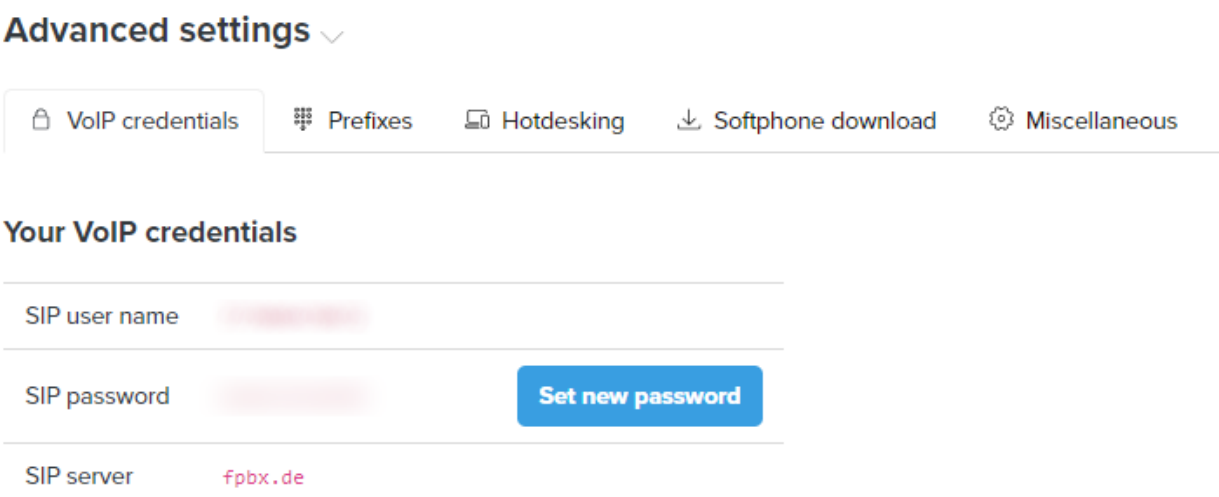
This speeds up ticket aiding, no matter if it's for existing tickets or new ones!

Note: The agent perspective is described within our user documentation.



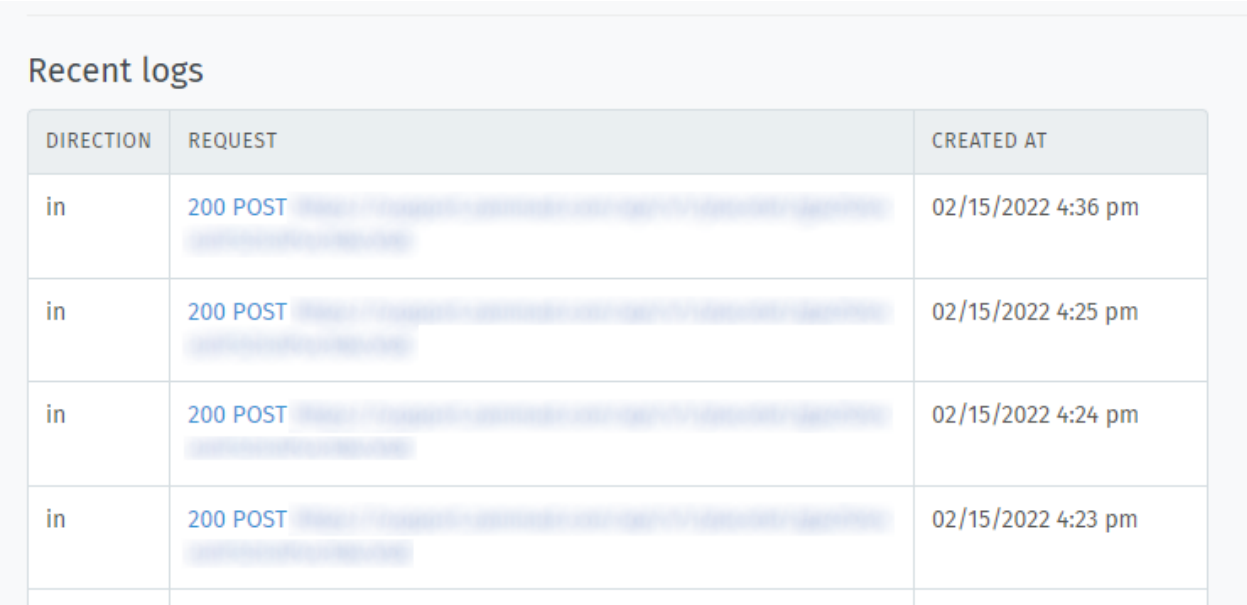
Hint: You can find your agents Placetel username combination required within PBX → VoIP destinations. Within the "Advanced settings" section you're looking for "SIP user name" and "SIP server".

Combine these two like so: `<sip-user-name>@<sip-server>`.



Recent Logs

With recent logs Zammad allows you to view the latest calls for the CTI functionality. This usually comes in handy, when you're looking for errors.



I'm just here to clear floats up.

By clicking on the entry of interest, Zammad will provide more details on the call in question. You'll see the payload it received and also the response that was sent.

x

Troubleshooting

My Phone page stays blank, signalling does work… If you've made sure that signalling works (check Recent logs) and your Caller Log still stays empty, please ensure that you only configured one CTI integration version. Specifically defining more than one agent mapping on the different integration variants will be the issue.

Clear the not needed agent mapping and reload your browser page.

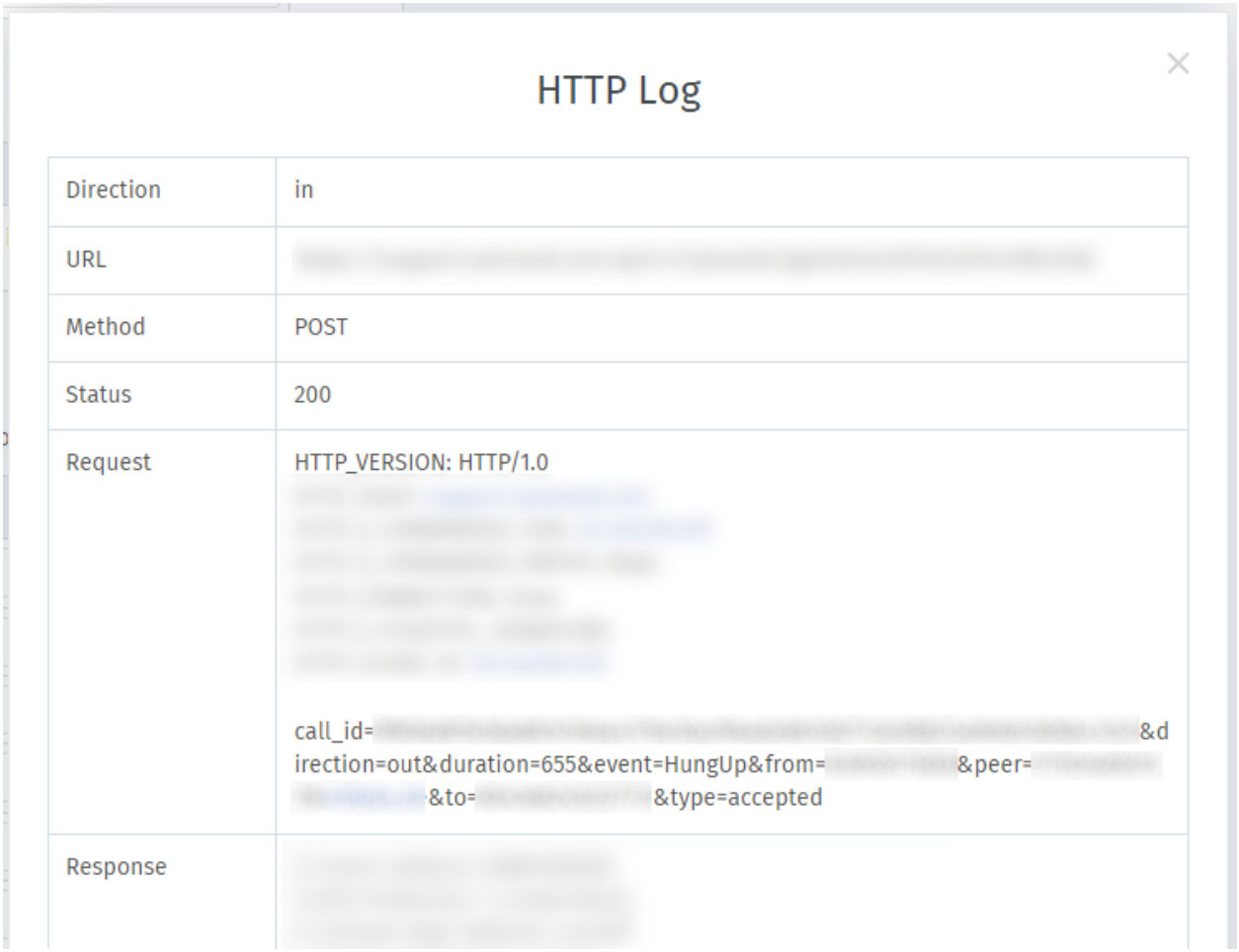
33.1.3 Sipgate (sipgate.io)

Setup Sipgate connection for Zammad

Note:

Sipgate has no english web interface which is why this documentation page is mixing up languages badly.
Please also note that the availability of API addons highly depends on your package trier. Usage of sipgate.io packages *is not free*, please check their pricing page before!

Step 1: Book sipgate.io package


HTTP Log
Direction
in
URL
Method
POST
Status
200
Request
HTTP_VERSION: HTTP/1.0
call_id=
&direction=out&duration=655&event=HungUp&from=
&peer=
&to=
&type=accepted
Response

Hint: Skip to step 2 if you already have the package booked!

Login to an administrative Sipgate account and navigate to `Accountverwaltung`. You'll see several different options depending on your booked packages. Select `Verträge & Produkte` to continue.

Scroll down to the section `Zusätzliche Produkte buchen` and look for `sipgate.io` - select this product by using the `Produkte anzeigen`-Button.

On the next page select either one of the `sipgate.io` packages or `Push-API Package Free`. Follow the dialogue by booking the addon. You'll be returned to your contract overview and now should see the selected addon in your list.

Note: The availability for `sipgate.io` packages and their levels highly depends on the overall account type and product you've booked with Sipgate.

Step 2: Configure webhook for Zammad Within your `Accountverwaltung` navicate to your product `sipgate.io`. In the newly opened tab, switch from "Clients" to "Webhooks" and paste the endpoint URLs from your Zammad instance like so:

- Inbound endpoint to "Incoming"
- Outbound endpoint to "Outgoing"

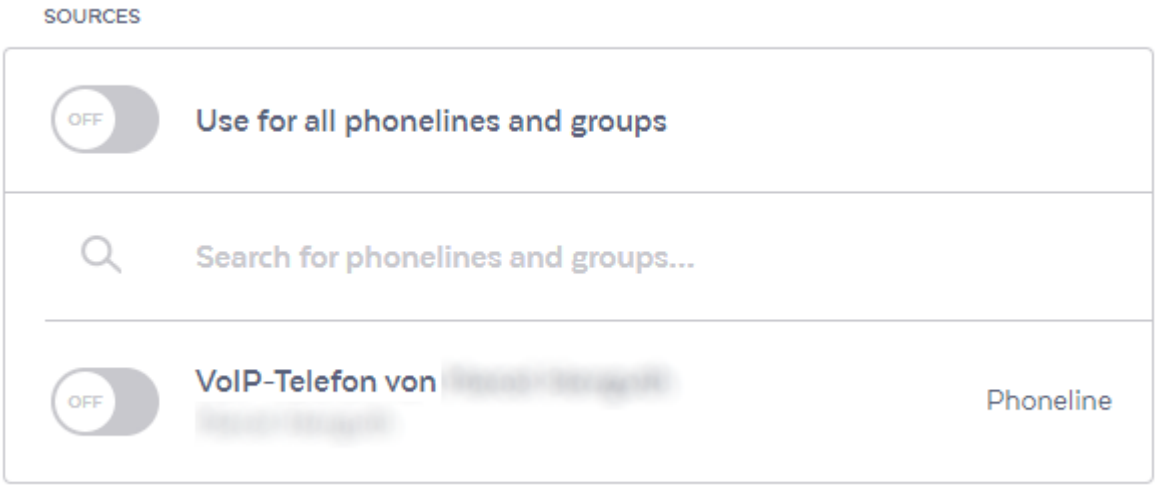
Note: Ensure to select at least one call group or phoneline within "Sources". Other wise Sipgate will not indicate any incoming or outgoing calls to Zammad.

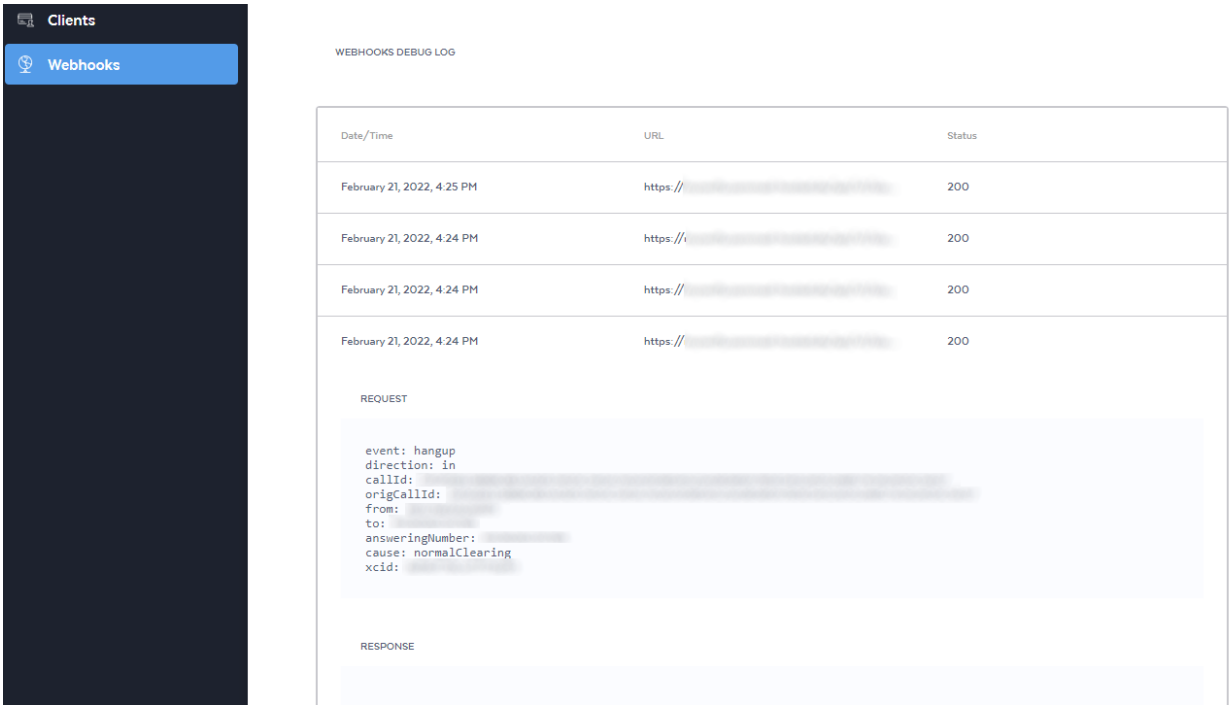
Step 3: Restrict the numbers to notify on Having a lot of numbers that shouldn't be used for notifying Zammad? Within the Webhooks → URLs section of Sipgate you can select which *sources* Sipgate should notify Zammad about in- and outgoing calls.

Use either specific phone lines or use the option "Use for all phonelines and groups" to notify Zammad about all existing lines of your Sipgate account.

Hint: This section also allows you to enable a Debug log.

After enabling you can use the Debug log section to see all sent webhook calls to Zammad. You'll also can see the response.





Step 4 (optional): Further configurations for Sipgate If needed, you can now configure Zammad's Sipgate integration in more detail. You can learn more about your options here: *Sipgate integration settings*.

This integration enables Zammad to provide a caller log to your agents. With this your agents can greet your customers by their name and are supported by Zammad with automatically opening a new ticket dialogue or the user profile.

If you want to learn more on how your agents can use this function, please refer the user documentation.

Note: Automatically opening new ticket dialogues or user profiles requires agent to extension mapping -

see more below.

Limitations

Please note the following limitations to reduce confusion later on:

- CTI integrations provide caller log functions only.
- This integration does not provide any VoIP functionalities; you can't make phone calls from within Zammad.
- If your browser supports `tel`-URLs, you can click on phone numbers for automatic dialing. This requires additional software / configuration on your agent's computers.

Requirements

Please provide the following requirements:

- You need an administrative Sipgate account for your organization.
- **Zammad requires a `sipgate.io` addon from the feature store.**
  - Please note that API calls are not free with Sipgate. Rates may apply and differ from account to account.
- Your Zammad instance must be allowed to communicate to external services.
- Sipgate must be able to reach your Zammad instance.

Setup Sipgate connection for Zammad Learn how to configure Sipgate to enable Zammad and Sipgate to communicate with each other.

Available settings

Hint: Click the button next to the `sipgate.io` heading to activate or deactivate this function.

Endpoint Settings Below endpoint settings affect the *sipgate.io configuration*.

Inbound This endpoint is required for incoming call hooks.

Outbound This endpoint is required for outgoing call hooks.

Important: All following options **do not save automatically**. Always use the Save button on the lower end of the integration page!

Call Settings

Inbound This option allows you to block specific incoming caller IDs. It allows you to temporarily reject e.g. spam callers without having to contact providers or PBX administrators.

Caller ID to block Provide caller IDs to block in `E.164` format.

Note Provide a meaningful note for your fellow administrators to remind yourself why you've chosen to block the number.

Outbound In many cases you may want to use a different caller ID depending on the destination you're calling. This may apply due to specific connection rates to other countries or because you want your customer to feel you're calling from the same country.

Note: This option expects `E.164` number formats.

Destination caller ID The caller ID or number you're trying to call.

Tip: You can use wildcards for e.g. country specific outbound numbers like:

- `49*` for Germany
- `4930*` for Berlin / Germany landlines
- `33*` for France

Set Outbound caller ID The outbound caller ID to set (the number your customer sees on his display) in `E.164` number format.

Note Provide a short description for fellow administrators.

Other Settings Below you can find all available additional settings for this CTI integration. For your overview we're sorting them by appearance and reference their description first.

Default caller ID for outbound calls In many cases you may want to use a different caller ID depending on the destination you're calling. This may apply due to specific connection rates to other countries or because you want your customer to feel you're calling from the same country.

Note: This option expects `E.164` number formats.

Shown records in caller log Allows you to set the number of shown caller log entries for all users. You can choose from the following values:

- 60 (default)
- 120
- 180
- 240
- 300

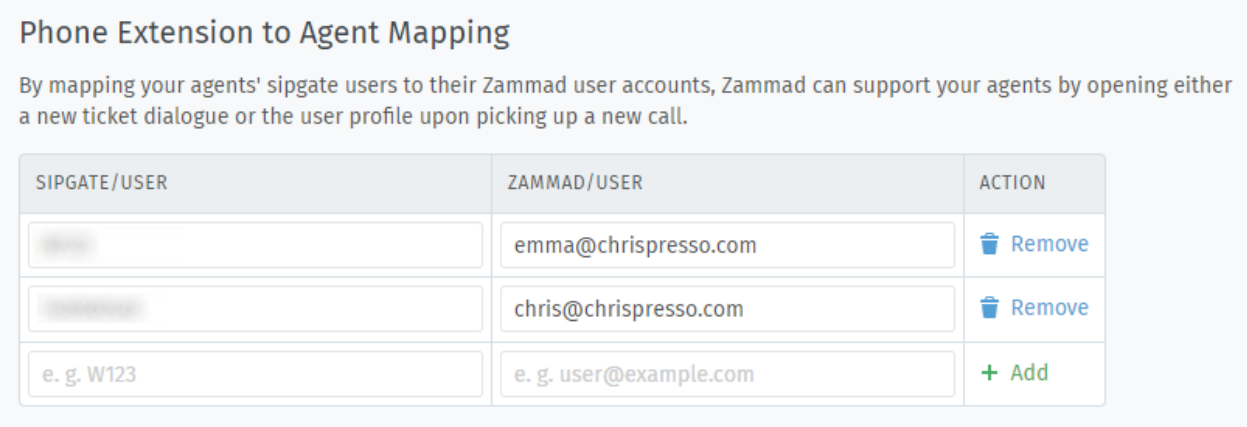
Warning: Potential performance issue

Setting this setting higher than 60 may cause serious performance issues on very busy instances. Keep in mind that this setting causes Zammad to poll and send up to 300 records to *all active agent sessions* in very short time periods.

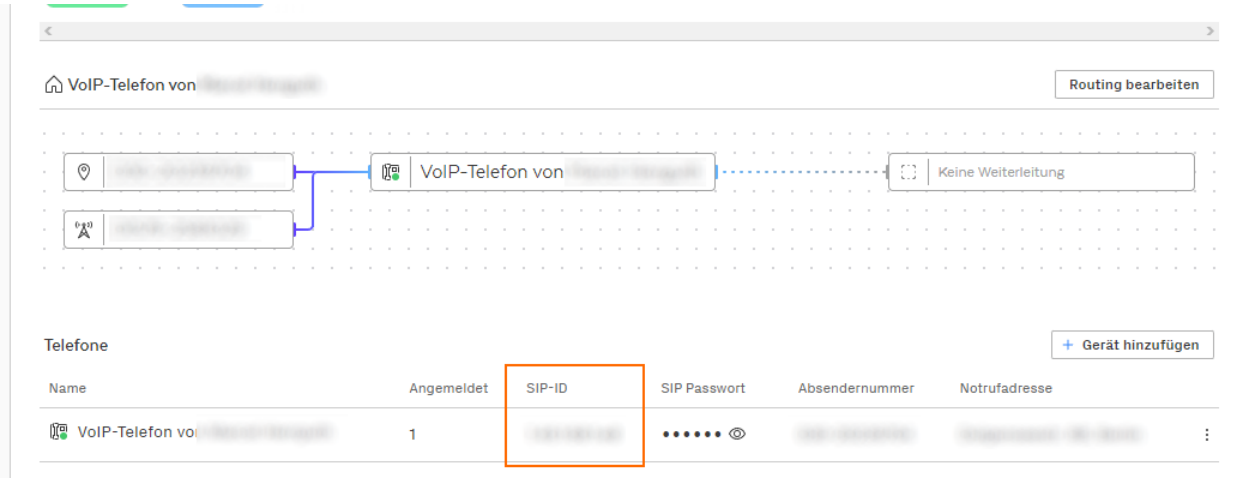
Phone Extension to Agent Mapping By mapping your agents extension to their existing Zammad users, Zammad can provide a new ticket dialogue or open the user profile for the agent that picks up the call.

This speeds up ticket aiding, no matter if it's for existing tickets or new ones!

Note: The agent perspective is described within our user documentation.



Hint: You can find your agent's Sipgate username within `Accountverwaltung` → `Benutzer`. You're looking for the SIP-ID.



Note: Users with several devices also have several SIP-IDs.

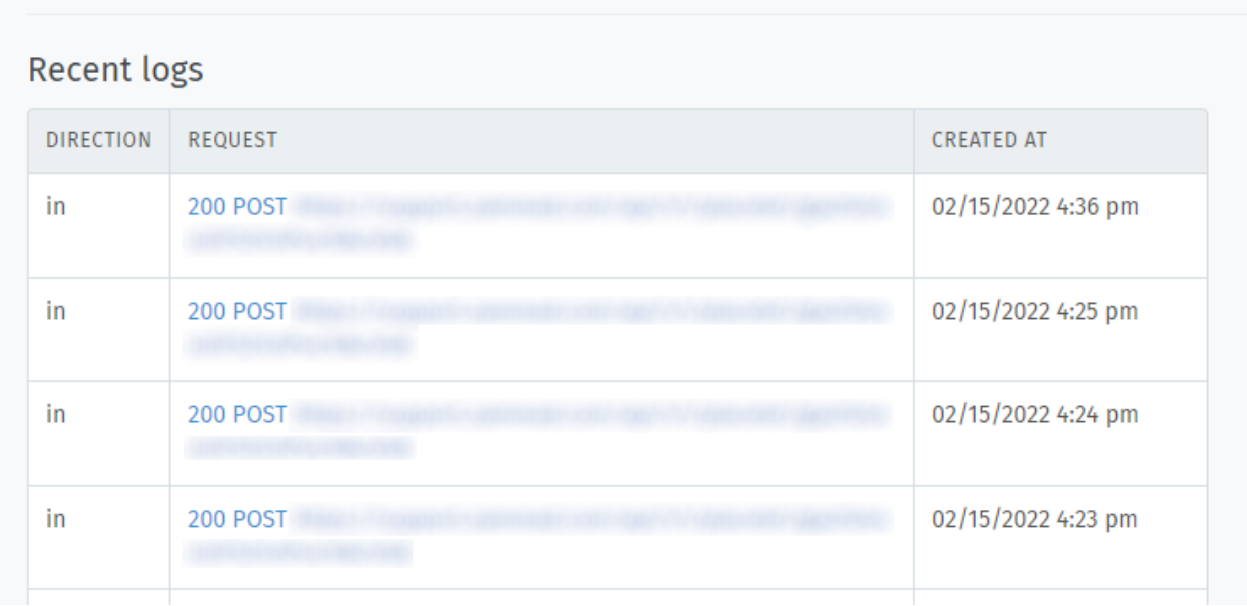
Recent Logs

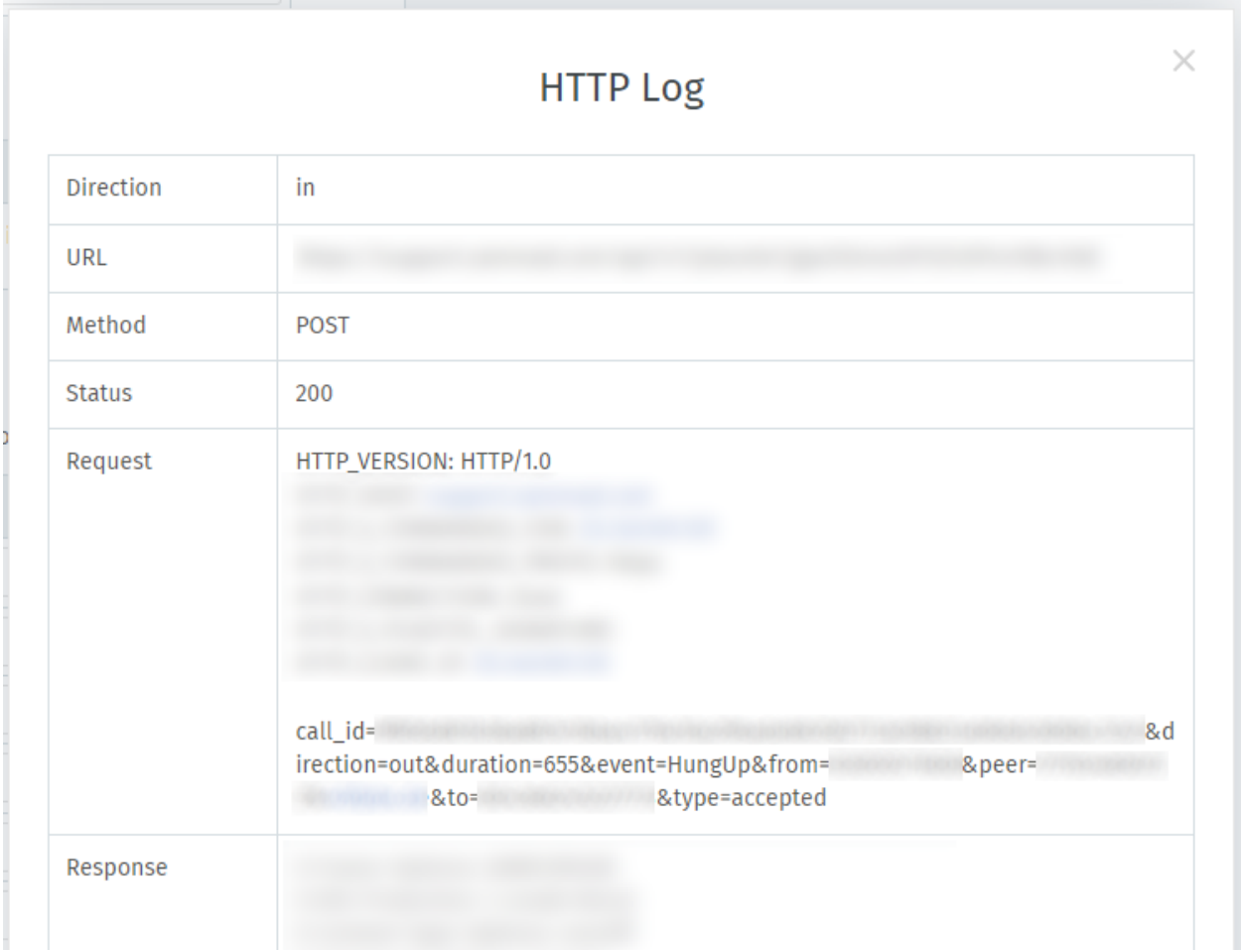
With recent logs Zammad allows you to view the latest calls for the CTI functionality. This usually comes in handy, when you're looking for errors.

I'm just here to clear floats up.

By clicking on the entry of interest, Zammad will provide more details on the call in question. You'll see the payload it received and also the response that was sent.

x


Recent logs
DIRECTION
REQUEST
CREATED AT
in
200 POST
02/15/2022 4:36 pm
in
200 POST
02/15/2022 4:25 pm
in
200 POST
02/15/2022 4:24 pm
in
200 POST
02/15/2022 4:23 pm


HTTP Log
Direction
in
URL
Method
POST
Status
200
Request
HTTP_VERSION: HTTP/1.0
irection=out&duration=655&event=HungUp&from=
&peer=
&to=
&type=accepted
Response

Troubleshooting

My Phone page stays blank, signalling does work… If you've made sure that signalling works (check Recent logs) and your Caller Log still stays empty, please ensure that you only configured one CTI integration version. Specifically defining more than one agent mapping on the different integration variants will be the issue.

Clear the not needed agent mapping and reload your browser page.

33.2 Integrations for authentication and customers

33.2.1 Clearbit

With our Clearbit integration, you can easily enrich the information provided by Zammad. If the customers or agents email address is known to Clearbit, it will share all information it has regarding the user with Zammad. Those information can include the following:

- Avatar
- Address information
- Website information
- A BIO (as Note by default)

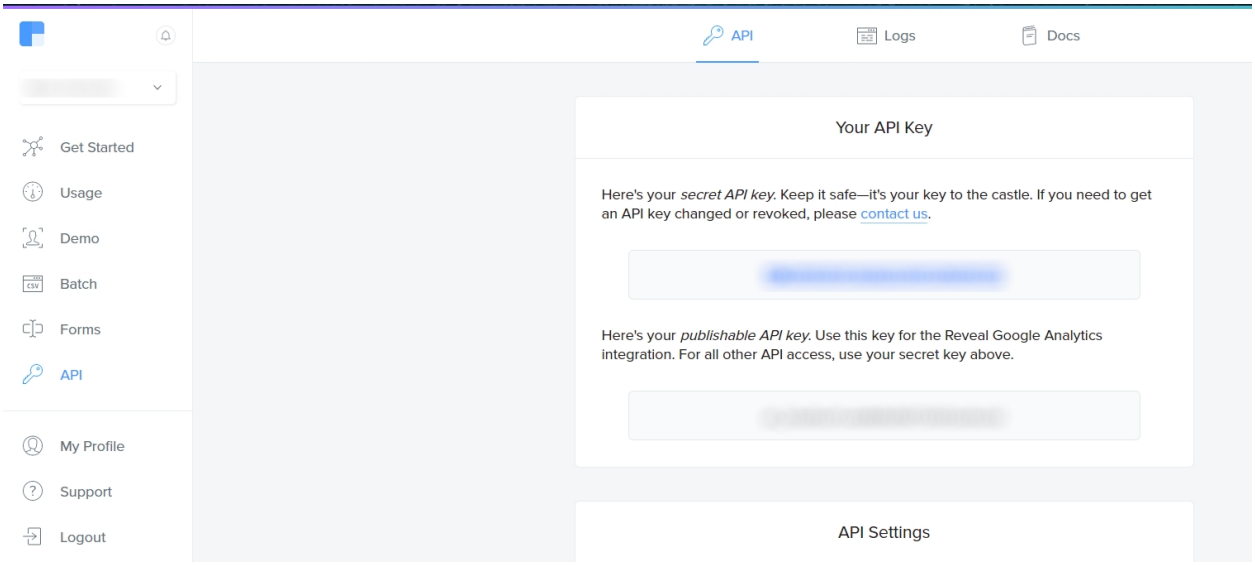
If needed, you can add further custom objects and add mappings to them, so the Clearbit information can be filled within the database. In general you can get any information from Clearbit, as long as you have a mapping to an Zammad object.

Hint: Clearbit does have a Mapping of fields like LDAP and Exchange have, but does not "win" against Zammad information. This means that if you have e.g. the last name field filled in already and Clearbit has other information on that, it will not be updated.

However: If let's say the last name has been set by Clearbit and Zammad notices that the last name on the Clearbit source has changed, Zammad will also update this value.

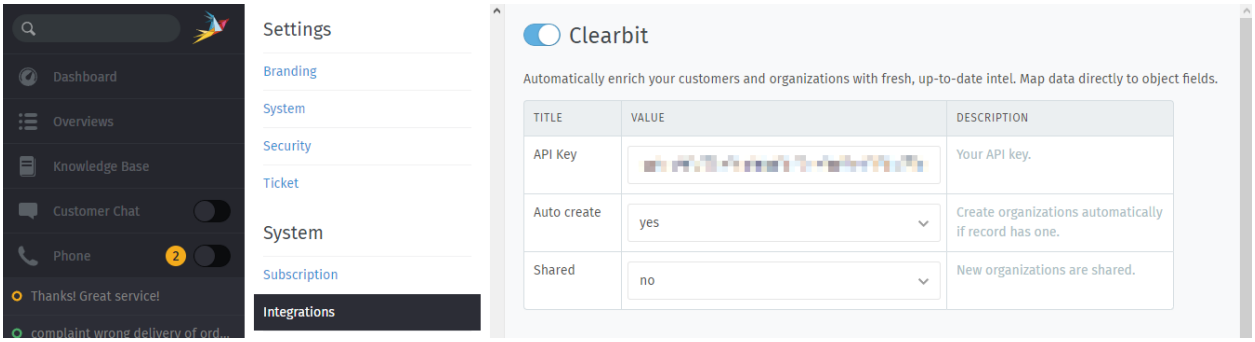
Configuration

The configuration of Clearbit is really easy and done fast! Just login to your Clearbit-Account, go to "API" and copy the secret-API-Key.



Now change to your Zammad instance, go to Integrations (System) -> Clearbit in the admin panel. Paste your API-Key into the API-Key-Field and decide if Zammad should create unknown Organizations automatically, if the user does not have one yet (and Clearbit knows it). The shared option decides if the new organizations Zammad creates with Clearbit should be shared ones.

Note: If you're unsure what option to choose, better stick with "no". You can also learn more about *Organizations*.

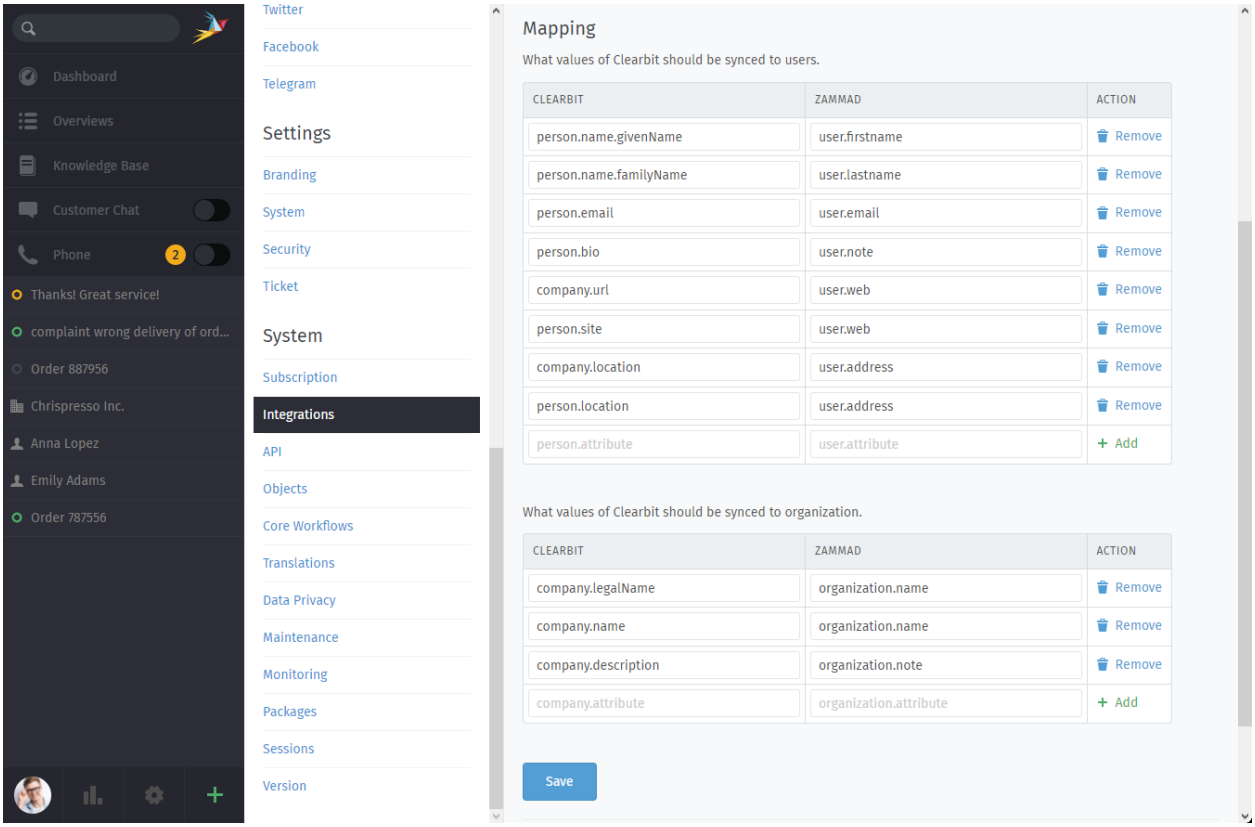


The Mapping option works similar to the mapping within the Exchange and LDAP sync. You can add further mappings for e.g. custom fields if you need more information that Clearbit can provide.

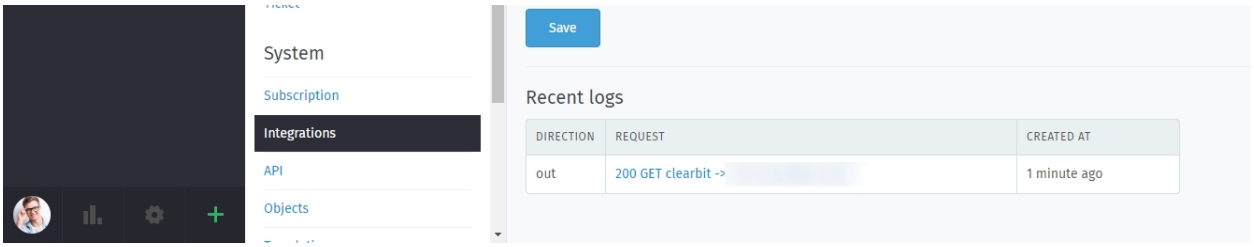
Note: If you want to add more Clearbit fields and want to learn more about available fields on their API, you can take a look at their API documentation .

If you're happy with the above chosen Settings and your mapping, just save the changes and enable Clearbit integration. Zammad will now start polling the Clearbit API as users contact you.

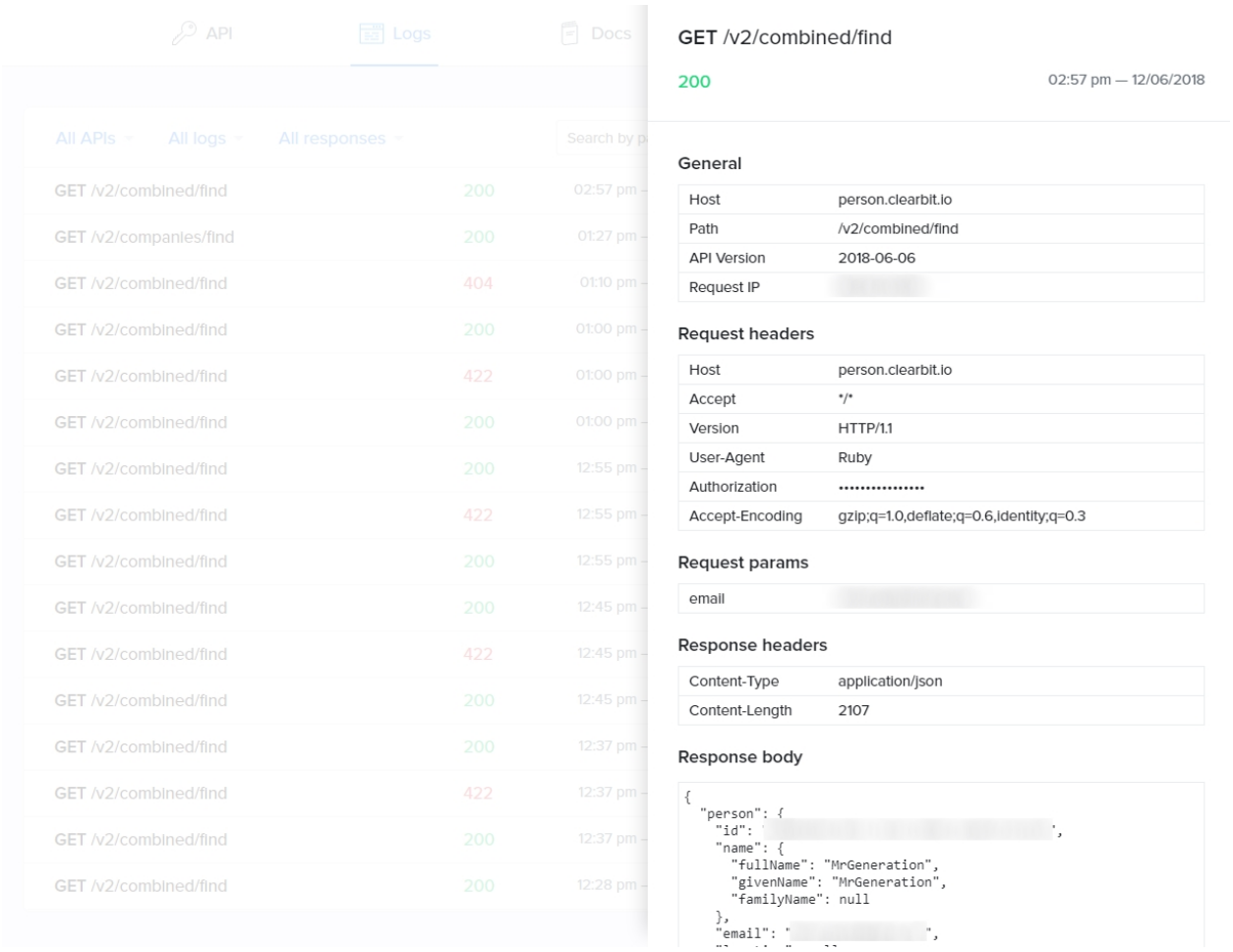
Note: Zammad does not synchronize with Clearbit on a regular basis, but on demand if needed. This saves API calls.



Below the Settings and Mappings, you'll find our Integration log. You can see what requests Zammad sent to Clearbit and also the APIs Response.



By the way, you can also view the API log on the Clearbit website - the information seen is basically the same.



Example when adding a user

To show you how fast information is added, we added a small Screencast below. This shows the creation of the User Alex from Clearbit. As we already set the last name of Alex to "X", Zammad did not update it. What Zammad did was to add further information it received from Clearbit. Really cool, right?

33.2.2 S/MIME

Prerequisites

- A certificate *and* private key for your own organization

  (Use this to **sign outgoing messages** and **decrypt incoming messages**.)

- Certificates belonging your contacts, or their issuing certificate authority (CA)

  (Use these to **verify incoming message signatures** and **encrypt outgoing messages**.)

Note: I'm new to S/MIME. Where can I get a certificate?

The easiest way to get certificates is to buy an annual subscription through a commercial CA, such as:

- Sectigo (formerly Comodo)
- Secorio
- GlobalSign

(Zammad is not affiliated with these CAs in any way.)

You can also generate your own self-signed certificates, but the process is complicated and usually **involves extra work for your contacts**.

Bear in mind that **S/MIME only works if the other party is using it, too.**

Limitations

Please note that Zammad will distrust senders by default. This means that you're always required to provide certificate data, no matter if for signing or encrypting.

This is by design and can't be adjusted.

Manage Certificates

S/MIME is **disabled by default**. Enable it to start adding certificates.

Fig. 1: Manage certificates in the Admin Panel under **System > Integrations > S/MIME**. Certificates may be pasted in as plain text or uploaded from a file.

Add Certificate data

Add Certificate Import public-key certificates for both your own organization and your contacts.

You can also add a bunch of certificates in one go by providing a single file with all relevant certificates.

> **Warning: ALWAYS verify certificates in-person or over the phone!**
>
> The whole point of signatures is to alert you when someone is trying to pretend to be someone they're not. Never accept a certificate from someone online without verifying it first.

Note: What about trusted certificate authorities?

In some cases (*e.g.,* when dealing with large enterprises), you may be given a certificate for an entire CA, rather than a single contact. Add it here to trust *all* certificates issued by that CA.

Commercial CAs can usually be verified online. Zammad does not include a list of built-in, trusted CAs.

Add Private Key Once you've added a public-key certificate, you can import its matching private key.

Private keys are for **your own organization only**; never ask your contacts for their private keys.

Note: Certificates and private keys must be uploaded separately.

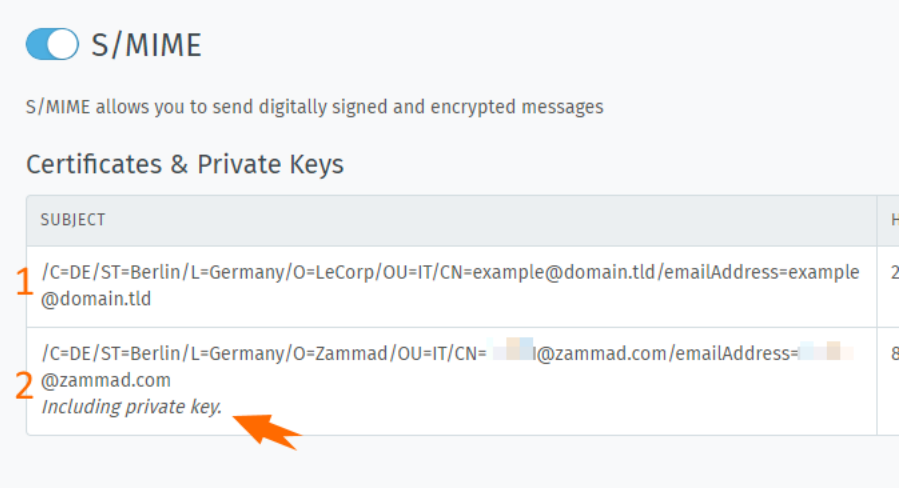


Fig. 2: A note is displayed on certificates with a matching private key (see line 2).

If your certificate and private key are bundled together in the same file or PEM block, import it twice (once using each button).

Please note that bulk imports of private keys are not possible.

Download Certificate data

You can download the earlier provided certificates and private keys at any time from your Zammad instance.

Note: Passphrase-protected private keys stay protected

Downloading private keys that originally were encrypted with a passphrase will also have this state after retrieval. Knowing the password is mandatory to continue working with keys in question.

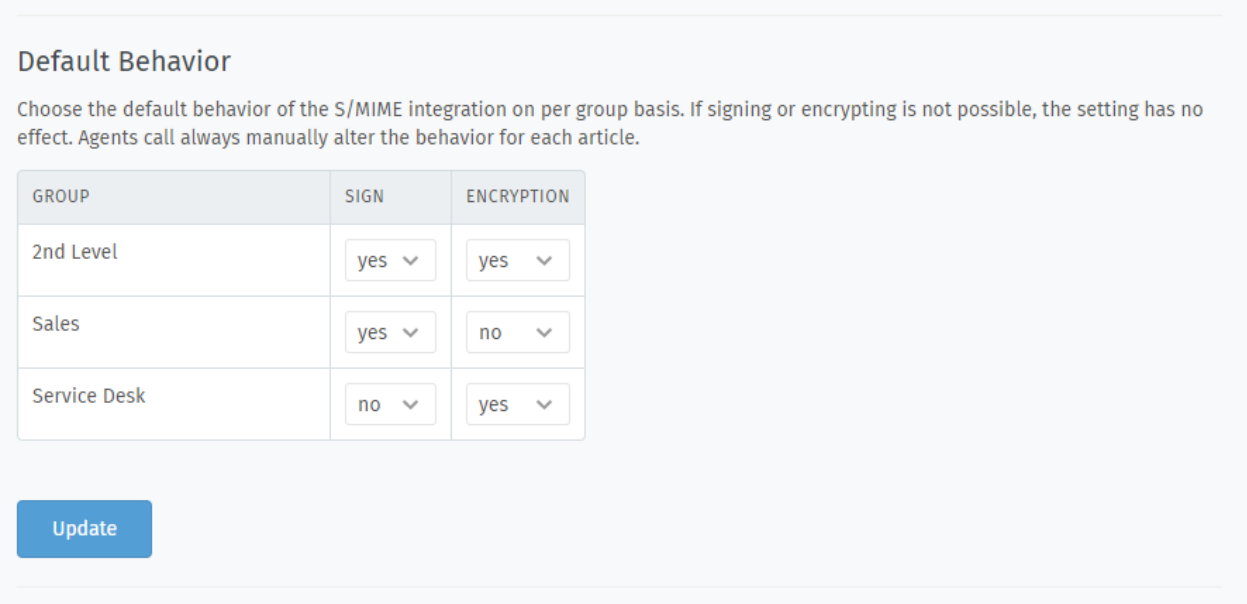
Fig. 3: Download stored certificates and their keys

Default Behavior

The ticket composer will set all outgoing messages to **signed and encrypted** by default (assuming the required certificates exist).

These defaults can be modified on a per-group basis:

Of course, agents can always manually change these settings on each email they send out.



Troubleshooting

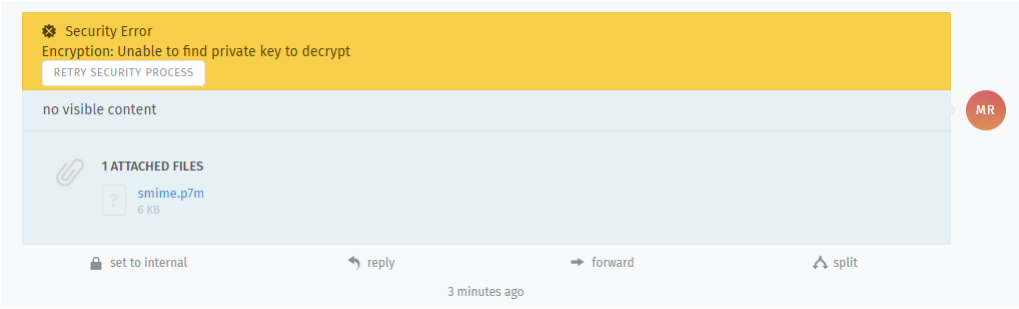
All of the system's latest S/MIME activity is displayed in the **Recent Logs** section.

Fig. 4: Logs report the status and details of all mail, both incoming and outgoing, that used signing/verification or encryption/decryption.

Note: This log does **not** include email sent by *triggers* or the *scheduler*. For those, check your `production.log`.

Common Issues

I received a signed/encrypted email before I set up S/MIME integration No problem. Once S/MIME has been enabled and the appropriate certificates have been added, agents will be prompted to retry verification/decryption on matching emails.



The Encrypt button is disabled

- Have you added the recipient's certificate?
- Are you sure the recipient's certificate is valid?

- Have you checked your `production.log` for more details?

> **Warning:** If encryption doesn't work in the composer, it won't work in *triggers* or the *scheduler* either!

The Sign button is disabled

- Have you added **both the certificate and private key** for your organization?
- Does the email address on the certificate match the email address of the agent/group composing the email?

Error: "Fingerprint already taken"

- Are you sure you haven't added this certificate already?

Error: " invalid byte sequence in UTF-8"

- Please ensure to provide PEM formatted certificate and keys.
- Did you check if the provided file is a valid certificate or key?

S/MIME is the most widely-supported method for secure email communication. With S/MIME, you can exchange **signed** and **encrypted** messages with others.

Signing is proof that a message hasn't been tampered with or sent by an impersonator.

In other words, it guarantees a message's **integrity** and **authenticity**.

Encryption scrambles a message so that it can only be unscrambled by the intended recipient.

In other words, it guarantees **privacy** and **data security**.

Fig. 5: Once S/MIME has been enabled, **Encrypt** and **Sign** buttons will appear in the ticket composer.

Prerequisites New to S/MIME? This section helps you to find certificate providers and points out Limitations.

Manage Certificates Add and download certificates; configure group default behaviors.

Troubleshooting Took a wrong turn? About recent logs and common issues.

Usage

For more details on how S/MIME integration works in practice, see the user docs.

33.3 Integrations for Monitoring Systems

Note: This section currently misses the following integrations:

- Icinga
- Monit
- Nagios

33.3.1 Checkmk

Checkmk is a powerful IT monitoring tool that can send real-time status alerts to Zammad via email or REST API. Set these alerts up in Checkmk, and Zammad will automatically create, update, and close tickets based on the health of your system:

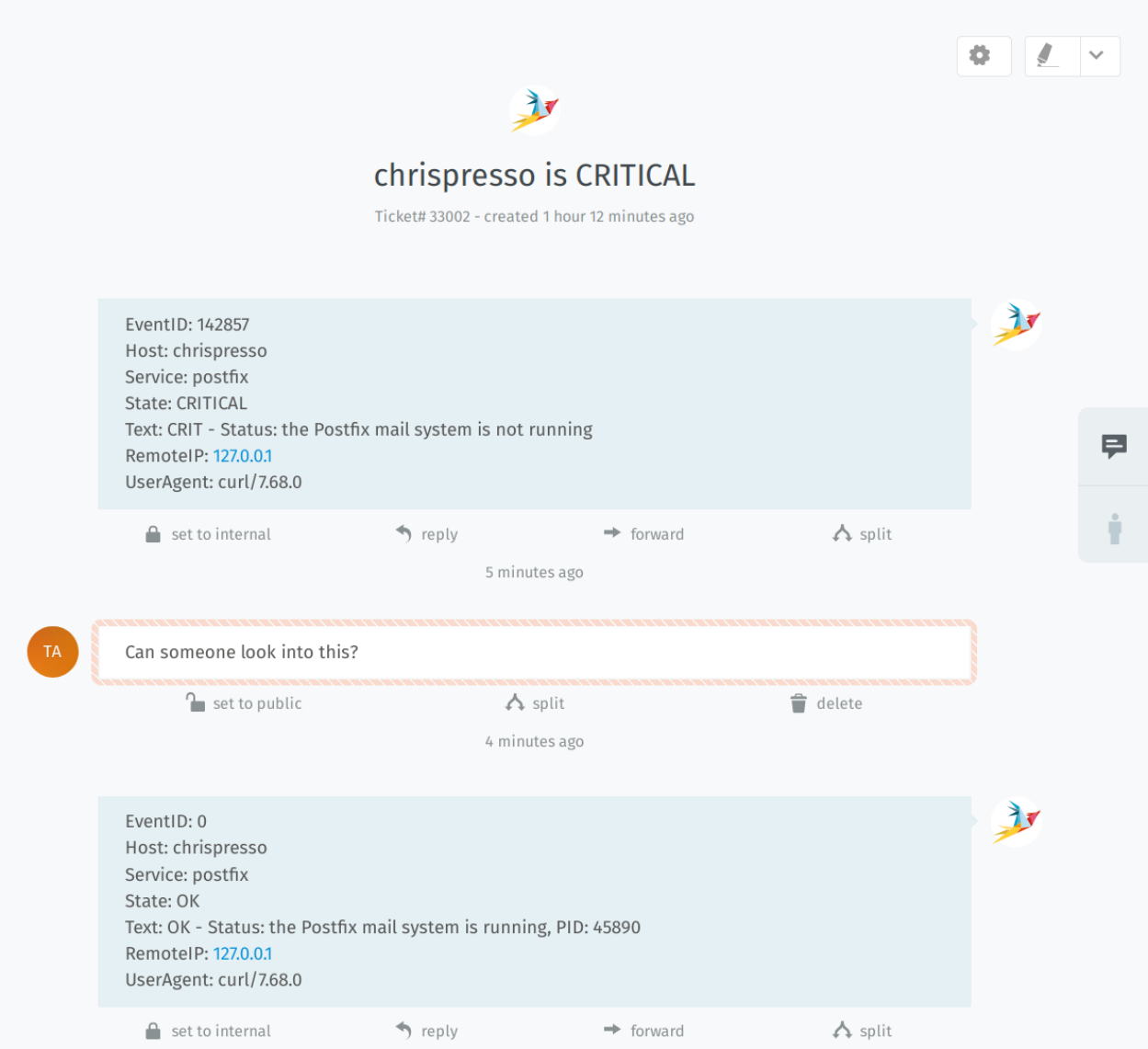


Fig. 6: Multiple alerts, same ticket.

Setup Guide

Whenever the Checkmk integration is enabled, Zammad listens for messages on its API and over email. As long as those messages follow the required format, Zammad will create new tickets (or find and update existing ones) based on the message contents.

That means that "setting up Checkmk integration" is simply a matter of **adding configuration to your Checkmk site**: telling it when to send these messages and what to say. To do that, you'll need to create a new **notification rule** (see Checkmk's official docs for help with that).

As part of this new rule, you'll have to choose a **notification method** (*i.e.,* a script to execute whenever the rule is triggered). This script will be written by you (samples below), and contain the logic for sending API/email messages to Zammad:

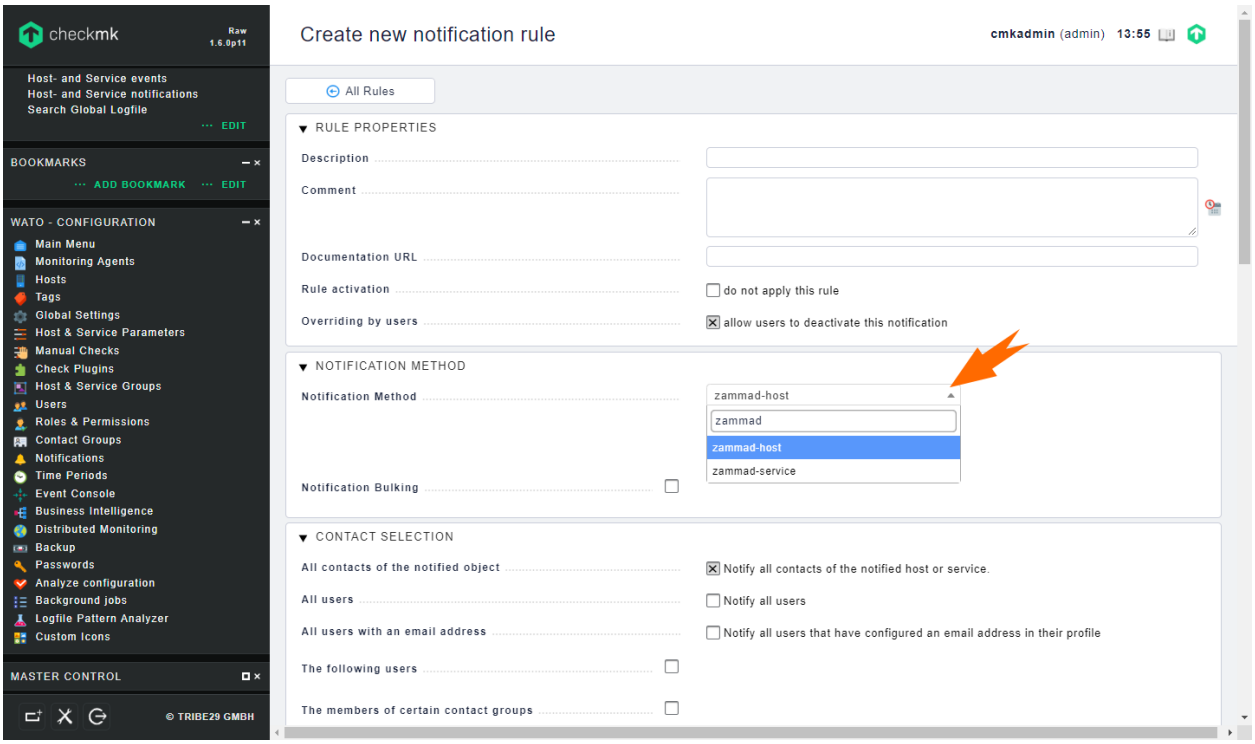


Fig. 7: Find your custom scripts in the Checkmk WATO under **Notifications > New Rule > Notification Method**.

Once you're done setting up your new rule, you're all set! New tickets should start coming in and auto-updating any time your rule is triggered.

(If you need help troubleshooting, be sure to check the *Recent Logs*.)

API Alerts

To add these scripts in the Checkmk WATO, copy them into your Checkmk installation directory and make them executable. (Be sure to replace the `zammad.example.com` callback URL with the one found in your admin panel.)

Service notification For updates on the status of the software running on your server (*e.g.,* postgres).

```
#!/bin/bash
# /opt/omd/sites/<SITE>/local/share/check_mk/notifications/zammad-service
```

(continues on next page)

(continued from previous page)

```
curl -X POST \
  -F "event_id=$NOTIFY_SERVICEPROBLEMID" \
  -F "host=$NOTIFY_HOSTNAME" \
  -F "service=$NOTIFY_SERVICEDESC" \
  -F "state=$NOTIFY_SERVICESTATE" \
  -F "text=$NOTIFY_SERVICEOUTPUT" \
  https://zammad.example.com/api/v1/... # see Admin Panel > System > Integrations >␣
↪Checkmk > Usage
```

Host notification For updates on the status of the server itself.

```
#!/bin/bash
# /opt/omd/sites/<SITE>/local/share/check_mk/notifications/zammad-host

curl -X POST \
  -F "event_id=$NOTIFY_HOSTPROBLEMID" \
  -F "host=$NOTIFY_HOSTNAME" \
  -F "state=$NOTIFY_HOSTSTATE" \
  -F "text=$NOTIFY_HOSTOUTPUT" \
  https://zammad.example.com/api/v1/... # see Admin Panel > System > Integrations >␣
↪Checkmk > Usage
```

Note: What's with all the env vars?

Whenever Checkmk runs these scripts, it needs to provide some information about the event that triggered the notification. This information is passed in the form of these `$NOTIFY_*` environment variables.

You can specify additional parameters to pass to the script when you're setting up your notification rule, but the ones you see here are all provided by default.

Email Alerts

Warning: There are known bugs in Zammad's processing of email from Checkmk. This section will be completed once they have been resolved. (Learn more at GitHub issues #2377 & #2180.)

In the meantime, we encourage you to set up API alerts instead.

API Reference

For most users, *the sample scripts from the Setup Guide* will do the job just fine. But if you want more fine-grained control—for instance, to create high- and low-priority tickets for different types of system events—then you'll have to customize the data those scripts send to Zammad.

Example

This custom script will automatically set all tickets it creates to **high priority** and **assign them to charlie@chrispresso.com**.

```
#!/bin/bash

curl -X POST \
  -F "event_id=$NOTIFY_HOSTPROBLEMID" \
  -F "host=$NOTIFY_HOSTNAME" \
  -F "state=$NOTIFY_HOSTSTATE" \
  -F "text=$NOTIFY_HOSTOUTPUT" \
  -F "priority=3 high" \
  -F "owner=charlie@chrispresso.com" \
  https://zammad.example.com/api/v1/...
```

How does it work?

There are two kinds of data you can pass to the API, both in the form of key-value pairs:

Checkmk parameters are required, and make up the **contents** of the resulting tickets/articles. They also determine whether an event creates a new ticket or updates/closes an existing one.

These are the only values used in the sample scripts. **Use them as-is**; technically, they can be customized, but it's hard to imagine a good reason for it.

Ticket attributes are optional, and can be used to **adjust settings** on newly created tickets (*e.g.,* set the owner, group, priority, or state).

If you want to customize your Checkmk alert script, do it with these. Simply add an extra "form" option for each one (`-F "key=value"`) to your script's `curl` command line, as in the example above.

Hint: It's just an API endpoint!

When using Checkmk integration, messages need to be formatted in a certain way, but that *doesn't mean the messages actually have to come from Checkmk*.

If you use another monitoring tool that's not officially supported by Zammad, there's probably a way to make it work with your Checkmk callback URL.

Checkmk Parameters

When a notification is received, Zammad creates a new article containing the details of the event that triggered it:

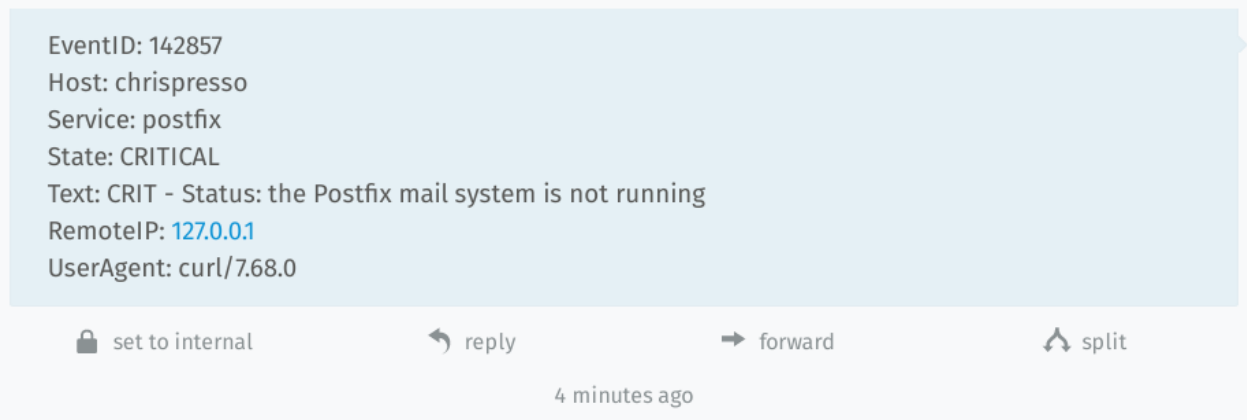
These details come from the fields listed below, which correspond to parameters provided by Checkmk (`$NOTIFY_*`).

Required fields are marked with an asterisk (*).

event_id* A unique ID for the system event. (`$NOTIFY_SERVICEPROBLEMID` / `$NOTIFY_HOSTPROBLEMID`)

host* The hostname of the system that the event originated from. (`$NOTIFY_HOSTNAME`)

Used to determine if a new event belongs to an existing ticket. Also used in the subject line of the resulting article ("<host> is <state>").



service The name of the service that the event originated from. (`$NOTIFY_SERVICEDESC`)

Used to determine if a new event belongs to an existing ticket.

Displayed as - when omitted.

state* The current state of the service or host in question. (`$NOTIFY_SERVICESTATE` / `$NOTIFY_HOSTSTATE`)

Used to detect when a ticket should be auto-closed (*i.e.,* on `OK`/`UP`). Also used in the subject line of the resulting article ("<host> is <state>").

text The output of the process that triggered the event. (`$NOTIFY_SERVICEOUTPUT` / `$NOTIFY_HOSTOUTPUT`)

Displayed as - when omitted.

Ticket Attributes

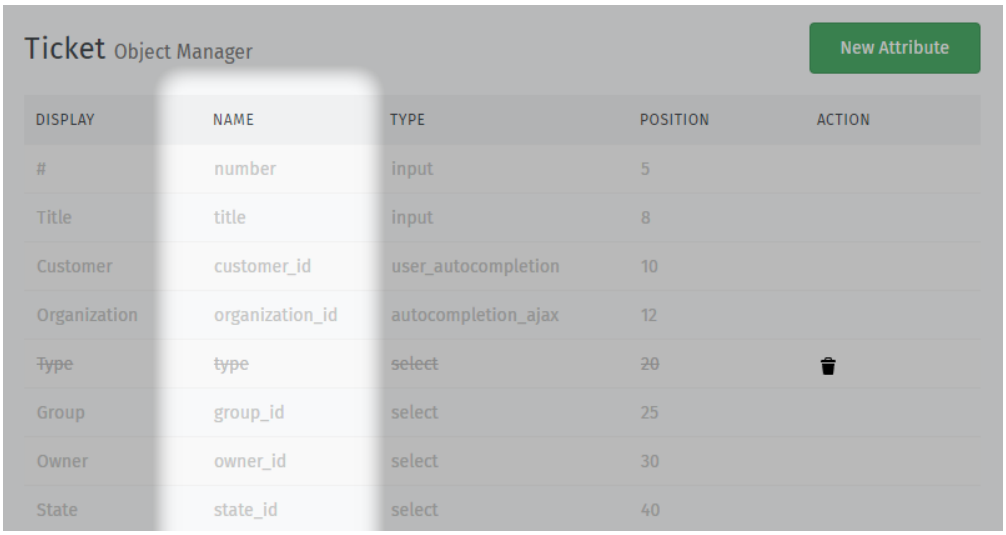


Fig. 8: Find a complete list of ticket attributes in the Object Manager.

Ticket attributes are entirely optional, and can be used to customize the tickets that Checkmk creates. (Note that these attributes will be ignored if a new event belongs to an existing ticket.)

Why would you want to do this? Maybe you have only one IT guy, and all system monitoring issues should be automatically assigned to him. Or, maybe you're creating multiple notification rules so that database outages take higher

priority than disk space warnings.

In most cases, you'll probably want to set one of the following:

- group
- owner
- state
- priority

but in practice, you can set almost any attribute, including *custom ones you created through the Object Manager*.

Note: The following attributes are **not customizable**:

- title
- id
- ticket number
- customer
- created_by_id
- updated_by_id

How do I know what values I can set?

> **Warning: Invalid values → unpredictable behavior**
>
> If you provide a value that Zammad doesn't understand (*e.g.,* `-F "priority=high"`), it's not always clear what will happen. In some cases, a ticket will be created with the default values instead—but in others, it may not be created at all!

So what values does Zammad understand, then? Well, it depends…

owner Use an email address or username:

```
-F "owner=it@chrispresso.com"
```

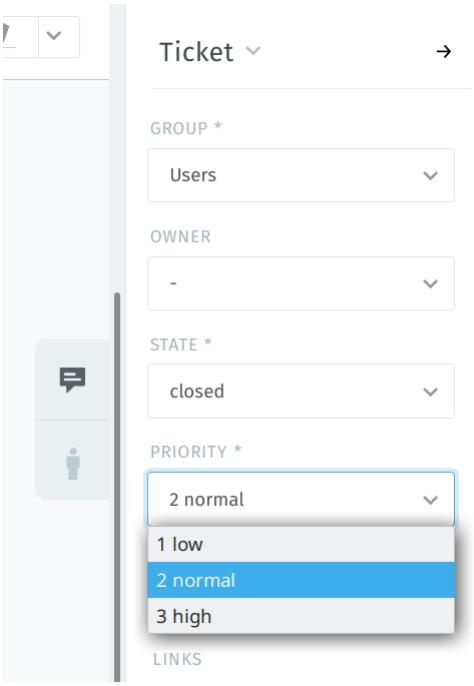
group & priority Refer to the dropdown menus in the ticket pane:

```
-F "group=Users"
-F "priority=3 high"
```

Note: Ticket state CANNOT be set this way!

Why? Because `-F "state=..."` is already used *as a Checkmk parameter*.

Everything Else To set any other attributes, it helps to know your way around the rails console. Valid values are those that you can set with a string:



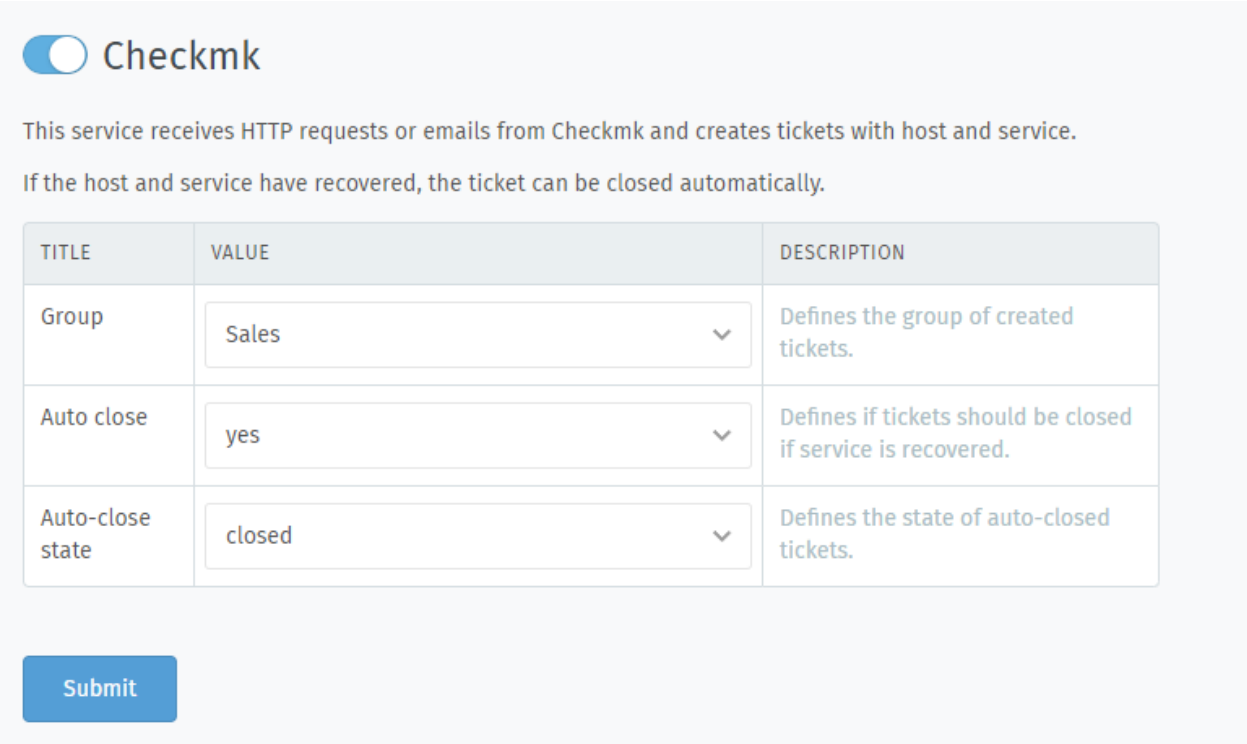

```
# valid
>> Ticket.first.update(note: "You're gonna need a bigger boat")
=> true
>> Ticket.first.note
=> "You're gonna need a bigger boat"

>> Ticket::State.find_by(name: "open").id
=> 2
>> Ticket.first.update(state_id: 2)
=> true
>> Ticket.first.state.name
=> "open"

# invalid
>> Ticket.first.update(preferences: "I'm a Checkmk ticket!")
=> true
>> Ticket.first.preferences
=> {}
```

These values can then be passed directly to the API:

```
-F "note=You're gonna need a bigger boat"
-F "state_id=2"
```

Admin Panel Reference

Settings



Group Which *group* should Checkmk tickets be assigned to as a default (*i.e.,* when none is specified)?

(Applies to *API alerts* only.)

Auto close Should Zammad automatically close tickets if a service has recovered on its own?

(Agents will receive notifications for such closures as appropriate.)

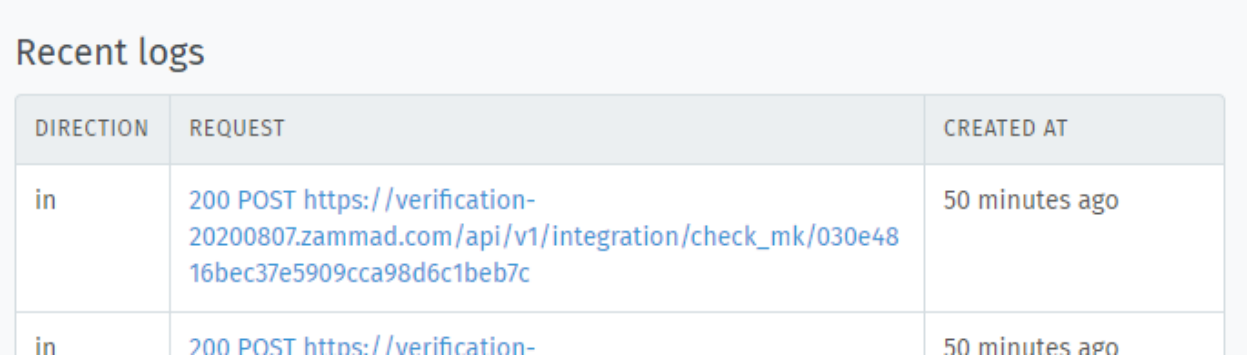
Auto-close state What ticket state should be applied when "auto-closing" a ticket?

(You may choose from the seven built-in ticket states, but if you change this setting from the default, you'll more likely want to define a new ticket state for this purpose. This can be especially useful for tracking tickets with *reports*.)

Recent Logs

If you're having trouble getting Zammad and Checkmk to play nicely together, this section can help you troubleshoot.

It contains a record of the fifty most recent transactions that Zammad knows about, including each one's request/response details and return status.



33.3.2 Zabbix Integration

This guide describes how to integrate your Zabbix 5.4 installation with Zammad using the Zabbix webhook feature. This guide will provide instructions on setting up a media type, a user and an action in Zabbix.

Requirements

- Zammad with enabled HTTP Token Authentication
- Zabbix version 5.4 or higher

Setting up a Zammad

1. Enable **API Token Access** in Settings > System > API.

2. Create a **new user** for a Zabbix alerter with an **email address** and create a personal user token with **ticket.agent** permissions.

Zabbix Webhook configuration

Create a global macro

1. Before setting up the **Webhook**, you need to setup the global macro **{$ZABBIX.URL}**, which must contain the **URL** to the **Zabbix frontend**.
2. In the **Administration** > **Media types** section, import the Template.
3. Open the added **Zammad** media type and set:
   - **zammad_access_token** to the your **Personal User Token**
   - **zammad_url** to the **frontend URL** of your **Zammad** installation
   - **zammad_customer** to your **Zammad user email**.
   - **zammad_enable_tags** to **true** or **false** to enable or disable trigger tags. **Important**: if you enable tag support, each tag is set with a separate request.
4. If you want to prioritize issues according to **severity** values in Zabbix, you can define mapping parameters:
   - **severity_<name>**: Zammad priority ID
6. Click the **Update** button to save the **Webhook** settings.

7. To receive notifications in **Zammad**, you need to create a **Zabbix user** and add **Media** with the **Zammad** type.

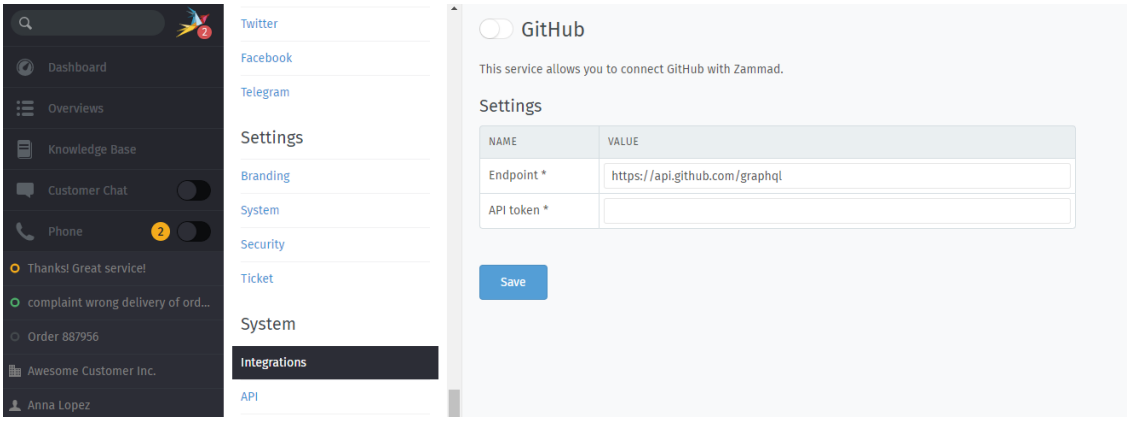
   For **Send to**: enter any text, as this value is not used, but is required.

For more information, use the Zabbix documentation.

33.4 Integrations for Issue Trackers

33.4.1 GitHub

Use GitHub integration to track GitHub issues directly within Zammad tickets. Add issue hyperlinks and get a live summary of metadata like status (open/closed), assignee, labels, and more.



Note: GitHub integration does not support pull requests.

Setup

1. In your GitHub settings, create a new API token under **Developer settings > Personal access tokens > Generate new token**. Leave the **Scopes** section empty.

Fig. 9: Create a new API key with no scopes/privileges.

Hint: Will this work for private repos?

No. To link private repo issues, **use the `repo` scope instead**. Bear in mind that the resulting token will have lots of permissions that it doesn't actually need, which presents a security risk if your token ever falls into the wrong hands.

Unfortunately, because of how GitHub's OAuth token scopes are set up, this is the only way to link issues on private repos.

2. Enter your new API token in Zammad and enable GitHub integration.

Hint: Leave the default API endpoint (`https://api.github.com/graphql`) as-is unless you're using GitHub Enterprise Server.

Once completed, a new GitHub issues tab will appear in the ticket pane.

Troubleshooting

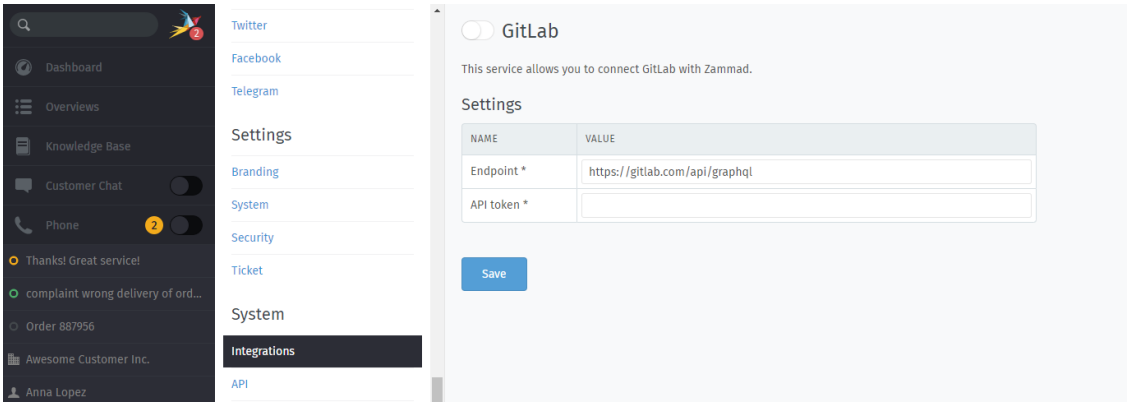
Token verification is taking a long time Slight delays are normal (<2 min.), especially for systems under heavy load.

Self-hosted administrators, please check your network settings to ensure that your Zammad server can reach `api.github.com`.

I reloaded the page and now the API token is gone This may indicate that Zammad is still verifying your API token. Try reloading the page again in a couple minutes.

33.4.2 GitLab

Use GitLab integration to track GitLab issues directly within Zammad tickets. Add issue hyperlinks and get a live summary of metadata like status (open/closed), assignee, labels, and more.



Note: GitLab integration does not support merge requests.

Setup

1. In your GitLab preferences, create a new API token under **Access Tokens**.

   Under **Select scopes**, choose `read_api` only.

   Hint: If you wish to link issues on any private repos…

   Your API token must belong to an account with access to those repos.

2. Enter your new API token in Zammad and enable GitLab integration.

Hint: Leave the default API endpoint (`https://gitlab.com/api/graphql`) as-is unless you're a self-hosted GitLab user.

Once completed, a new GitLab issues tab will appear in the ticket pane.

Troubleshooting

Token verification is taking a long time Slight delays are normal (<2 min.), especially for systems under heavy load.

Self-hosted administrators, please check your network settings to ensure that your Zammad server can reach `gitlab.com`.

I reloaded the page and now the API token is gone This may indicate that Zammad is still verifying your API token. Try reloading the page again in a couple minutes.

33.5 Other Integrations

This section will hold any other integration that can't be grouped up (yet).

33.5.1 Slack

Note: In order to use this feature, please add a new Slack app to your Workspace. The App you need is called `Incoming WebHooks` .

Why do I need this feature?

If you're already using Slack for your team communication, you'll love this feature! Our Slack integration can push ticket notifications about the last ticket article based on the following events:

- on Ticket creation
- on Ticket updates
- on reached reminders
- a Ticket has escalated
- a Ticket is going to escalate

Zammad will provide the Ticket title, a direct link to the Ticket, the event type (creation, updated, escalation), the customer, time and the last article that has triggered the notification.

This will give you further options, as you can see e.g. escalating tickets that are assigned to an agent that's e.g. absent. You can interact faster, as you might see problems earlier (or even before the problem gets one).

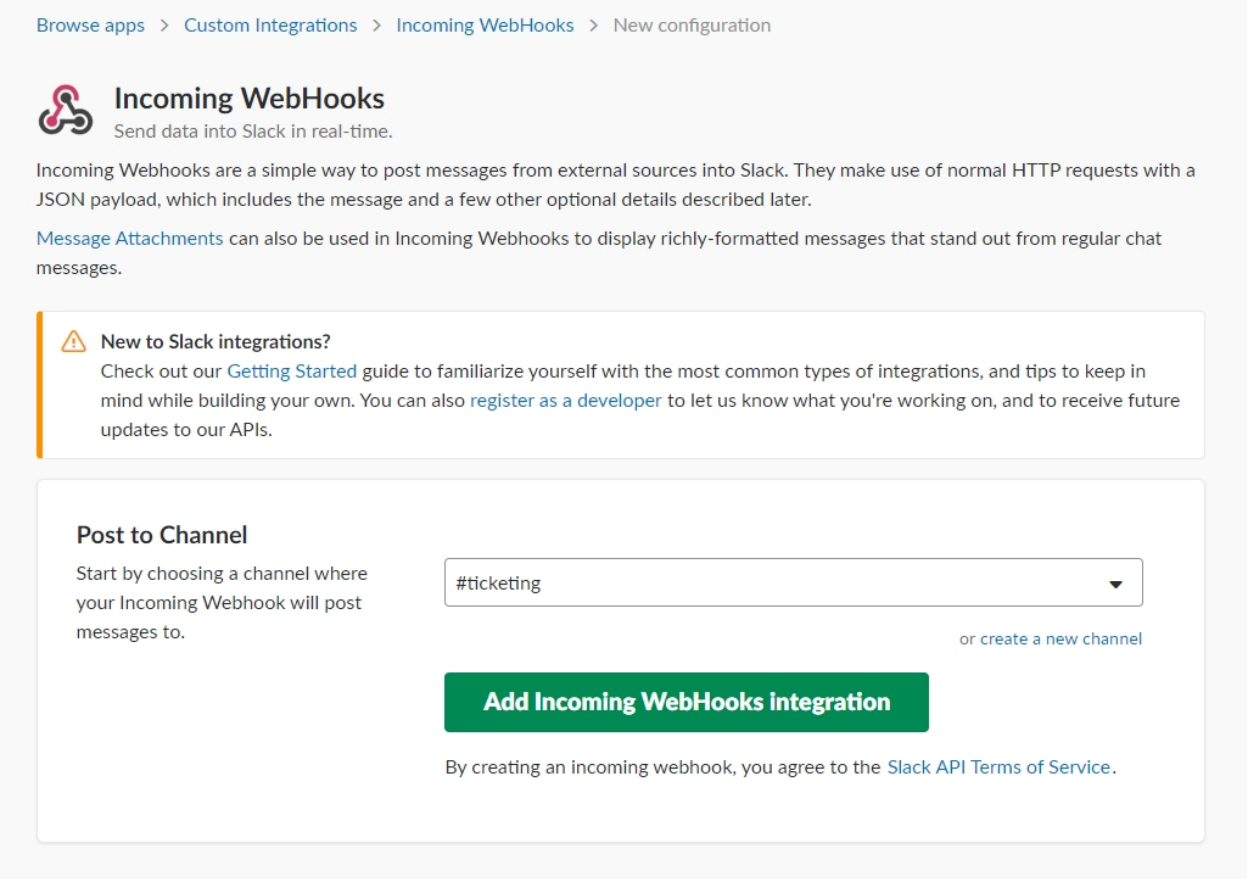
If needed and wanted, you can even discuss directly about the topic before sending an answer to the customer. Another possible use case would be a agent monitoring new agents and their answers to tickets.

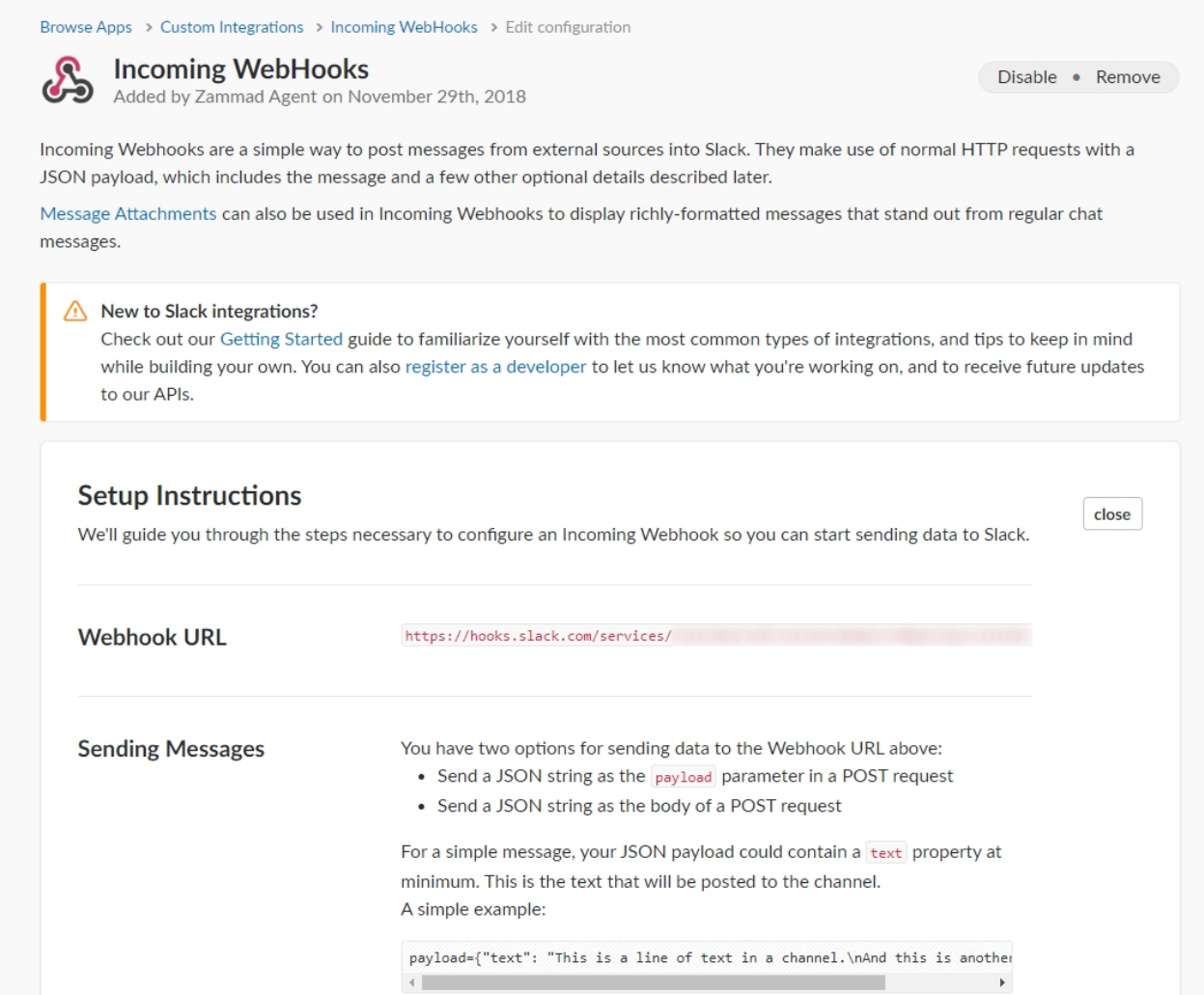
Configure the integration

First of all, please go to your slack workspace - go to administration => Manage Apps. If you don't have an app yet, you can simply add a new one - just search for `` Incoming WebHooks`` and customize the app to your needs.

Choose (or create) the channel Zamma should post it's information to and press on "Add Incoming WebHooks integration". If you're ready, copy the provided WebHook URL and go to your Zammad installation.

Hint: You need administrative rights on the Slack Workspace. The link to the app directory is normally `https://[workspace-name].slack.com/apps` .



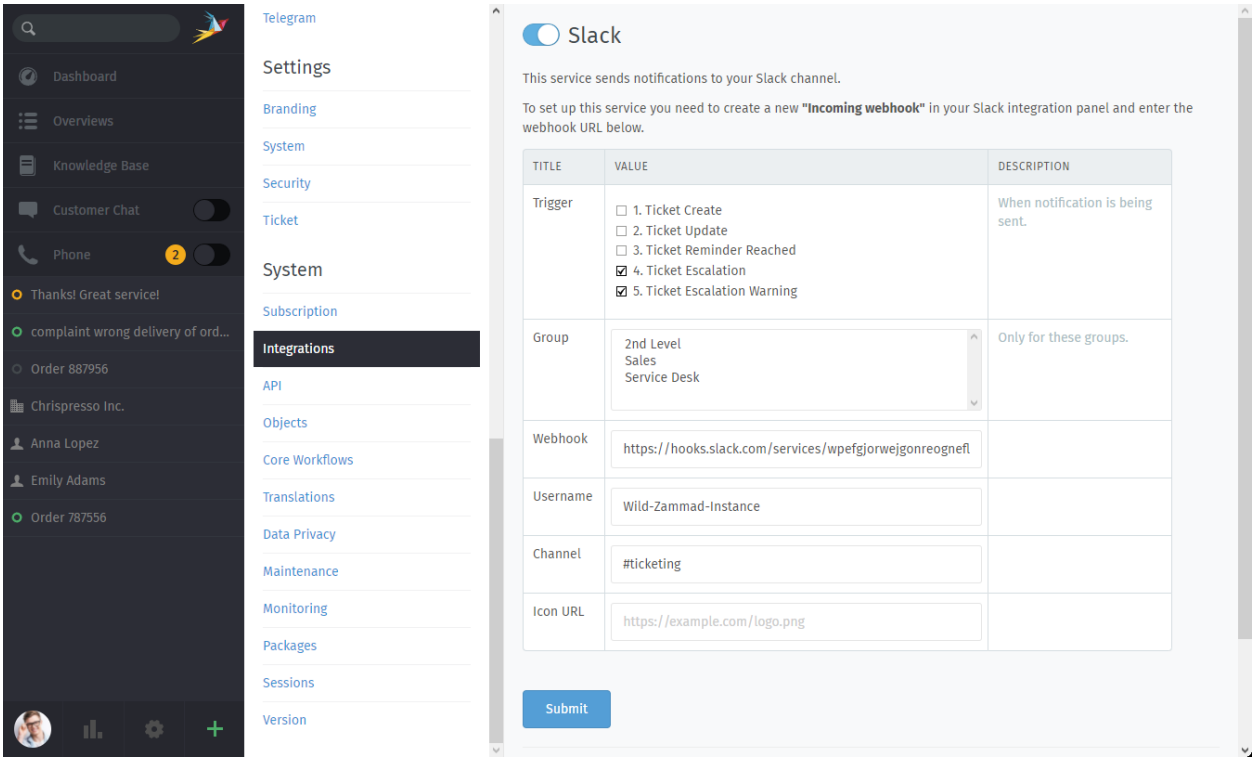


To configure the slack integration, log in to Zammad and go to Integrations (System) => Slack in the admin panel.

Here you can choose on what evens Zammad should post information about a ticket to your Slack channel. Next you need to device what groups shall be affected by this, as anybody with access to that specific Slack channel can read at least parts of the ticket this might be a privacy issue, if you select the wrong groups. The username is simply the name that Zammad uses as display name inside the Slack chat. The channel defines the Slack channel the information is being posted in. As last option, you can set a custom icon for posting to slack.

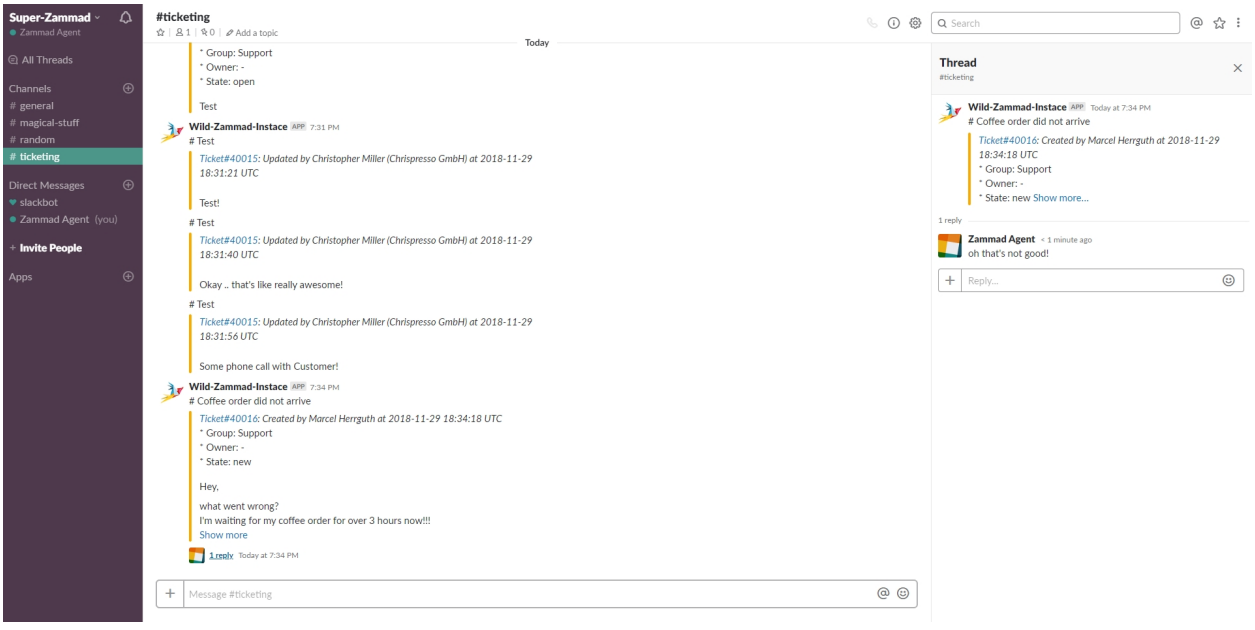
When you're ready, just hit "Submit" and enable the integration. Zammad will now post new ticket information based on the trigger you chose. Below the options you have the recent log that shows the latest requests to Slack for debugging if needed.

Note: If you leave the Icon URL empty, Zammad will use the Zammad logo instead. The icon should be a square PNG file.

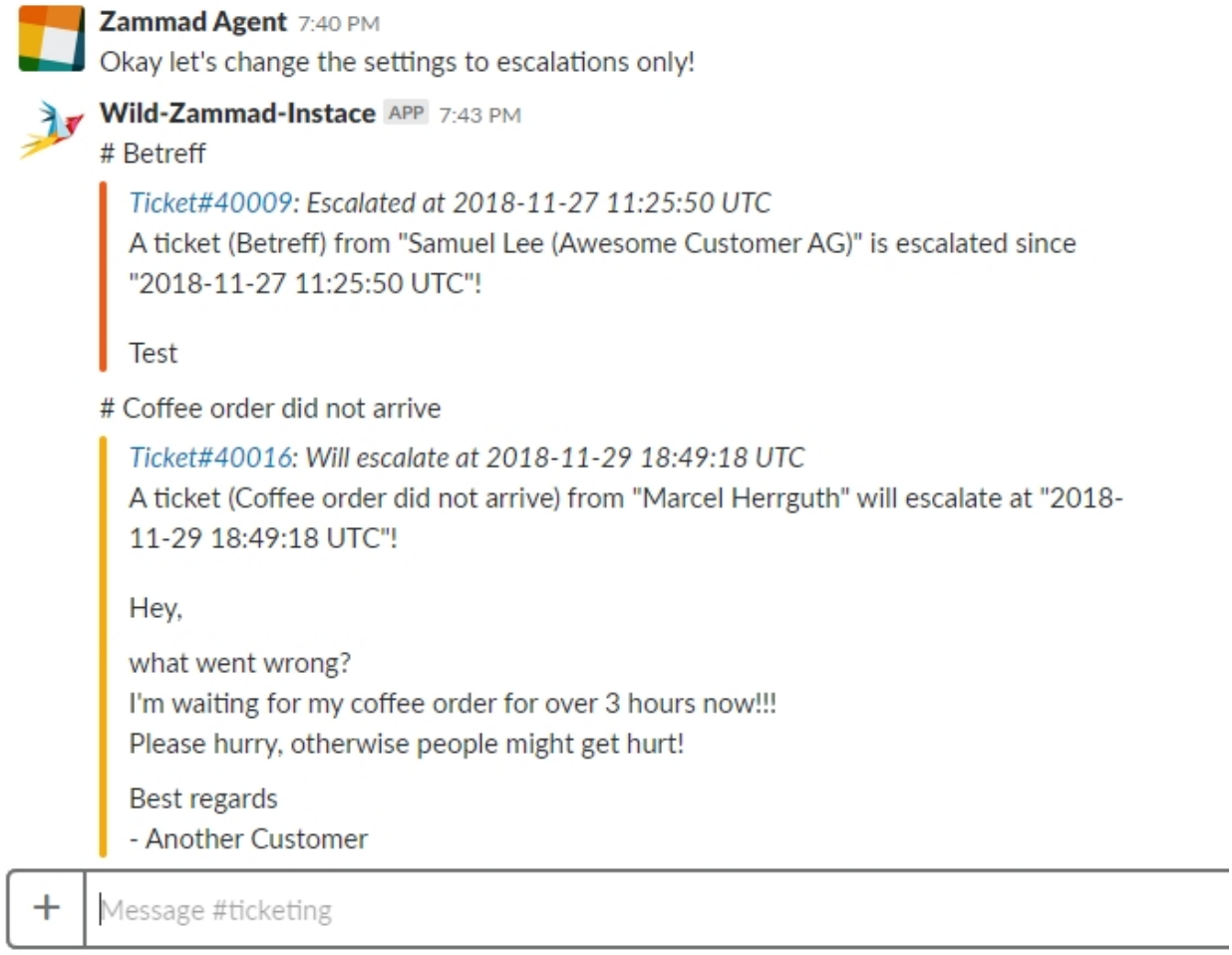


The result

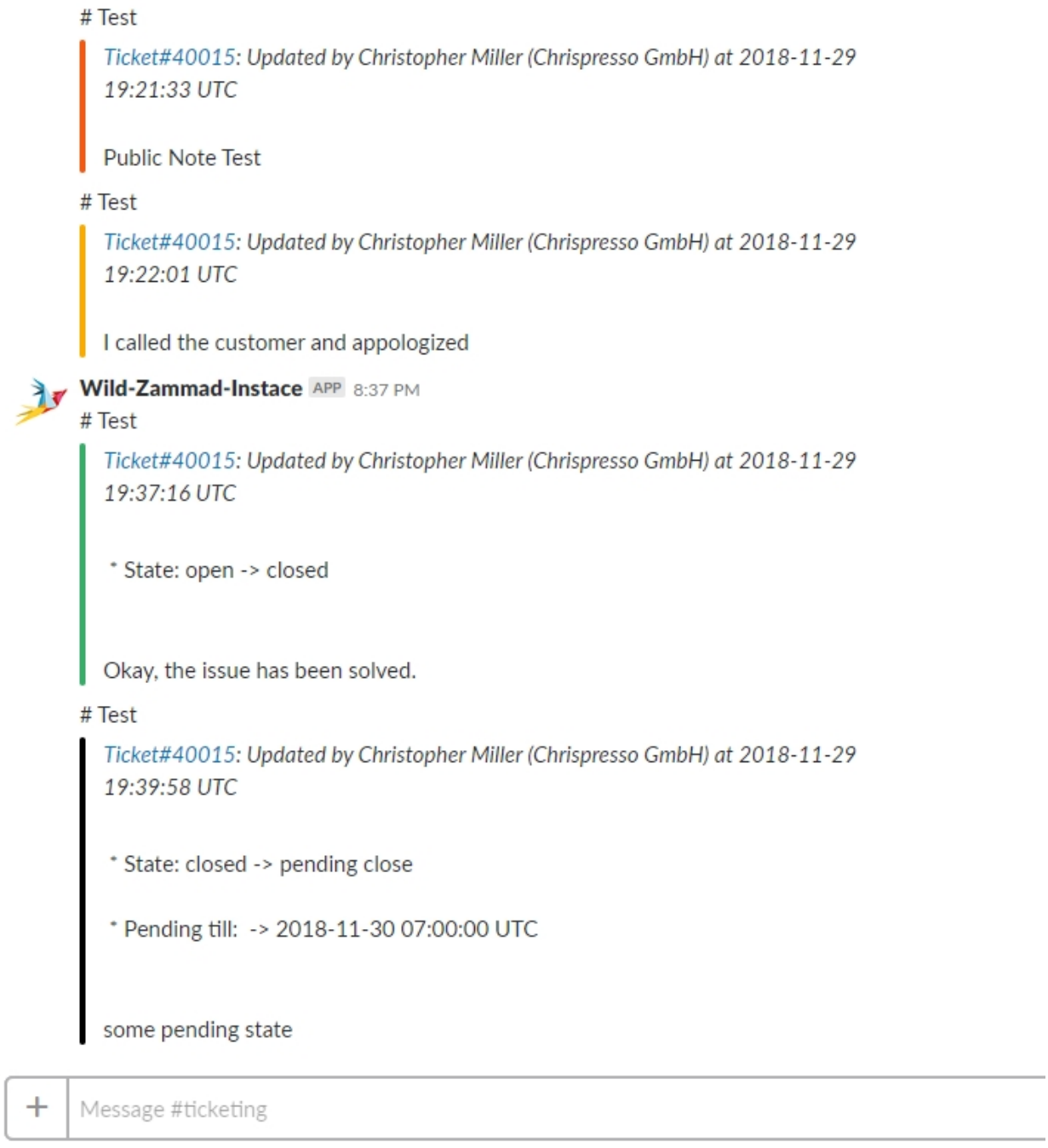
The following figure shows how it will look if you choose to receive updates on created and updated tickets. On every post Zammad sends to the Slack channel, you can create new threads to discuss about the new article.



If you just want to keep track of soon escalating or already escalated tickets, it will look the the following figure. Zammad changes the color in front of the post so you can easily see the state of the ticket.



If you change the state, Zammad will also put information on what state the ticket gets and (if pending state) the date until it pends. You'll also recognize the color codes in front of posts on slack, as they are exact the same the ticket state colors you'll see in Zammad!


Test
Ticket#40015: Updated by Christopher Miller (Chrispresso GmbH) at 2018-11-29 19:21:33 UTC
Public Note Test
Test
Ticket#40015: Updated by Christopher Miller (Chrispresso GmbH) at 2018-11-29 19:22:01 UTC
I called the customer and appologized
Wild-Zammad-Instace APP 8:37 PM
Test
Ticket#40015: Updated by Christopher Miller (Chrispresso GmbH) at 2018-11-29 19:37:16 UTC
* State: open -> closed
Okay, the issue has been solved.
Test
Ticket#40015: Updated by Christopher Miller (Chrispresso GmbH) at 2018-11-29 19:39:58 UTC
* State: closed -> pending close
* Pending till: -> 2018-11-30 07:00:00 UTC
some pending state
Message #ticketing

33.5.2 i-doit

i-doit is an open-source **configuration management database**—in other words, a tool for keeping tabs on every single piece of your physical and digital infrastructure, from network equipment to virtual machines on down to the faceplates on your rack shelves and more.

What's that got to do with Zammad? Well, if you used tickets to track issues with all that hardware, you might start wishing there was a way they could talk to each other.

Zammad gives you two:

1. Add i-doit Links to Zammad Tickets

What users see

Fig. 10: The i-doit integration will appear under a new tab in the ticket pane.

i-doit integration puts a new tab in Zammad's ticket pane where you can add links to existing i-doit devices for easy reference. (See our user documentation to learn how it works in practice).

How to set it up

Note: **Requires i-doit's** API Add-on.

Use the following settings:

Active Yes

Enforce autentication by username and password No

To set it up, enable the integration in the Zammad admin panel under **System > Integrations > i-doit**:

Endpoint The root URL of your i-doit installation.

API token Found in the i-doit admin panel under **Interfaces / external data > JSON-RPC API > Common Settings**.

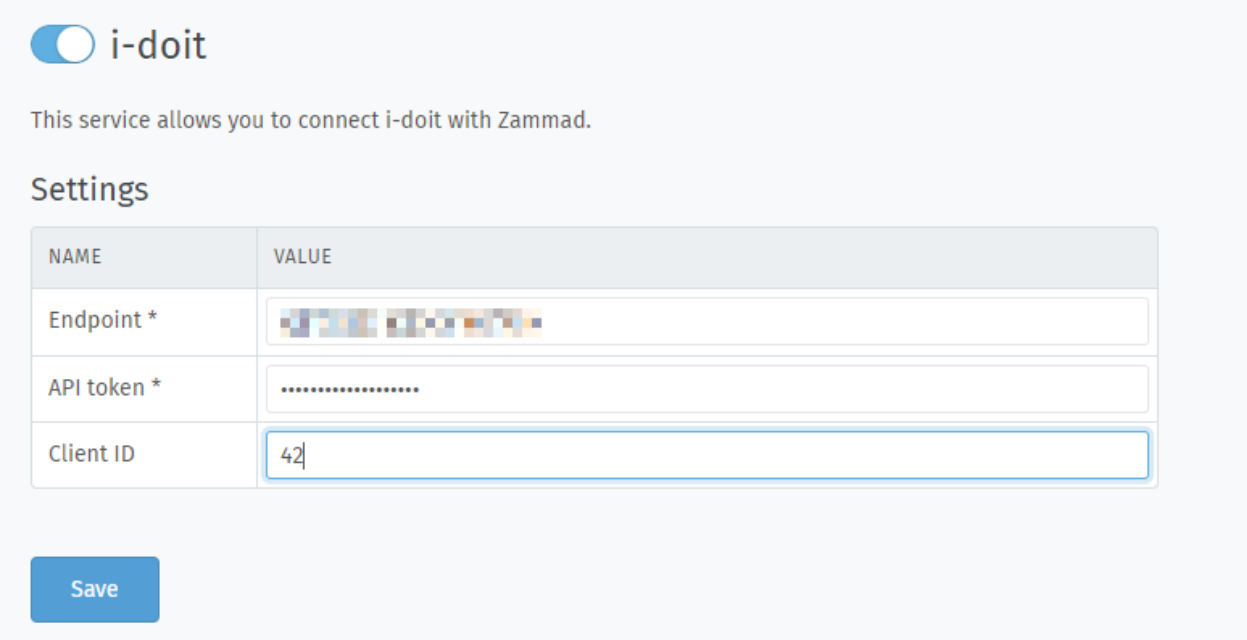
Client ID A unique name to identify Zammad within i-doit.

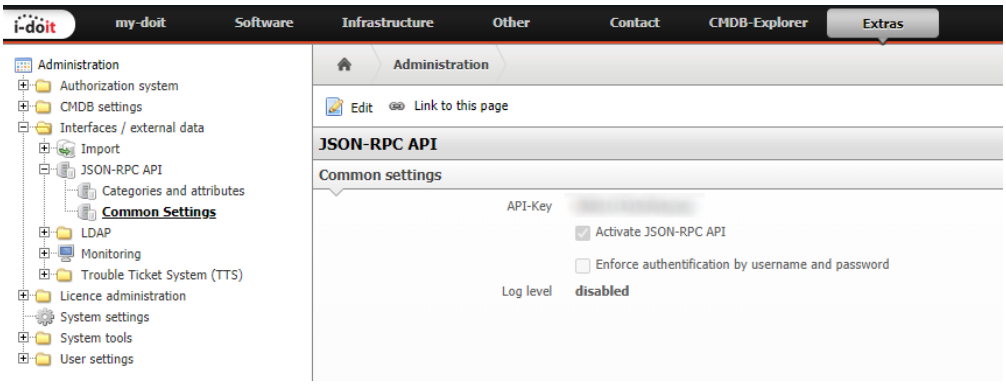
(Zammad does not require you to enter a value here, but i-doit might!)

2. List / Create Zammad Tickets in i-doit

What users see

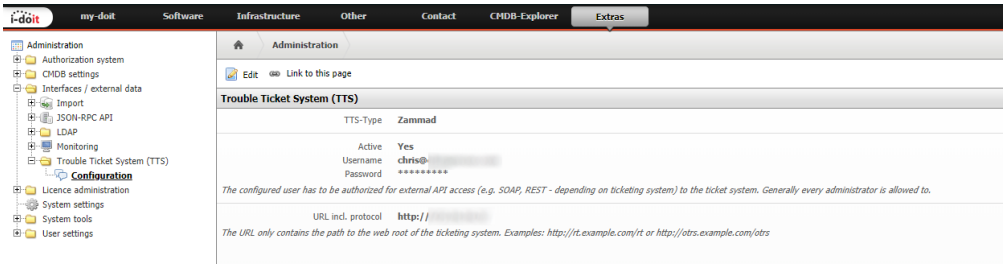
i-doit's ticket system integration gives you a way to see all the tickets for a given device without ever leaving i-doit. (See our user documentation to learn how it works in practice).


i-doit
This service allows you to connect i-doit with Zammad.
Settings
NAME
VALUE
Endpoint *
API token *
Client ID
42
Save


i-doit
my-doit
Software
Infrastructure
Other
Contact
CMDB-Explorer
Extras
Administration
Authorization system
CMDB settings
Interfaces / external data
Import
JSON-RPC API
Categories and attributes
Common Settings
LDAP
Monitoring
Trouble Ticket System (TTS)
Licence administration
System settings
System tools
User settings
Administration
Edit
Link to this page
JSON-RPC API
Common settings
API-Key
Activate JSON-RPC API
Enforce authentification by username and password
Log level
disabled

How to set it up

Enable this integration in the i-doit admin panel under **Interfaces / external data > Trouble Ticket System (TTS) > Configuration**:



TTS-Type `Zammad`

Username / Password Login credentials for a Zammad agent.

> **Note:** This agent must have **read permission** for all *groups* that plan on using the i-doit integration.
>
> You may even wish to create a dedicated agent account just for this integration. (Otherwise, if the agent ever changes her password, you will have to remember to update it here.)

URL incl. protocol `https://your.zammad.domain`

33.5.3 Elasticsearch (SaaS)

The Elasticsearch integration allows you to create a read-only user to use with your favorite reporting tool (e.g. like Grafana).

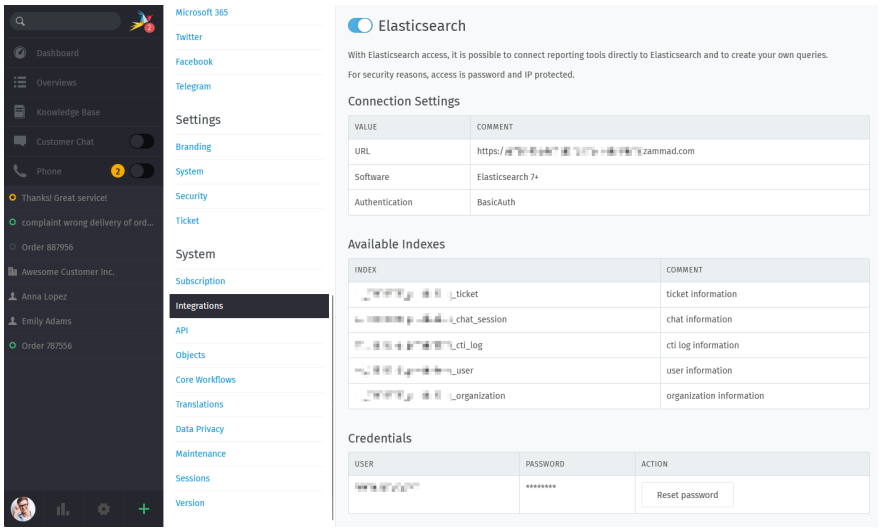
If you want to take full advantage of the Zammad reporting, have a look at our Grafana setup page as well.

> **Warning: Hosted environment specific**
>
> This integration is only available for Hosted setups. In order to use Elasticsearch you'll need the **Plus** subscription.
>
> Self hosted users have all the control over their self hosted Elasticsearch instances.

Limitations

Please note the following limitations of Elasticsearch access on hosted environments:

- access to the Elasticsearch index is read-only access
- currently you're limited to user only
- Reporting tools that require to write into the indexes (like Kibana) are not supported
- IP access restriction is currently not yet supported



Activating Elasticsearch access

By default external access to your Elasticsearch index is not active. You can enable the integration at any time if needed.

Please ensure to note down the password provided - you won't have access to it afterwards.

Connection Settings

This section holds the most important general information for accessing your Elasticsearch indexes - such as:

URL A unique subdomain that does not tell your real instance URL.

Software The major version of the search index being used. This is required by some Reporting tools like Grafana.

Authentication The authentication type being supported. `Basic Authentication`

Available Indexes

Within this section we're displaying the -in our opinion- most important indexes for a Zammad instance.

Tip: If you require all indexes or our listing is not good enough for you, point your browser to the URL we're providing and append `/_aliases?pretty=true`. The result should look like so: `https://<URL>.zammad.com/_aliases?pretty=true`.

Your browser will automatically ask for your credentials - you'll then see something like this:

```
{
   "XXXXXXXX" : {
      "aliases" : { }
   },
   "XXXXXXXX_cti_log" : {
      "aliases" : { }
   },
   "XXXXXXXX_knowledge_base_answer_translation" : {
```

(continues on next page)

(continued from previous page)

```
      "aliases" : { }
   },
   "XXXXXXXX_ticket" : {
      "aliases" : { }
   },
   "XXXXXXXX_knowledge_base_category_translation" : {
      "aliases" : { }
   },
   "XXXXXXXX_knowledge_base_translation" : {
      "aliases" : { }
   },
   "XXXXXXXX_ticket_state" : {
      "aliases" : { }
   },
   "XXXXXXXX_user" : {
      "aliases" : { }
   },
   "XXXXXXXX_stats_store" : {
      "aliases" : { }
   },
   "XXXXXXXX_chat_session" : {
    "aliases" : { }
   },
   "XXXXXXXX_group" : {
    "aliases" : { }
   },
   "XXXXXXXX_ticket_priority" : {
      "aliases" : { }
   },
   "XXXXXXXX_organization" : {
      "aliases" : { }
   }
}
```

Credentials

Within this section Zammad displays your available users. The password is provided once (upon activation) and cannot be retrieved after that.

> **Note: I need my Elasticsearch user password to be changed**
>
> Within the credentials table use the "Reset password" button to receive a brand new password for the account in question. This change is immediate, keep in mind that this may affect third party tools connected to your instance.
>
> *Above does not change indexes.*

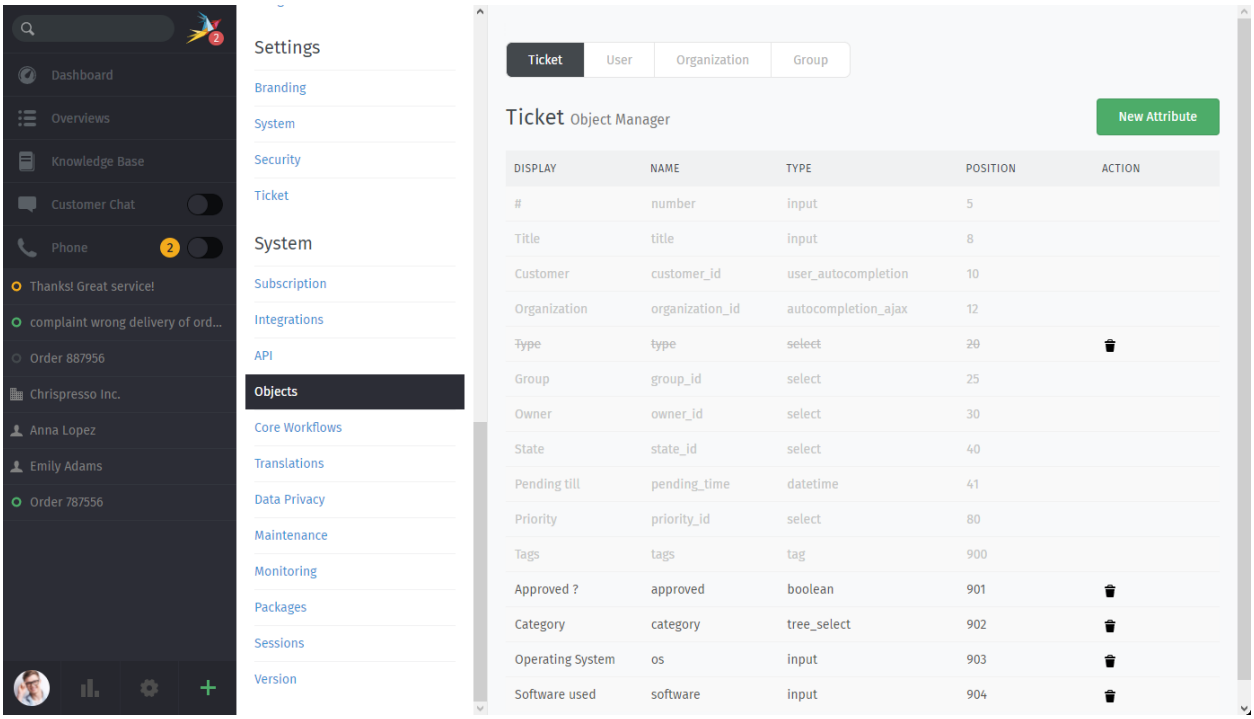
CHAPTER

THIRTYFOUR

OBJECTS

In Zammad you can add your own fields to tickets, users, organizations and even groups. This can be useful if you need to add further information to a ticket that you don't want to have in a note (you'll find it easier).

Note: Try to avoid deleting objects (and disable them instead) as Zammad might run into unexpected conditions if they are referenced somewhere.



Here's an overview of the objects. On the upper right you can add new Attributes. By default, there will be no custom fields - standard objects will be grayed out, you can't delete or change those. Custom objects (will be displayed in black font and have a trash bin on the right site to delete unnecessary objects. By clicking on "custom objects" you can edit them to suit your needs.

Note: Attributes you add to Zammad, no matter if they have default values or not, will not update existing information. This means a new ticket field technically will be empty unless you populate it.

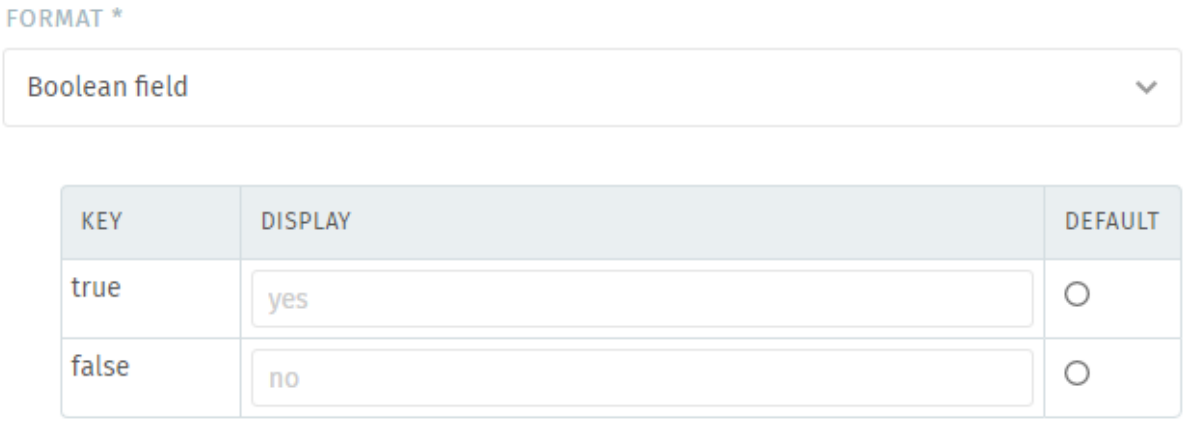
Especially in ticket scope this also means that newly added attributes will be indicated as "changed" to agents that view the ticket. This may interfere with Zammad's tabs behavior.

34.1 Object types

When adding a new object, you can choose between the following object types.

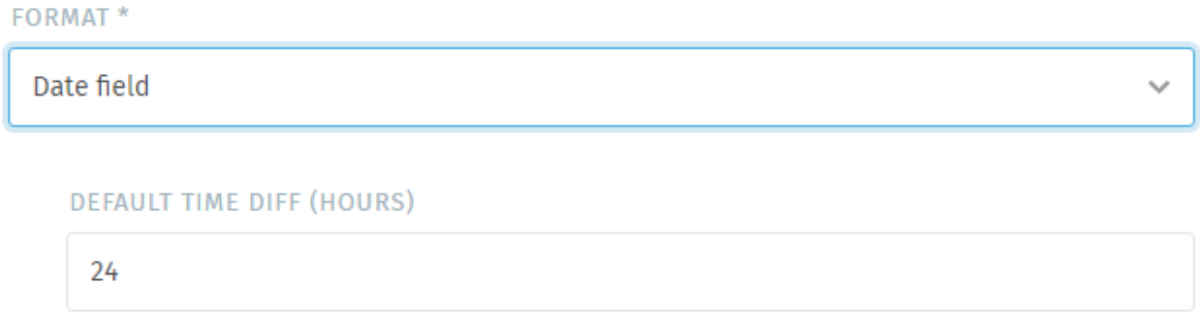
> **Warning:** You cannot change the object format / type as soon as it is applied. If you no longer need an object, consider disabling it instead of removing.

Boolean field Provides a drop-down field with display values for `true` and `false`. Allows setting a default value.



Date field Provides a date picker field and does not allow default values.

Default time diff (hours) This setting helps the user by highlighting the day from now plus the provided value. It *does not* pre-fill the field.



Date & time field Provides a date and time picker – does not allow default values

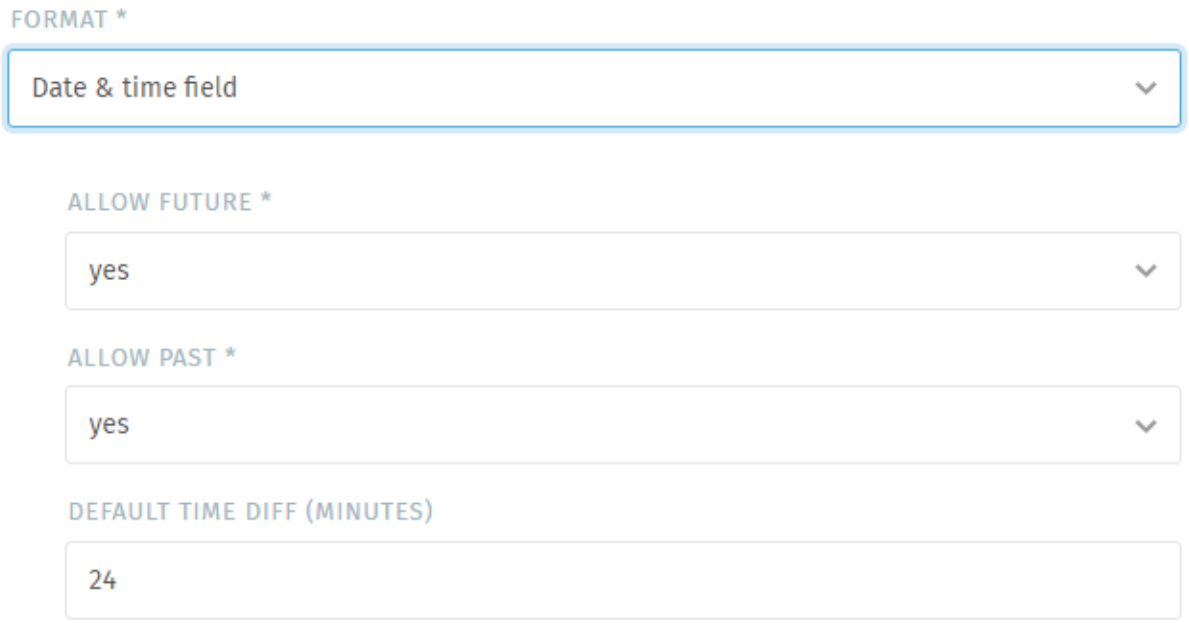
Allow future

Forbid dates and times in the future.
Default: `yes`

Allow past

Forbid dates and times in the past.
Default: `yes`

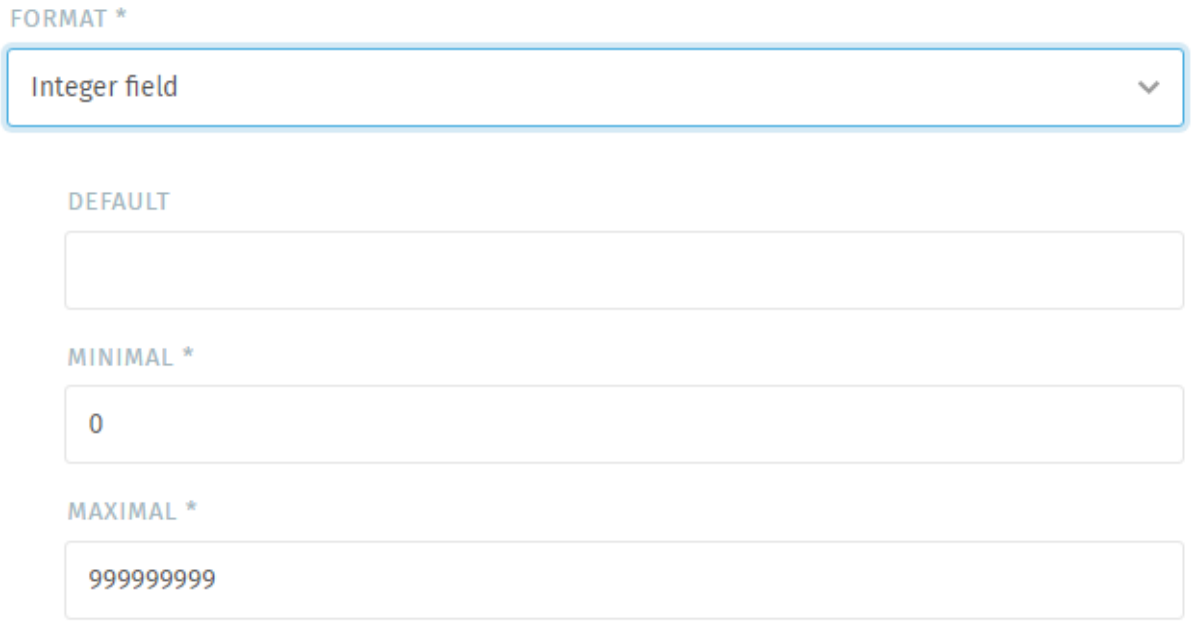
Default time diff (minutes) This setting helps the user by highlighting the day from now plus the provided value. It does not pre-fill the field.



Integer field Provides an input field that allows integer usage only. You may define a default value. You cannot enforce comma separation.

Minimal The minimal value the field accepts.

Maximal The maximum value the field accepts.

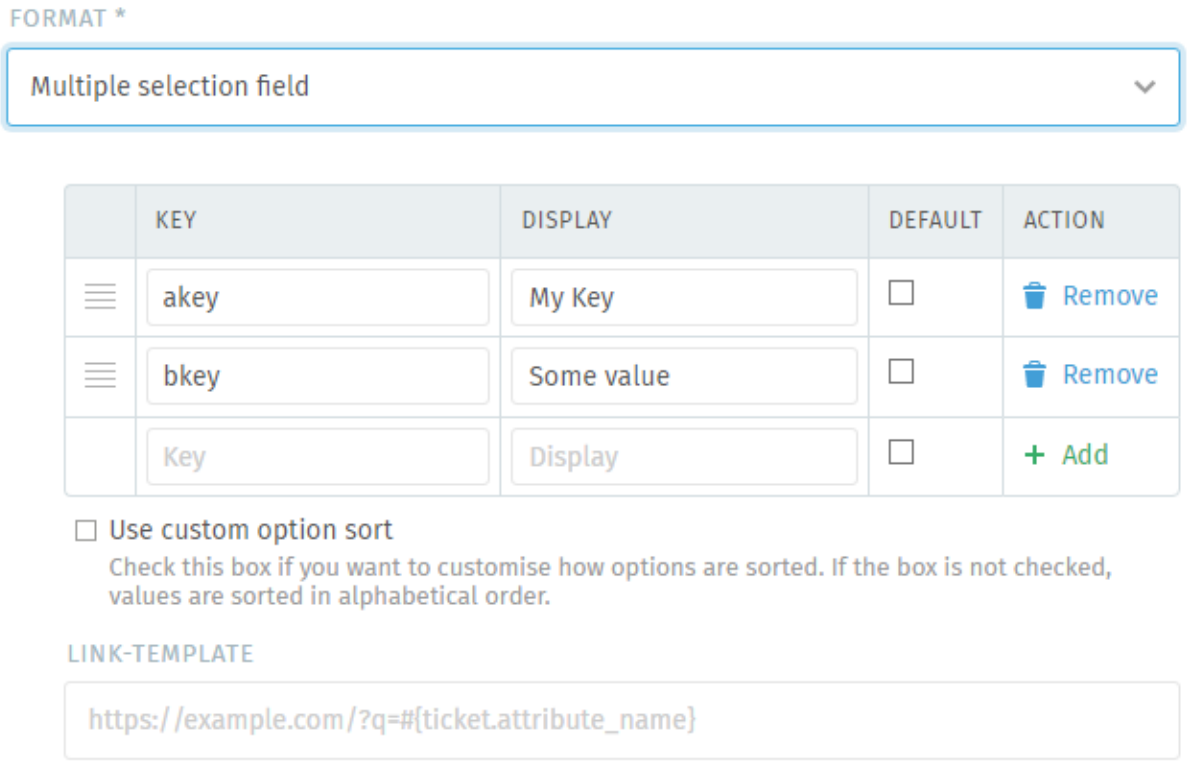


Multiple selection field Provides a selection field that allows the selection of one or more out of several. This field does allow setting a default value.

Tip: Adding values can be tricky for first timers, don't forget to press " Add" after typing your values. Otherwise

you may loose a value.

Hint: This field allows using *URL fields (Link Templates)*.



Tip: This object allows position of its values

In order to re-arrange the fields options, edit the field and scroll below the values. Make sure to tick the option "Use custom option sort".

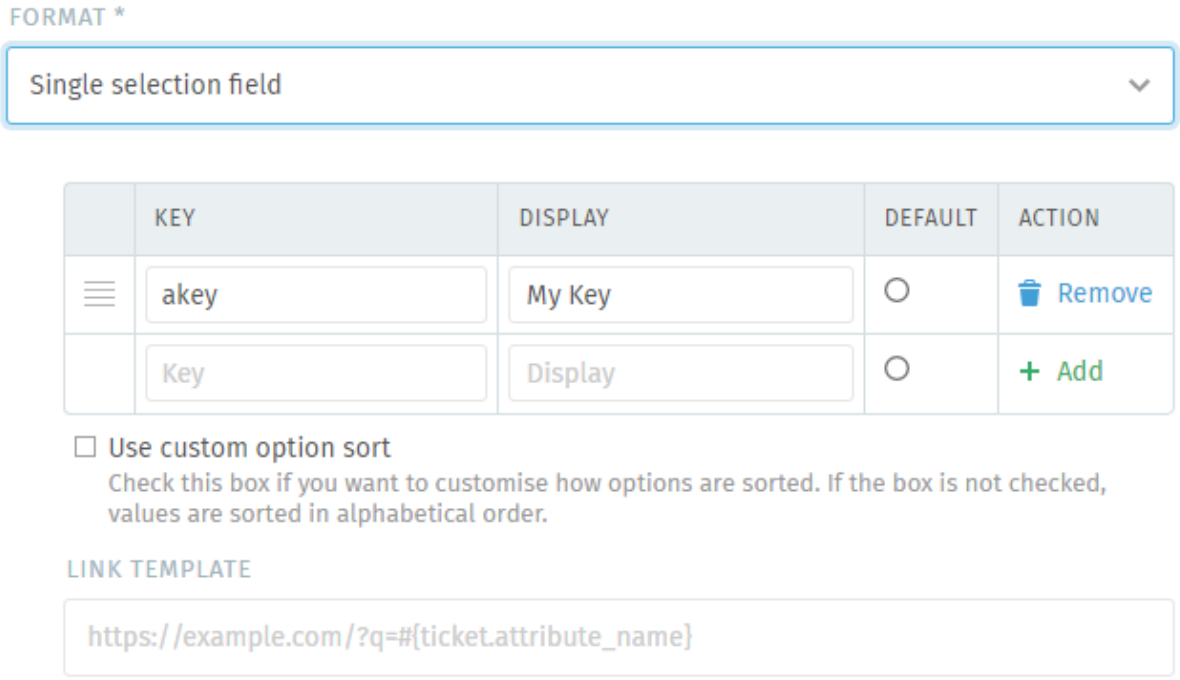
Warning: If you do not tick this field, all manual position you did above will be lost upon saving!

Now use to drag the values in question to the correct position. When you're ready, submit your changes to save the object.

Single selection field Provides a drop-down field that allows selection of one value out of several. This field does allow setting a default value.

Tip: Adding values can be tricky for first timers, don't forget to press " Add" after typing your values. Otherwise you may loose a value.

Hint: This field allows using *URL fields (Link Templates)*.



Tip: This object allows position of its values

In order to re-arrange the fields options, edit the field and scroll below the values. Make sure to tick the option "Use custom option sort".

Warning: If you do not tick this field, all manual position you did above will be lost upon saving!

Now use to drag the values in question to the correct position. When you're ready, submit your changes to save the object.

Textarea field Provides a text area input field (multiple lines) and thus allows e.g. new lines. You can set a default field value.

Note: Please note that this field *does not* support text formatting or HTML content (rich text).

Warning: This field can consume a lot of visual space

Depending on where you use this field type, it may use a lot of visual space if you provide a lot of text. This may be an issue to work with.

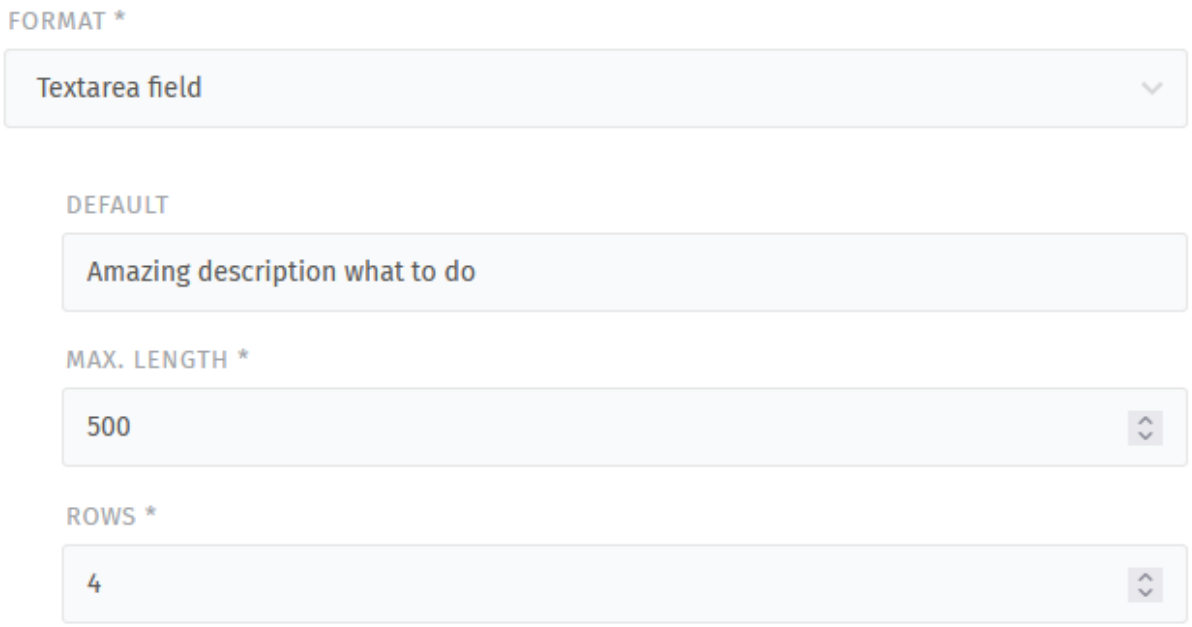
Default The here provided text will be shown within the text area field or new data sets.

Maxlength

You can pick the maximum length of the field.
The default length of this object is `500`.

Rows

Change the number of rows to dislay so that you can use only the space you really need.
The default number of rows is `4`.



Text field Provides a text field (one line) and allows choosing a default value.

Type Defines the type of the input field. This allows e.g. your browser to ensure that you provide the specific type.

Currently available:

- Email
- Phone
- Text
- Url *(URL fields disable link-template availability)*

Maxlength You can pick the maximum length of the field.

Hint: This field allows using *URL fields (Link Templates)*.

Single tree selection field Provides a select-like field with up to 6 layers of options. Does not allow setting a default value.

Tip: This object allows position of its values

In order to re-arrange the fields options, edit the field and to the values.

Use to drag the values in question to the correct position. If you want to change the layer depth, double click on . By this you can cycle through the available layers.

When you're ready, submit your changes to save the object.

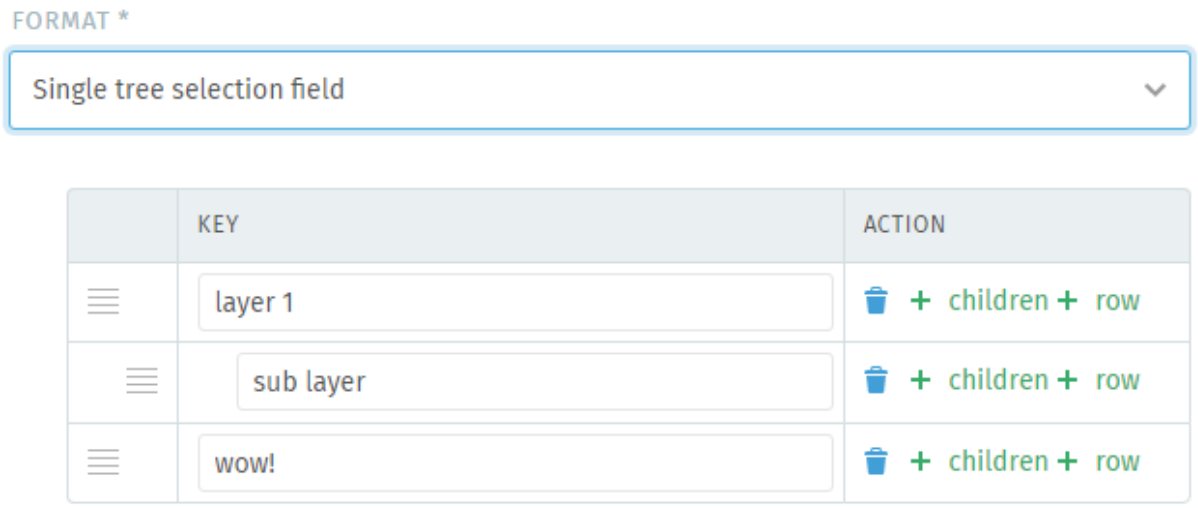
Multiple tree selection field Provides a select-like field with up to 6 layers of options allowing the selection of multiple values. Does not allow setting a default value.

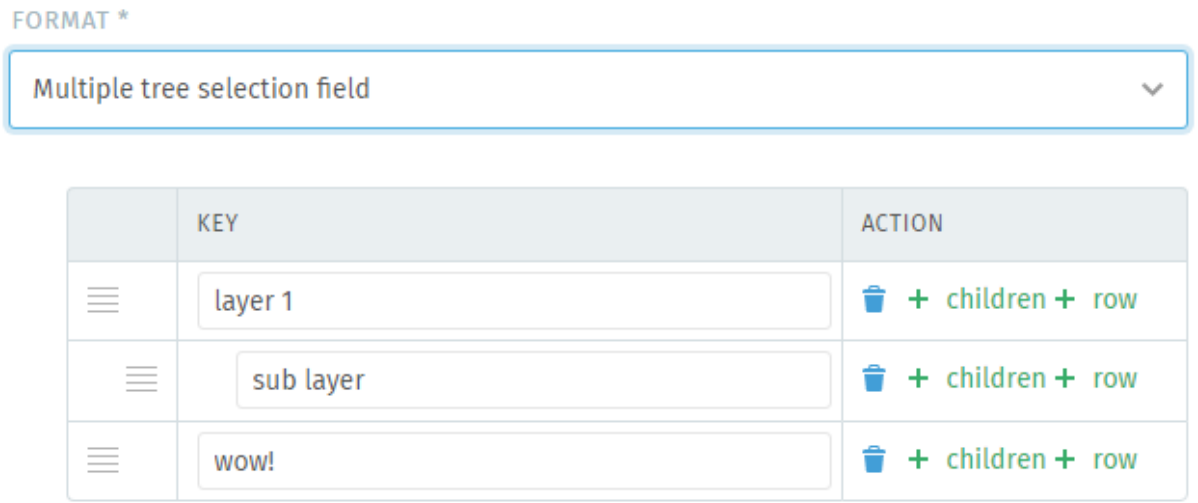
Tip: This object allows position of its values

In order to re-arrange the fields options, edit the field and to the values.

Use to drag the values in question to the correct position. If you want to change the layer depth, double click on . By this you can cycle through the available layers.

When you're ready, submit your changes to save the object.


FORMAT *
Single tree selection field
KEY
ACTION
layer 1
+ children + row
sub layer
+ children + row
wow!
+ children + row


FORMAT *
Multiple tree selection field
KEY
ACTION
layer 1
+ children + row
sub layer
+ children + row
wow!
+ children + row

34.1.1 URL fields (Link-Template)

Note: This function is restricted to Text and Select objects only.

Link-Templates are an amazing way to dynamically generate URLs. They allow you to integrate other systems better without having to manually copy data from Zammad if possible.

Note: Another great way of communicating with another system may be Zammad's *Webhooks*.

After filling a link-template enabled field, an URL icon will appear on its right. Clicking on the icon opens a new tab.

Hint: Even though Zammad displays the link template within object edit and create screens, the function is optional. It's only active if you populate the field.

What's the difference between URL and text fields with link template...?!

Both fields have different use cases. Use `text` type text fields when ever you have a *static* url that requires dynamic parameters. If you require a drag & drop like field that you can put in any URL, use `URL` type text fields.

The difference is easier to spot when comparing the fields directly, below screencast shows the result - the dynamic approach uses existing values in the moment of updating to built the URL - e.g. `https://google.com/search?q=cookies` - while the URL approach uses the actual URL of the field - e.g. `https://zammad.com`.

How does this work...?!

As an example, let's say you have an object called `amazingobject` - you want to open a google search directly with the input from that field.

Providing below to the link-template field allows you to do so: `https://www.google.com/search?q=#{ticket.amazingobject}`

Tip: You can use any *Zammad variable* as long as it's available in the moment you need it.

The result will look as follows.

Fig. 1: The above screencast shows how the link template will perform after object creation.

34.2 Object permissions

PERMISSIONS	SCREEN	OPTIONS
ticket.customer		
	create_middle	shown: ☑ required: ☐
	edit	shown: ☑ required: ☐
	Not applicable to: merging, emails, form, Facebook, Telegram, Twitter, SMS	
ticket.agent		
	create_middle	shown: ☑ required: ☐
	edit	shown: ☑ required: ☐
	Not applicable to: merging, emails, form, Facebook, Telegram, Twitter, SMS	

Fig. 2: Some of the possible permission and screen options for objects.

Whenever needed you can restrict access to objects based on the *user permission* (`admin`, `ticket.agent` & `ticket.customer`).

Tip: Below is not set in stone

You can always adjust below settings with *Core Workflows*. This also allows role based restriction.

Note: In some situations, Zammad internally overrules below screen, requirement and permission settings. This is because at some points you can't set fields which would mean we couldn't create the ticket.

This currently affects:

- merging
- emails no matter of the originating channel (incoming)
- *Forms* (incoming)
- *Facebook* (incoming)
- *Telegram* (incoming)
- *Twitter* (incoming)
- SMS (incoming)

34.2.1 About screens

Zammad cares about the screen you're going to use the object in.

create Every time you use a creation dialogue for not yet existing data.

edit Every time you're editing existing data - viewing existing tickets counts as edit screen.

view Affects view screens of existing data like e.g. user profiles.

Note: This setting is available for the following object contexts:

- User
- Organization
- Group

---

invite_customer & invite_agent Shown when using the invitation dialogue from "First Steps" in the dashboard.

34.2.2 About screen options

Now that we know the different possible situations, let's talk about available options.

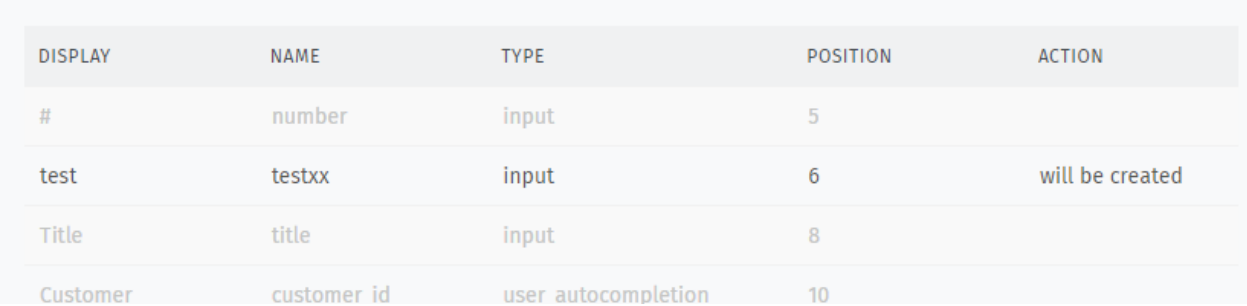
shown Show or hide a field.

required Set a field to mandatory. Forces users (via UI and API) to populate the field.

34.3 Ordering objects

Since Zammad introduced *Core Workflows* the need to have a custom positioning for objects has become more important than ever.

To adjust the position of an object, simply click on the object entry in question, scroll down and adjust the position number.



DISPLAY	NAME	TYPE	POSITION	ACTION
#	number	input	5	
test	testxx	input	6	will be created
Title	title	input	8	
Customer	customer_id	user_autocompletion	10	

Hint: In case two objects have the same position value, Zammad will sort alphabetically by name automatically.

Limitation
Please note that you cannot change the positioning of default objects at this time.

34.4 Updating database after adding or editing objects

When adding or changing objects, Zammad will not apply the changes instantly, but instead shows you the changed objects first. If you're ready to go, just click on "Update database" to apply the changes to Zammad. If you made a mistake or just want to discard your changes, click "Discard changes".

Warning: After applying the object changes with "Update Database" a restart of Zammad is **mandatory**. If you don't perform it, you may experience unexpected behavior or even errors. You may want to do those kind of configurations during maintenance windows.

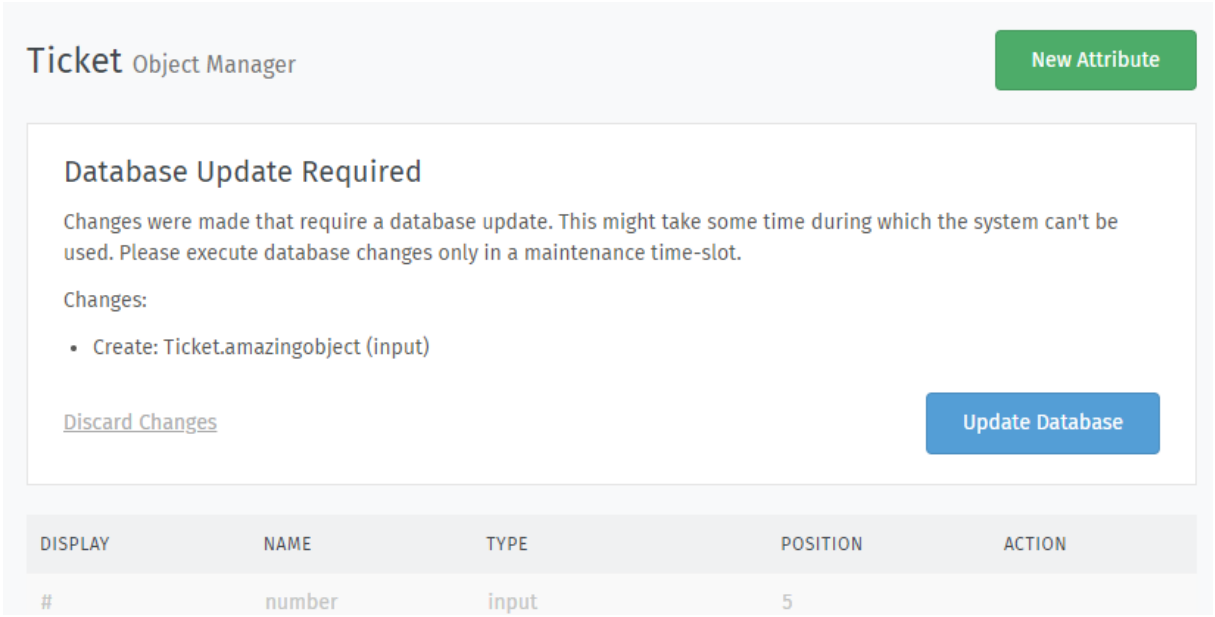


Fig. 3: Changes on objects require you to update the database to apply these changes.

Tip: Service restarts can be automated

Hosted environments automatically restart for you.
If you're using a self-hosted installation you can use environment variables

34.5 System objects

Zammad comes with pre-configured objects. Some of these currently do not provide the possibility to edit them via UI (or at all).

This is not a bug but is to save you from possibly nuking Zammad.

Tip: There are technical exceptions which can be solved via console. This e.g. affects ticket states and priorities, see console section.

If you're a hosted customer, please contact your support for more.

CHAPTER

THIRTYFIVE

CORE WORKFLOWS

Core Workflows allow you to create complex ticket processes just as (but not limited to):

- show / hide
- adjust mandatory setting
- manipulate available options

With this, you're all set to provide exactly those information you really need!

Note:

- If the pre-defined *Objects* are not enough, please add them beforehand.
- If you experience slow or unreliable field updates, please see *Core Workflow Ajax Modus*

Warning: This is a very enhanced functionality and can cause unexpected UI behavior. Please ensure to test your use cases after configuration to reduce surprises.

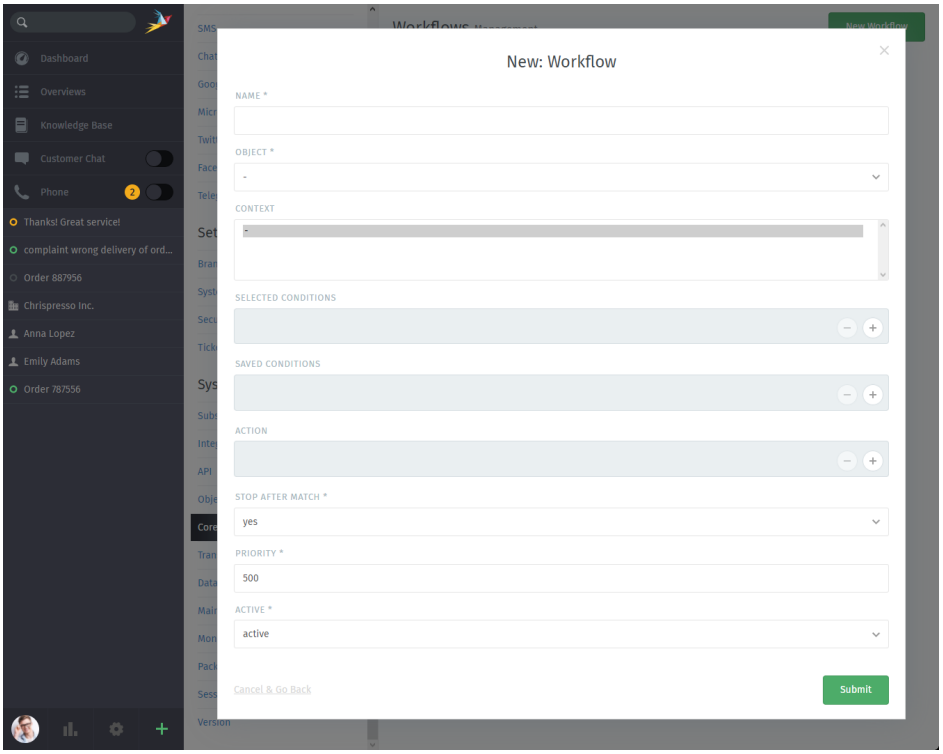
35.1 Learn by example

This page provides some of the ideas we had for Core Workflows. Of course you can build much more complex workflows.

Hint: If they don't make sense to you, don't worry—just skip ahead to *How do they work?* to learn about all the options in detail, then come back here to see them in action.

All following workflows have the same base configurations. The workflow may not use them all.

- **Groups**
  - Sales
  - Support
  - 2nd Level
- **Attributes**
  - Category (Single tree selection field, not mandatory, agents only)



- Approved (Boolean field, not mandatory, not shown, `false` as default)
- Operating System (Text field, not mandatory, not shown)
- Software used (Single selection field, not mandatory, not shown)

1. **Group specific values and fields** This workflow set depends on the category field. It reduces the available set of values based on the group selected.

   Workflow 2nd Level group

   This reduces the category options to `2nd Level/*`, `Internal/*` and `Others`. It also sets further required fields to mandatory and visible.

   Workflow Support group

   This reduces the category options to `Support/*`, `Internal/*` and `Others`. It also sets further required fields to visible.

   Workflow Sales group

   This reduces the category options to `Sales/*`, `Internal/*` and `Others`.

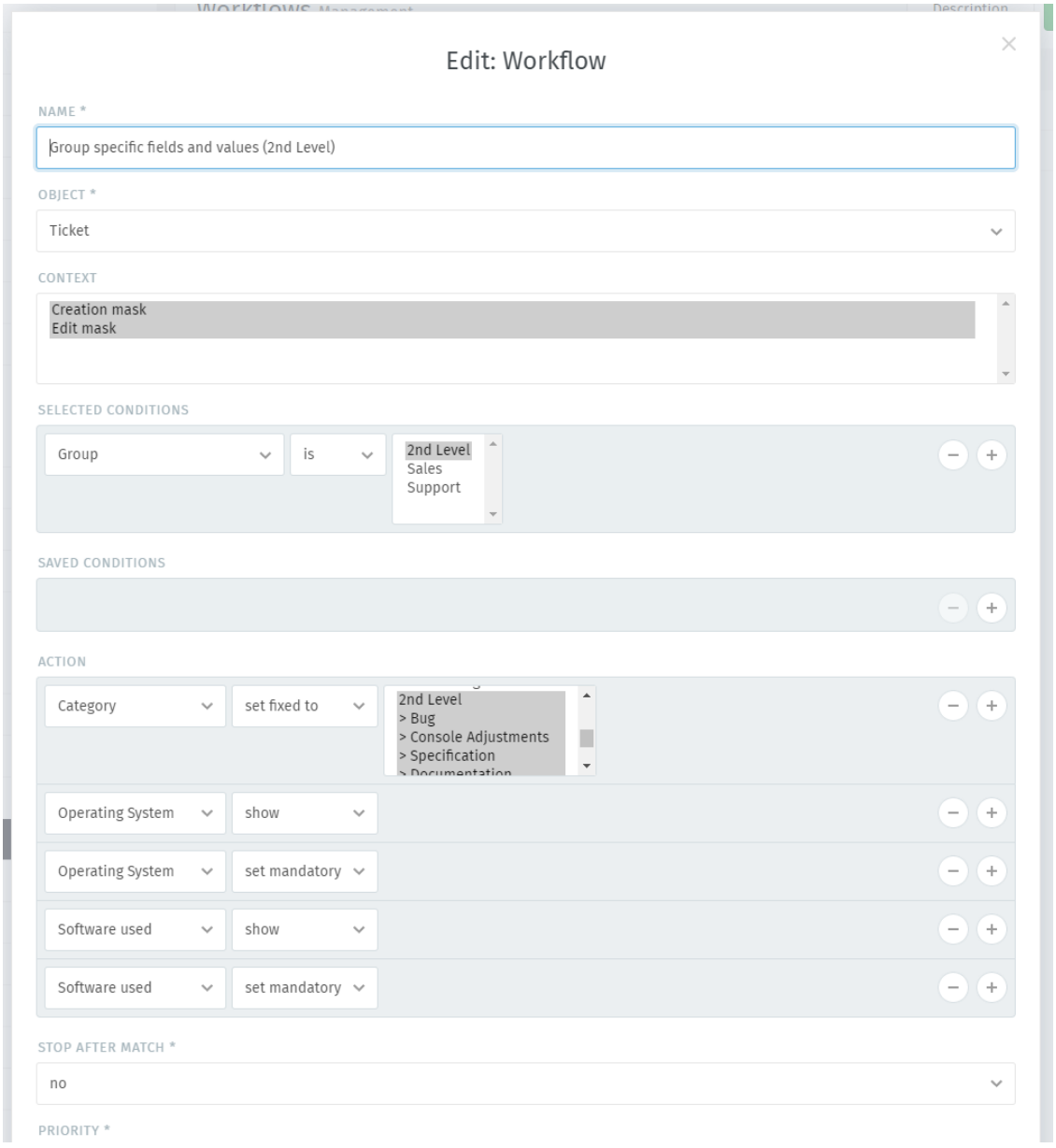
   The Result This is what the agent would experience with the above workflows in place.

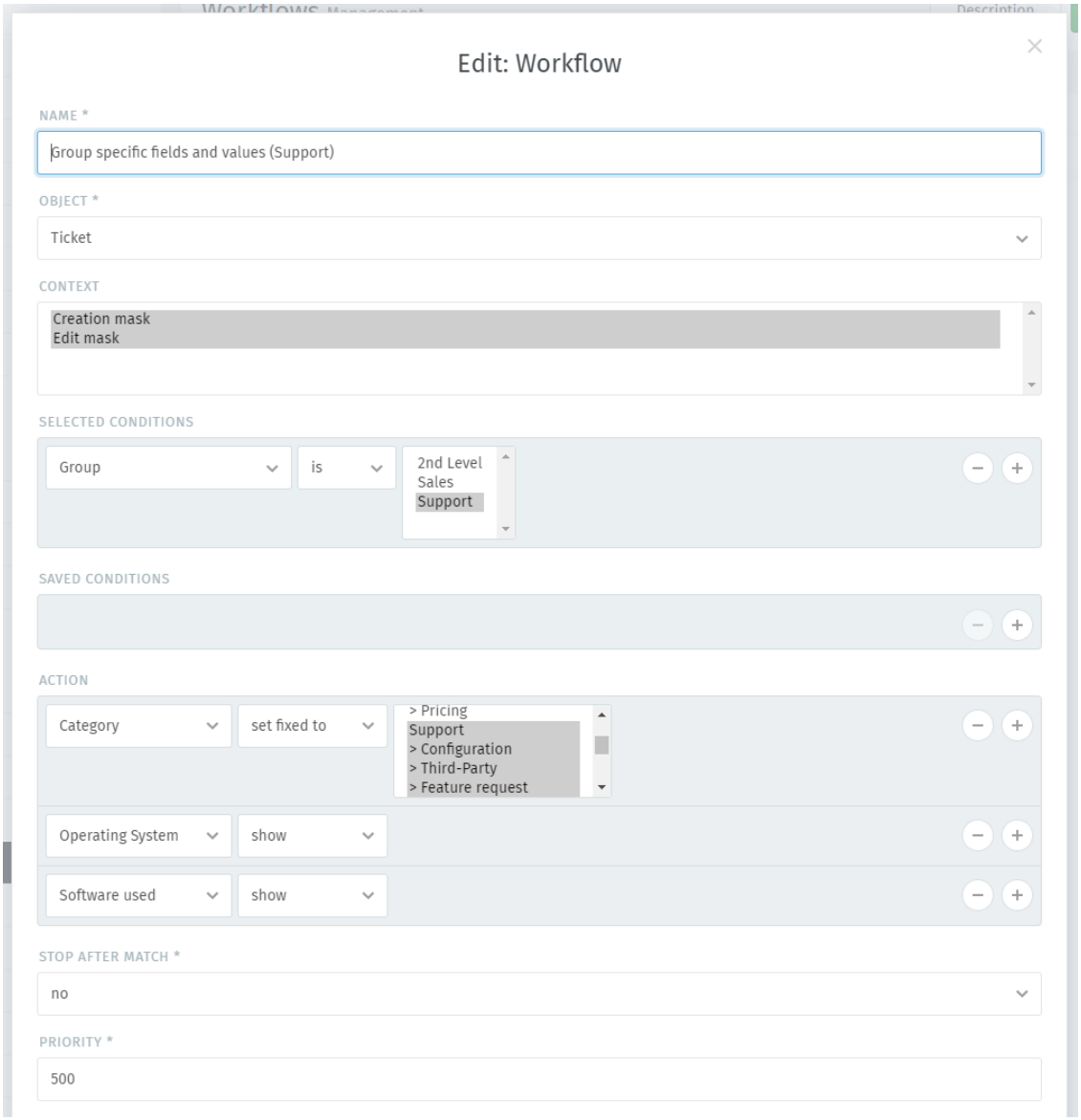
2. **Approval process** In this case `approved` is visible to agents by default. For this workflow, an additional role `Approval person` is required (no further permissions).

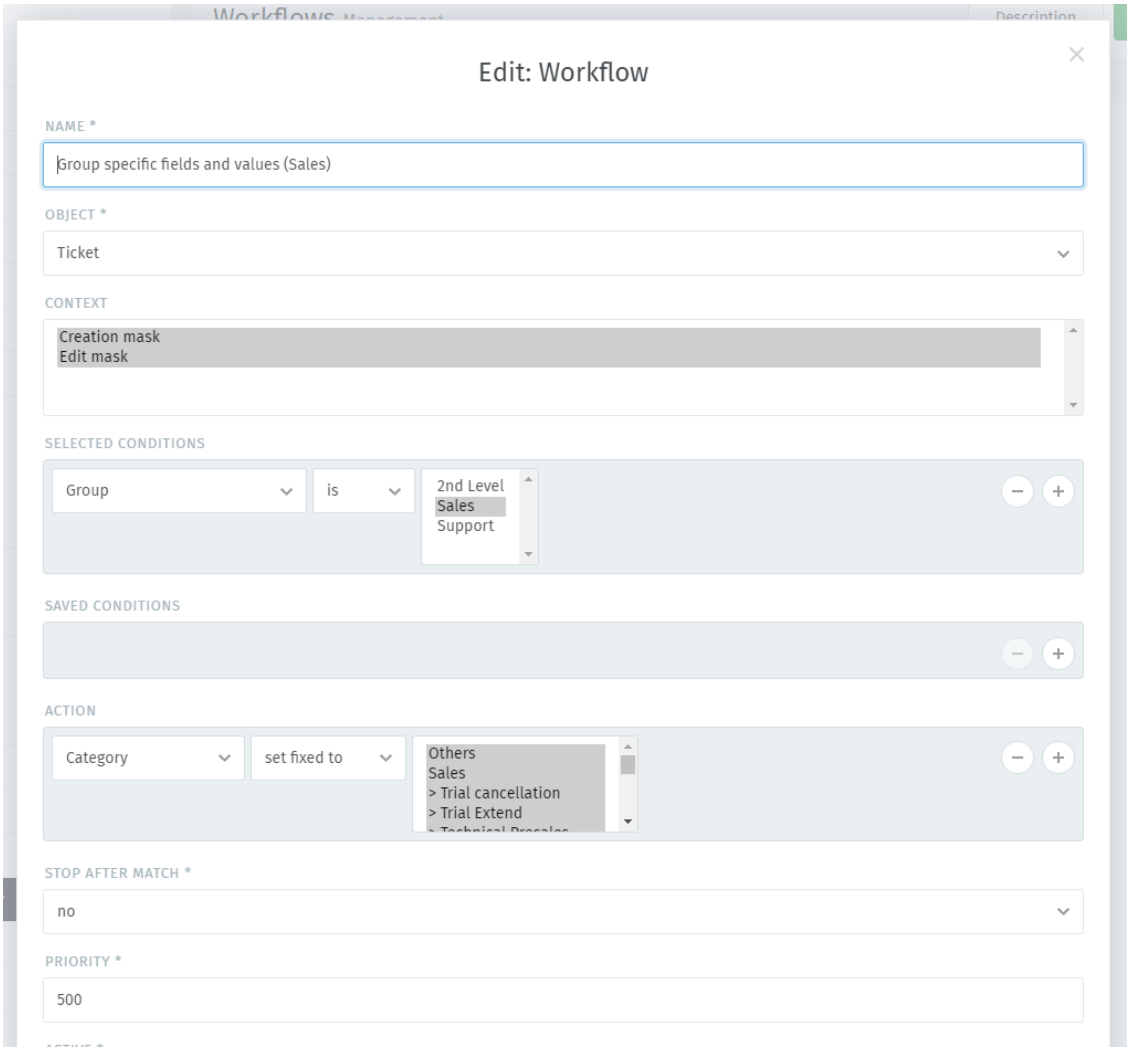
Tip: This workflow may work best in combination with a *trigger* but technically, this is not required.

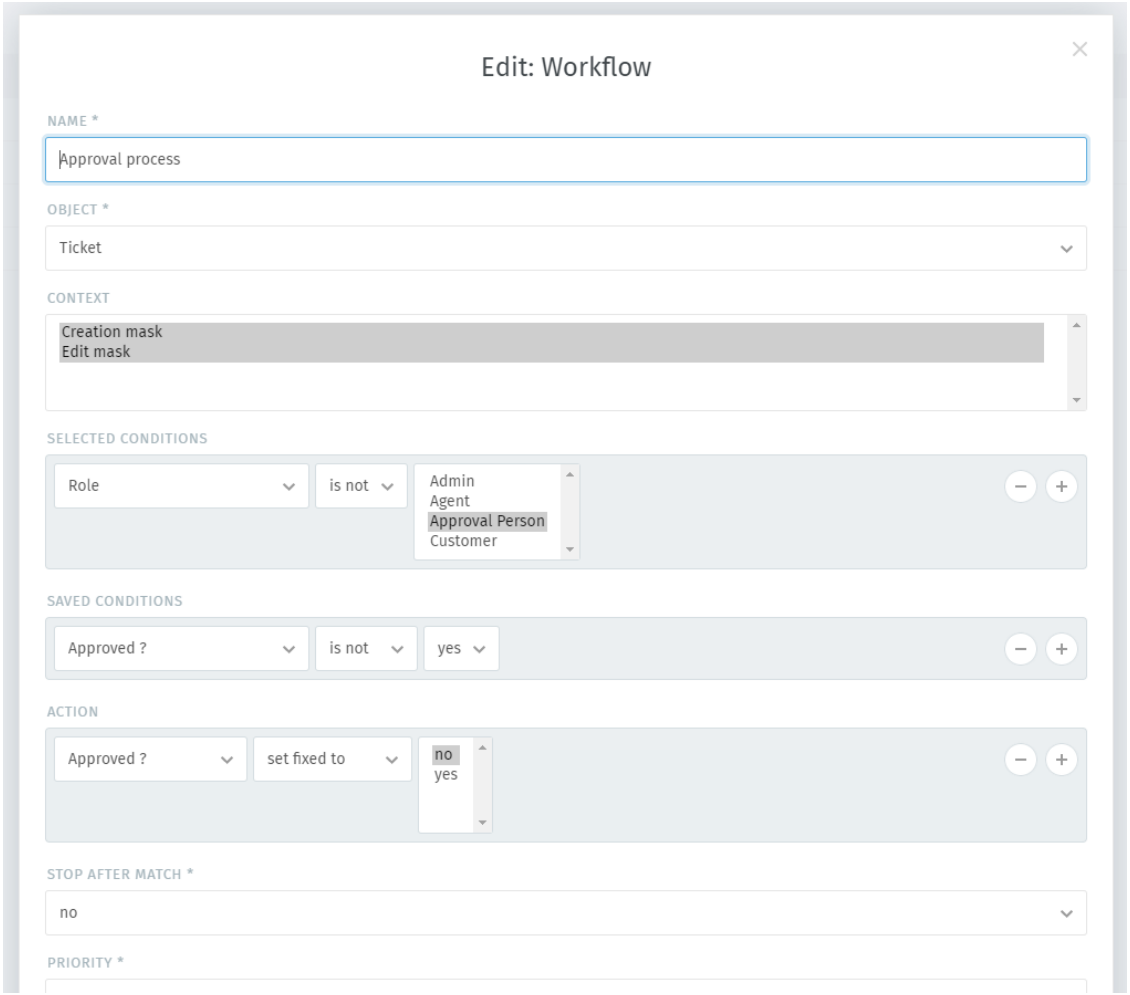
Select fields may be a better approach because they allow more values than just a simple `true` or `false`.

The result

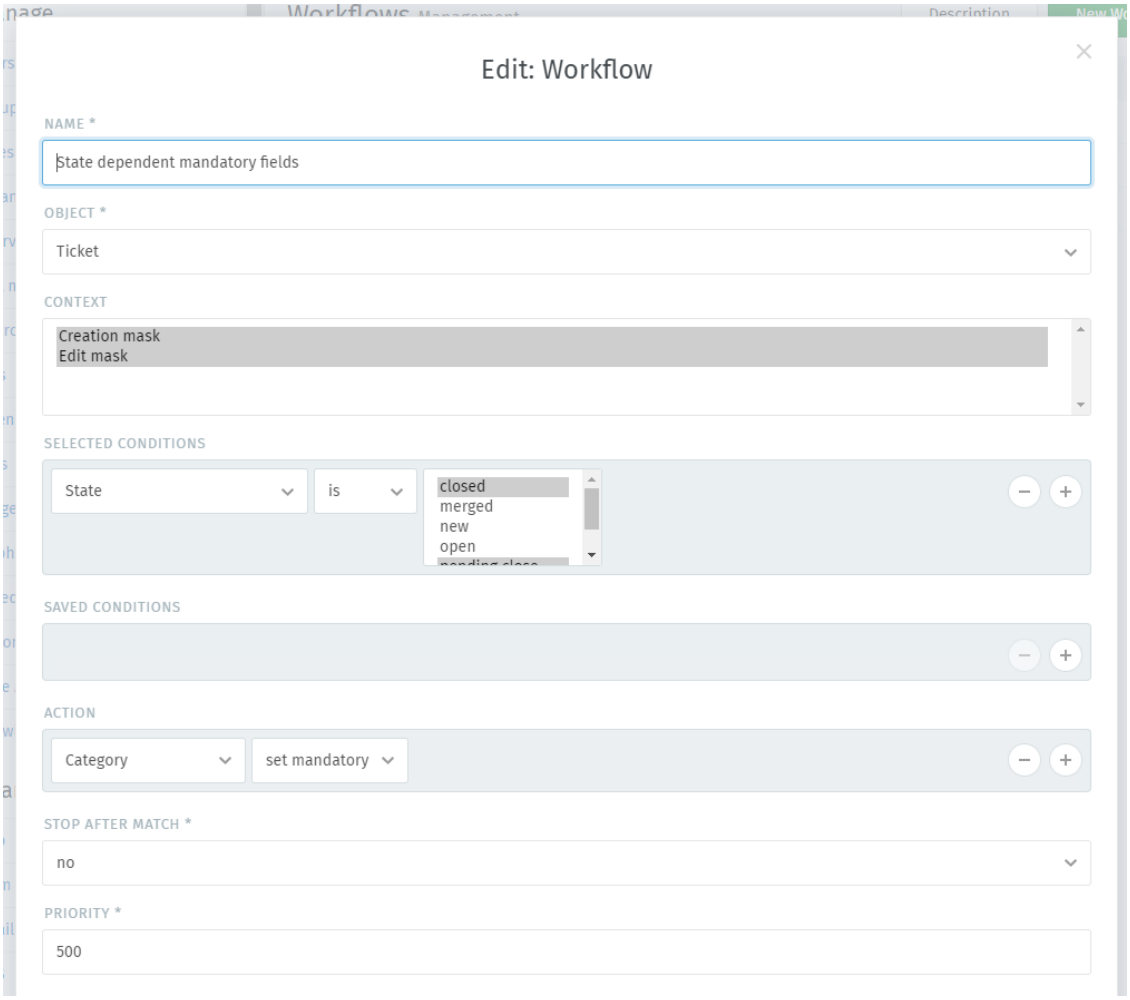

Edit: Workflow
NAME *
Group specific fields and values (2nd Level)
OBJECT *
Ticket
CONTEXT
Creation mask
Edit mask
SELECTED CONDITIONS
Group
is
2nd Level
Sales
Support
SAVED CONDITIONS
ACTION
Category
set fixed to
2nd Level
> Bug
> Console Adjustments
> Specification
Operating System
show
Operating System
set mandatory
Software used
show
Software used
set mandatory
STOP AFTER MATCH *
no
PRIORITY *


Edit: Workflow
NAME *
Group specific fields and values (Support)
OBJECT *
Ticket
CONTEXT
Creation mask
Edit mask
SELECTED CONDITIONS
Group
is
2nd Level
Sales
Support
SAVED CONDITIONS
ACTION
Category
set fixed to
> Pricing
Support
> Configuration
> Third-Party
> Feature request
Operating System
show
Software used
show
STOP AFTER MATCH *
no
PRIORITY *
500


Edit: Workflow
NAME *
Group specific fields and values (Sales)
OBJECT *
Ticket
CONTEXT
Creation mask
Edit mask
SELECTED CONDITIONS
Group
is
2nd Level
Sales
Support
SAVED CONDITIONS
ACTION
Category
set fixed to
Others
Sales
> Trial cancellation
> Trial Extend
STOP AFTER MATCH *
no
PRIORITY *
500


Edit: Workflow
NAME *
Approval process
OBJECT *
Ticket
CONTEXT
Creation mask
Edit mask
SELECTED CONDITIONS
Role
is not
Admin
Agent
Approval Person
Customer
SAVED CONDITIONS
Approved ?
is not
yes
ACTION
Approved ?
set fixed to
no
yes
STOP AFTER MATCH *
no
PRIORITY *

3. **State dependent mandatory fields** This workflow sets `Category` to mandatory if the agent wants to set the states `closed` or `pending close` to enforce categorization.



The result

35.2 How do they work?

Core Workflows are evaluated by priority. If 2 workflows have the same priority by alphabetical order by **name**. Workflows are evaluated in alphabetical order, by **name.**

Because of the way Core Workflows works all changes to attributes are checked with the application server – please see *Limitations* for possible issues.

Below we're talking about settings that have an impact and are not self-explanatory.

35.2.1 Object

Choose the object context you want to run the workflow in. This will decide on your available conditions and actions.

Tip: You will be able to use objects that are in relation to your selection in your conditions.

This means:
Ticket objects also have access to the ticket customer.

35.2.2 Context

Choose in which situation the workflow is applied. Contexts can be combined to reduce workflows.

Creation mask Once selected your conditions and actions will affect all applicable creation masks.

Edit mask Once selected your conditions and actions will affect all applicable edit masks.

35.2.3 Conditions

Zammad decides in between selected and saved conditions. These can be combined wherever needed.

Tip: Combining conditions allows "OR"-selections

However, note that each condition type counts as *and* selector and can't overrule the other condition type.

Every attribute can be used once per condition type.

Warning: Restrict on role basis if needed

By default, unless configured in conditions, workflow rules are evaluated for **all roles**. This also affects your customers!

Selected Conditions These conditions only match if they're active in selection. This applies for drafts (active selection) and currently saved values.

Saved Conditions These conditions only apply if they're saved within the database regardless of the current value or selection of the field.

Note: Keep in mind that the value has to be available in the situation where you need it. Otherwise the condition won't match.

35.2.4 Action

Which actions should we run on the relevant fields? The possible actions depend on the object type, however, usually you can at least change the visibility and whether the field is mandatory.

Note: Actions are not available for relations

Let's say you're working in ticket context. While you can have customer conditions, you can't adjust objects with actions in that scope.

That's because this wouldn't have any impact on the ticket dialogue. All ticket attributes (state, owner, ...) are available.

> **Warning:** Please also have a look at our *Limitations* to be safe from surprises.

Available Operators

Note: The availability of operators depends on the object type and scope.

Hint: Actions can cause confusion

Please note that actions may or may not restrict API based access to attributes. We're displaying the following icons for your overview to understand these limits better.

{API}
: This icon indicates the action affects the API.

This icon indicates the action only affects the web interface.

---

show

Display the field in question. Allows setting of values.

hide

Hide the field in question however, technically still allows setting the field.

> **Warning:** The field is **not** gone and still contains any value it provides! You may want to consider *remove* instead.

remove

Entirely removes the field. The field value will no get evaluated.

set mandatory

Sets the field to mandatory.

set optional

Sets the field to optional.

add option

Allows adding options to tree selects or selects.

Note: This requires options to be hidden beforehand (remove option). It allows to use *existing* configured values.

---

remove option

Allows removing options from tree selects or selects.

Note: It allows to use *existing* configured values.

---

set fixed to

Reduces the available options by your selection.

Tip: This may indirectly reduce your workflows in terms of *add option* and *remove option*.

---

fill in

Allows population of string and integer fields with your value.

fill in empty

Allows population of string and integer fields with your value **if the field is empty**.

select

Select a specific value within a select, tree select or boolean fields.

auto select

Helps the user on tree selects and select fields:
If the field has one option to select only and has no value yet, the value is automatically set.

Warning: This option only works if you have one value and acts passively with more options.

set readonly

Allows you to display an attribute as read only.

unset readonly

In case a workflow set the field in question to read only, you can undo this with above option.

35.2.5 Stop after match

Stop evaluation of other, following workflows that would match otherwise.

Default: `no`

35.2.6 Priority

You decide at which point your workflow is evaluated. Priorities are sorted descending – this means that a workflow matching can stop matching in specific situations.

Default: `500`

35.3 Limitations

Core Workflows does not replace Trigger Workflows manipulate behavior of fields, however, they do not set values in fields because of actions.

API calls are only partly affected Some options affect UI only and thus do not restrict responses and calls.

This affects the following actions:

- select
- auto select
- show
- hide

Some fields stay unavailable to customers For technical and security reasons, some default fields (the pale ones you can't edit) stay unavailable for display and usage on customer permissions.

Hint: If you require your customers to change e.g. priorities, please consider using workarounds via *Objects* and *Triggers*.

What is out of scope of Core Workflows…? There are some things that would count as workflow but are either done via *Triggers*, *Scheduler* or over the current top.

Such as (but not limited to):

- up- or downgrade permissions of users
- affect article creation or listing

CHAPTER

THIRTYSIX

VARIABLES

Note: Please note that this is just an overview of available variables. It might be incomplete or variables might not be available within some functions. If you're missing variables or are not sure if something is not working as expected, feel free to ask over at the Community.

Variables can be called by typing `::` (just like text modules in the frontend), as long as you're in a supported text field within the Backend. Zammad will show display all variables being available within this context and replace it to the variable as soon as you selected an entry.

Hint: You have an empty field which you referenced and it appears as `-`? That's currently working as designed - you might want to ensure that these fields always have a value (in text fields `` `` is a value!).

36.1 Variable Categories

36.1.1 Config

Note: If you're missing variables or are not sure if something is not working as expected, feel free to ask over at the Community.

Below you'll find config related variables. These hold useful configuration information that you can use within e.g. triggers to show necessary information to your customer.

The below list gives you an example what kind of data you can expect, it's not intended to explain the data itself.

Table 1: Config Variables

name	variable	example
Config > Fully Qualified Domain Name	`#{config.fqdn}`	`zammad.example.com`
Config > Ticket Hook	`#{config.ticket_hook}`	`Ticket#`
Config > HTTP type	`#{config.http_type}`	`https` or `http`
Config > SystemID	`#{config.system_id}`	`31` (value between 1 and 99)
Config > Organization	`#{config.organization}`	`Zammad GmbH` value set in Branding
Config > Product Name	`#{config.product_name}`	`Helpdesk` value set in Branding

36.1.2 Current User

Note: If you're missing variables or are not sure if something is not working as expected, feel free to ask over at the Community.

Current user variables always return values of the current user that caused e.g. a trigger to run.

Note: Due to the above fact, these variables are often not (yet) set or available for usuage.

In situations where e.g. schedulers or triggers run, this most likely is nothing you want to use.

Table 2: Current User Variables

name	variable	example
Current User > Web	`#{user.web}`	`https://zammad.org` or empty if not set
Current User > VIP	`#{user.vip}`	`false` or `true`
Current User > Updated by > Web	`#{user.updated_by.web}`	`https://zammad.org` or empty if not set
Current User > Updated by > VIP	`#{user.updated_by.vip}`	`false` or `true`
Current User > Updated by > Phone	`#{user.updated_by.phone}`	`004930123456789` or empty if not set
Current User > Updated by > Note	`#{user.updated_by.note}`	`Some note to this user` or empty if not set
Current User > Updated by > Mobile	`#{user.updated_by.mobile}`	`0049176123456789` or empty if not set
Current User > Updated by > Login	`#{user.updated_by.login}`	`jdoe`

continues on next page

Table 2 – continued from previous page

name	variable	example
Current User > Updated by > Lastname	#{user.updated_by.lastname}	Doe or empty if not set
Current User > Updated by > Firstname	#{user.updated_by.firstname}	John or empty if not set
Current User > Updated by > Fax	#{user.updated_by.fax}	004930123464789 or empty if not set
Current User > Updated by > Email	#{user.updated_by.email}	jdoe@customer.tld
Current User > Updated by > Department	#{user.updated_by.department}	Sales or empty if not set
Current User > Updated by > Address	#{user.updated_by.address}	Some street 1, 12345 Berlin or empty if not set
Current User > Updated at	#{user.updated_at}	2019-10-07 16:25:00 UTC
Current User > Phone	#{user.phone}	004930123456789 or empty if not set
Current User > Organization > Shared organization	#{user.organization.shared}	true or false
Current User > Organization > Note	#{user.organization.note}	A note to the organization of the user or empty if not set
Current User > Organization > Name	#{user.organization.name}	
Current User > Organization > Domain based assignment	#{user.organization.domain_assignment}	
Current User > Organization > Domain	#{user.organization.domain}	Zammad GmbH or empty if not set
Current User > Note	#{user.note}	Some note to this user or empty if not set
Current User > Mobile	#{user.mobile}	0049176123456789 or empty if not set
Current User > Login	#{user.login}	jdoe
Current User > Lastname	#{user.lastname}	Doe or empty if not set
Current User > Firstname	#{user.firstname}	John or empty if not set
Current User > Fax	#{user.fax}	004930123464789 or empty if not set
Current User > Email	#{user.email}	jdoe@customer.tld
Current User > Department	#{user.department}	Sales or empty if not set
Current User > Created by > Web	#{user.created_by.web}	https://zammad.org or empty if not set

continues on next page

Table 2 – continued from previous page

name	variable	example
Current User > Created by > VIP	#{user.created_by.vip}	true or false
Current User > Created by > Phone	#{user.created_by.phone}	004930123456789 or empty if not set
Current User > Created by > Note	#{user.created_by.note}	Some note to this user or empty if not set
Current User > Created by > Mobile	#{user.created_by.mobile}	0049176123456789 or empty if not set
Current User > Created by > Login	#{user.created_by.login}	jdoe
Current User > Created by > Lastname	#{user.created_by.lastname}	Doe or empty if not set
Current User > Created by > Firstname	#{user.created_by.firstname}	John or empty if not set
Current User > Created by > Fax	#{user.created_by.fax}	004930123464789 or empty if not set
Current User > Created by > Email	#{user.created_by.email}	jdoe@customer.tld
Current User > Created by > Department	#{user.created_by.department}	Sales or empty if not set
Current User > Created by > Address	#{user.created_by.address}	Some street 1, 12345 Berlin or empty if not set
Current User > Created at	#{user.created_at}	2019-10-07 16:25:00 UTC
Current User > Address	#{user.address}	Some street 1, 12345 Berlin or empty if not set

36.1.3 Articles

Note: If you're missing variables or are not sure if something is not working as expected, feel free to ask over at the Community.

Below you can find all available ticket article-based variables within Zammad. These can be called via Triggers for example. If you're unsure if Zammad does support variables at the point you're at, you can try to type `::` to check.

The below list gives you an example what kind of data you can expect, it's not intended to explain the data itself.

Table 3: Article Variables

name	variable	example
Article > Updated by > Web	#{article.updated_by.web}	https://zammad.com or empty if not set in user object
Article > Updated by > VIP	#{article.updated_by.vip}	true or false
Article > Updated by > Phone	#{article.updated_by.phone}	+4930123456789 or empty if not set in user object
Article > Updated by > Note	#{article.updated_by.note}	Some note about user or empty if not set in user object
Article > Updated by > Mobile	#{article.updated_by.mobile}	+4930123456789 or empty if not set in user object
Article > Updated by > Login	#{article.updated_by.login}	jdoe
Article > Updated by > Lastname	#{article.updated_by.lastname}	Doe or empty if not set
Article > Updated by > Firstname	#{article.updated_by.firstname}	Joe or empty if not set
Article > Updated by > Fax	#{article.updated_by.fax}	+4930123456789 or empty if not set in user object
Article > Updated by > Email	#{article.updated_by.email}	jdoe@example.com
Article > Updated by > Department	#{article.updated_by.department}	Sales or empty if not set in user object
Article > Updated by > Address	#{article.updated_by.address}	Some street 1, 12345 Berlin or empty if not set in user object
Article > Updated	#{article.updated_at}	2019-10-08 15:24:47 UTC
Article > Type > Name	#{article.type.name}	email (list of article types)
Article > To	#{article.to}	helpdesk@example.com
Article > TicketID	#{article.ticket_id}	1 (not ticket number)
Article > Subject	#{article.subject}	My amazing subject
Article > Sender > Name	#{article.sender.name}	Customer, Agent or System
Article > Visibility	#{article.internal}	false or true (false if not internal)
Article > From	#{article.from}	Joe Doe <jdoe@example.com> may differ, depends on FROM of send mail
Article > Created by > Web	#{article.created_by.web}	https://zammad.com or empty if not set in user object

continues on next page

Table 3 – continued from previous page

name	variable	example
Article > Created by > VIP	#{article.created_by.vip}	true or false
Article > Created by > Phone	#{article.created_by.phone}	+4930123456789 or empty if not set in user object
Article > Created by > Note	#{article.created_by.note}	Some note about user or empty if not set in user object
Article > Created by > Mobile	#{article.created_by.mobile}	+4930123456789 or empty if not set in user object
Article > Created by > Login	#{article.created_by.login}	jdoe
Article > Created by > Lastname	#{article.created_by.lastname}	Doe or empty if not set
Article > Created by > Firstname	#{article.created_by.firstname}	Joe or empty if not set
Article > Created by > Fax	#{article.created_by.fax}	+4930123456789 or empty if not set in user object
Article > Created by > Email	#{article.created_by.email}	jdoe@example.com
Article > Created by > Department	#{article.created_by.department}	Sales or empty if not set in user object
Article > Created by > Address	#{article.created_by.address}	Some street 1, 12345 Berlin or empty if not set in user object
Article > Created	#{article.created_at}	2019-10-08 15:24:47 UTC
Article > Cc	#{article.cc}	jdoe@example.com, company@example.com
Article > Text	#{article.body}	Test without formatting (plain)
Article Text as HTML (not referenced)	#{article.body_as_html}	Test with formatting
Ticket > Article#	#{ticket.article_count}	1 number of ticket articles

36.1.4 Ticket

Note: If you're missing variables or are not sure if something is not working as expected, feel free to ask over at the Community.

Below you can find all available ticket-based variables within Zammad. These can be called via Triggers for example. If you're unsure if Zammad does support variables at the point you're at, you can try to type `::` to check.

The below list gives you an example what kind of data you can expect, it's not intended to explain the data itself.

Table 4: Ticket Variables

name	variable	example
Ticket > Updated by > Web	`#{ticket.updated_by.web}`	`https://zammad.org` or empty if not set
Ticket > Updated by > VIP	`#{ticket.updated_by.vip}`	`false` or `true`
Ticket > Updated by > Phone	`#{ticket.updated_by.phone}`	`004930123456789` or empty if not set
Ticket > Updated by > Note	`#{ticket.updated_by.note}`	`Some note to this user` or empty if not set
Ticket > Updated by > Mobile	`#{ticket.updated_by.mobile}`	`0049176123456789` or empty if not set
Ticket > Updated by > Login	`#{ticket.updated_by.login}`	`jdoe`
Ticket > Updated by > Lastname	`#{ticket.updated_by.lastname}`	`Doe` or empty if not set
Ticket > Updated by > Firstname	`#{ticket.updated_by.firstname}`	`John` or empty if not set
Ticket > Updated by > Fax	`#{ticket.updated_by.fax}`	`004930123464789` or empty if not set
Ticket > Updated by > Email	`#{ticket.updated_by.email}`	`jdoe@customer.tld`
Ticket > Updated by > Department	`#{ticket.updated_by.department}`	`Sales` or empty if not set
Ticket > Updated by > Address	`#{ticket.updated_by.address}`	`Some street 1, 12345 Berlin` or empty if not set
Ticket > Updated at	`#{ticket.updated_at}`	`2019-10-07 16:25:00 UTC`
Ticket > Title	`#{ticket.title}`	`My amazing Subject` (normally subject, can be edited within Interface and thus differ)

continues on next page

Table 4 – continued from previous page

name	variable	example
Ticket > Accounted Time	`#{ticket.time_unit}`	`1`, `2.75` or empty response
Ticket > Tags	`#{ticket.tags}`	Currently not available, see Issue 2769
Ticket > State > Name	`#{ticket.state.name}`	`new`, `open`, ...
Ticket > Priority > Name	`#{ticket.priority.name}`	`2 normal`
Ticket > Pending till	`#{ticket.pending_time}`	`2019-10-07 16:25:00 UTC` or empty if not set
Ticket > Owner > Web	`#{ticket.owner.web}`	`https://zammad.com` or empty if not set
Ticket > Owner > VIP	`#{ticket.owner.vip}`	`false` or `true`
Ticket > Owner > Phone	`#{ticket.owner.phone}`	`004930123456789` or empty if not set
Ticket > Owner > Note	`#{ticket.owner.note}`	`Some note to this user` or empty if not set
Ticket > Owner > Mobile	`#{ticket.owner.mobile}`	`0049176123456789` or empty if not set
Ticket > Owner > Login	`#{ticket.owner.login}`	`agent`
Ticket > Owner > Lastname	`#{ticket.owner.lastname}`	`Mustermann` or empty if not set
Ticket > Owner > Firstname	`#{ticket.owner.firstname}`	`Max` or empty if not set
Ticket > Owner > Fax	`#{ticket.owner.fax}`	`004930123456789` or empty if not set
Ticket > Owner > Email	`#{ticket.owner.email}`	`agent@company.tld` or empty if not set
Ticket > Owner > Department	`#{ticket.owner.department}`	`Support` or empty if not set
Ticket > Owner > Address	`#{ticket.owner.address}`	`Some street 1, 12345 Berlin` or empty if not set
Ticket > Organization > Shared organization	`#{ticket.organization.shared}`	`false` or `true`
Ticket > Organization > Note	`#{ticket.organization.note}`	`A note to the organization of the user` or empty if not set
Ticket > Organization > Name	`#{ticket.organization.name}`	`Zammad GmbH` or empty if not set
Ticket > Organization > Domain based assignment	`#{ticket.organization.domain_assignment}`	`false` or `true`
Ticket > Organization > Domain	`#{ticket.organization.domain}`	`domain.tld` or empty if not set
Ticket > #	`#{ticket.number}`	`31001`, `201910731001`, ...

continues on next page

Table 4 – continued from previous page

name	variable	example
Ticket > Last contact (customer)	#{ticket.last_contact_customer}	2019-10-07 16:25:00 UTC or empty if not applicable yet (Please note Ticket last contact behaviour Settings for this)
Ticket > Last contact	#{ticket.last_contact_at}	2019-10-07 16:25:00 UTC
Ticket > Last contact (agent)	#{ticket.last_contact_agent_at}	2019-10-07 16:25:00 UTC or empty if not applicable yet
Ticket > Group > Note	#{ticket.group.note}	Note about this group
Ticket > Group > Name	#{ticket.group.name}	Sales
Ticket > Group > Follow-up possible	#{ticket.group.follow_up_possible}	no or yes
Ticket > Group > Assign Follow-Ups	#{ticket.group.follow_up_assignment}	false or true
Ticket > Group > Assignment Timeout	#{ticket.group.assignment_timeout}	20 or empty if not configured
Ticket > First response	#{ticket.first_response_at}	2019-10-07 16:25:00 UTC or empty if not applicable yet
Ticket > Escalation at	#{ticket.escalation_at}	2019-10-07 16:25:00 UTC or empty if not applicable
Ticket > Customer > Web	#{ticket.customer.web}	https://zammad.org or empty if not set
Ticket > Customer > VIP	#{ticket.customer.vip}	false or true
Ticket > Customer > Phone	#{ticket.customer.phone}	004930123456789 or empty if not set
Ticket > Customer > Note	#{ticket.customer.note}	Some note to this user or empty if not set
Ticket > Customer > Mobile	#{ticket.customer.mobile}	0049176123456789 or empty if not set
Ticket > Customer > Login	#{ticket.customer.login}	jdoe
Ticket > Customer > Lastname	#{ticket.customer.lastname}	Doe or empty if not set
Ticket > Customer > Firstname	#{ticket.customer.firstname}	Joe or empty if not set
Ticket > Customer > Fax	#{ticket.customer.fax}	004930123456789 or empty if not set
Ticket > Customer > Email	#{ticket.customer.email}	jdoe@customer.tld
Ticket > Customer > Department	#{ticket.customer.department}	Sales or empty if not set
Ticket > Customer > Address	#{ticket.customer.address}	Some street 1, 12345 Berlin or empty if not set

continues on next page

Table 4 – continued from previous page

name	variable	example
Ticket > Created by > Web	`#{ticket.created_by.web}`	`https://zammad.org` or empty if not set
Ticket > Created by > VIP	`#{ticket.created_by.vip}`	`false` or `true`
Ticket > Created by > Phone	`#{ticket.created_by.phone}`	`004930123456789` or empty if not set
Ticket > Created by > Note	`#{ticket.created_by.note}`	`Some note to this user` or empty if not set
Ticket > Created by > Mobile	`#{ticket.created_by.mobile}`	`0049176123456789` or empty if not set
Ticket > Created by > Login	`#{ticket.created_by.login}`	`jdoe`
Ticket > Created by > Lastname	`#{ticket.created_by.lastname}`	`Doe` or empty if not set
Ticket > Created by > Firstname	`#{ticket.created_by.firstname}`	`Joe` or empty if not set
Ticket > Created by > Fax	`#{ticket.created_by.fax}`	`004930123456789` or empty if not set
Ticket > Created by > Email	`#{ticket.created_by.email}`	`jdoe@customer.tld`
Ticket > Created by > Department	`#{ticket.created_by.department}`	`Sales` or empty if not set
Ticket > Created by > Address	`#{ticket.created_by.address}`	`Some street 1, 12345 Berlin` or empty if not set
Ticket > Created at	`#{ticket.created_at}`	2019-10-07 16:25:00 UTC
Ticket > Closing time	`#{ticket.close_at}`	2019-10-07 17:25:00 UTC
Ticket > Article#	`#{ticket.article_count}`	`3`, any number of articles existing in the ticket right now

36.2 Wait, what about custom objects?

Good point! Of course, we can't predict what objects you might create, but we can give you a hint on how to put the puzzle together to get your custom values.

For this, we'll talk about the inner part of the variable (so everything within `#{}`), please put the outer part around to have a valid variable. The first part of the variable name consist of the object type. Currently these are:

- Ticket (`ticket`)
- User (`user`)

- Organization (`organization`)
- Group (`group`)

The second part is the name of the object. This is the name you define during object creation and can be found within the object menu at any time. The first and second part of a variable is divided by a point, e.g.: `ticket.number` which will in total be `#{ticket.number}`.

Now, in some situations it's possible that you'll need to use a third part. The best example for such a situation would be a `select` or `tree-select` field which will by default return the key value, not it's display name. For this, just extend your variable with `.value`. This will result in `#{ticket.select.value}`.

CHAPTER

THIRTYSEVEN

TRANSLATIONS

Translations of Zammad are processed centrally on our Weblate instance. This allows non development people to review and update translation strings of the language they actually speak.

Your language is in an incomplete translation state or has typos? Please consider helping us sorting this out! All you need is either a GitHub account or register directly on our instance.

Hint:

Did you know?
This is also where documentation translations are handled.

Apart from these centralized translations, there's still local translations you can manage. This is relevant if you added custom objects that require translations or even states.

Warning: You can no longer synchronize these translations upstream like prior Zammad 5.1.

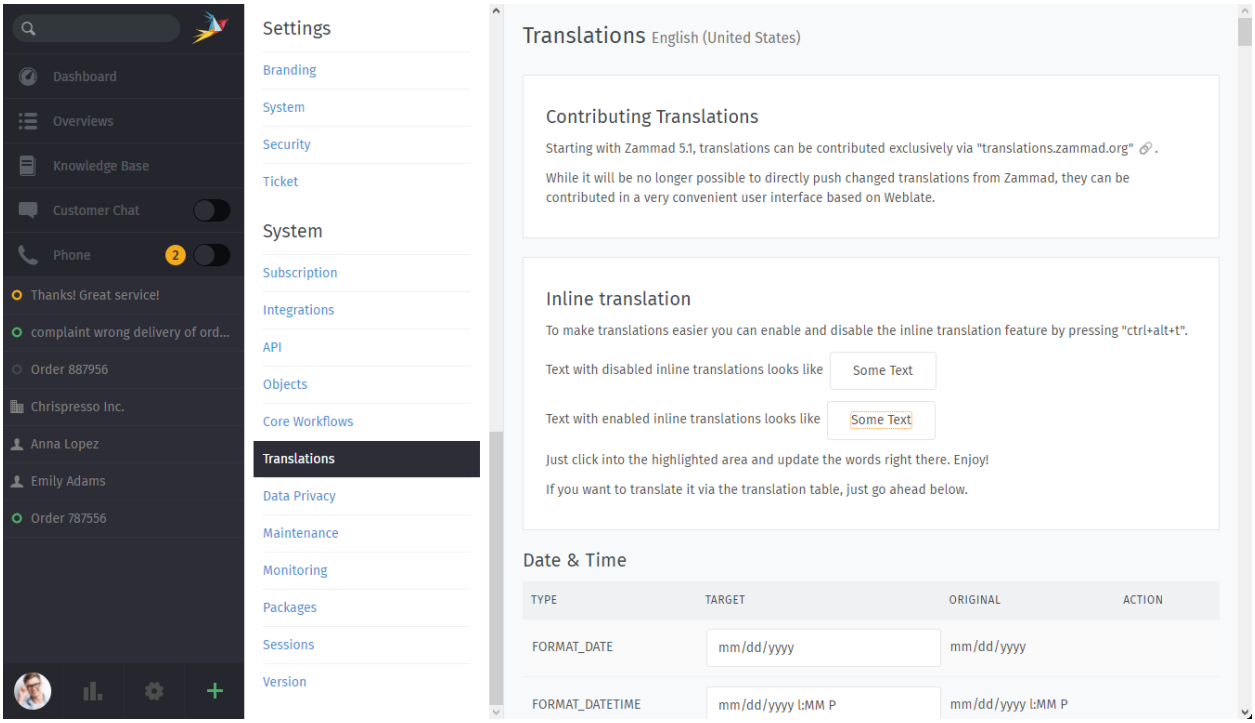
37.1 How do I get the most recent translations?

Because of how Zammad ships translations, the only way to update the central translations is to update your Zammad installation. The benefit of this is that you no longer need internet access during a Zammad upgrade as the package has everything it requires already.

37.2 But... I have custom objects?!

No problem at all! If you can't find the strings of your objects already within the translation list, please ensure to add the translation strings to your installation.

Now you can translate your objects as needed.



37.3 So how does this local translation work?

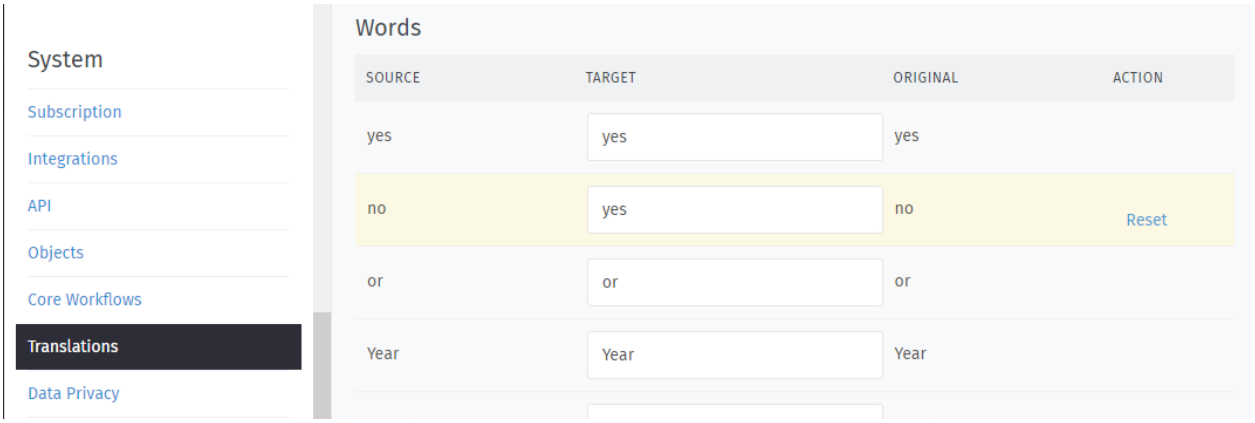
Within the translation menu, look up the string you're searching for and update its target as designed. As soon as you leave the target input field, the change will be saved.

Such locally translated strings are slightly highlighted and come with a "Reset" action.

The translation strings shown by Zammad are always those of the profile language you've chosen. If you need to translate a different language, change the language in your profile settings up front.

Hint: Due to caching you may have to reload your browser session to see the changes.

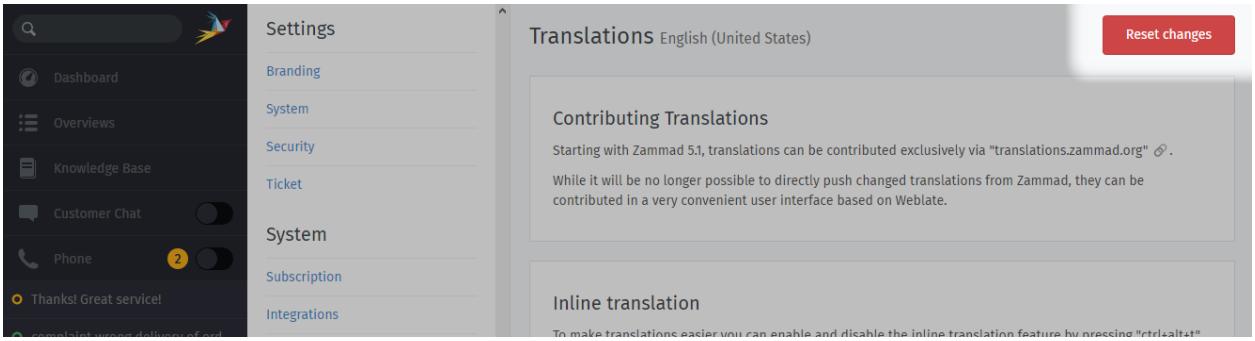
Yes, these changes are update safe!



37.3.1 Reverting to original translations

If you want to reset all translation changes on your instance, use the red “Reset” button on the upper right of the translation management.

In case you just intend to reset a specific translation, lookup the translation string and use the “reset” action.



CHAPTER

THIRTYEIGHT

DATA PRIVACY

For compliance with GDPR and other data privacy laws, you may wish to **permanently delete users from the system**, along with **all of their associated tickets**.

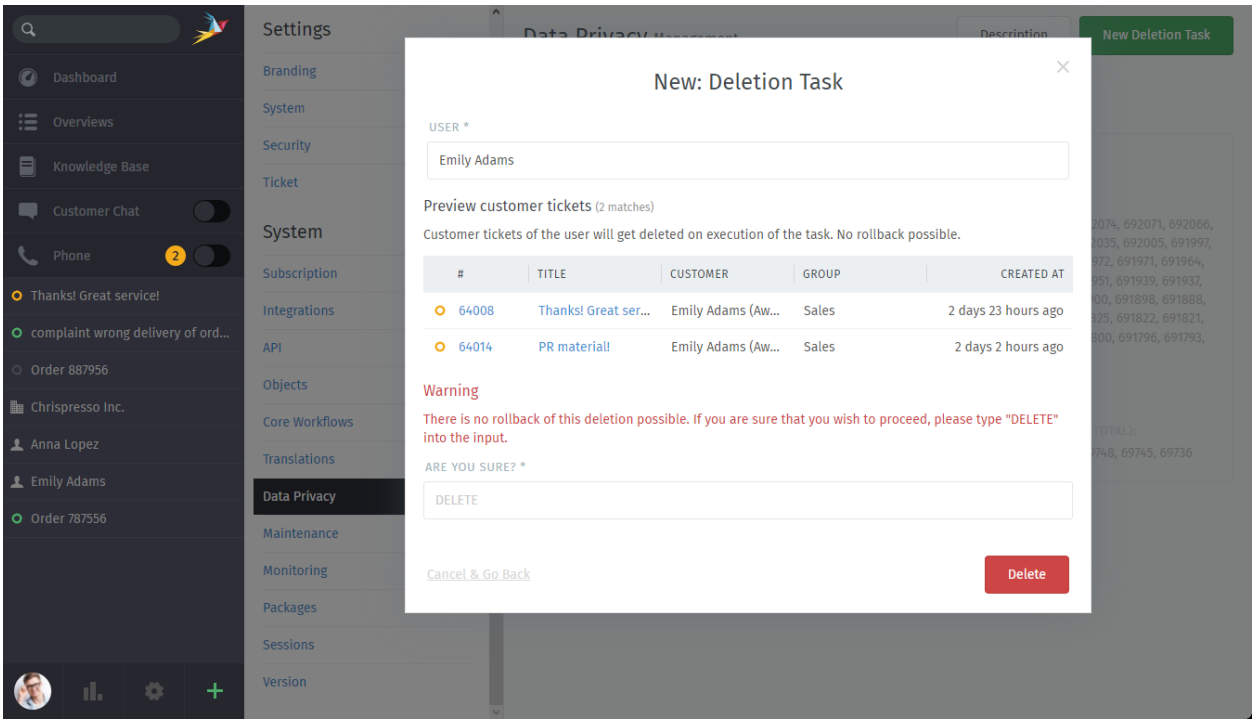


Fig. 1: The user deletion dialog lists some of the tickets that will be removed from the system along with the user.

Note: Huh? I don't see the Data Privacy panel…

Access to this panel requires `admin.data_privacy` permissions (introduced in Zammad 3.5).

On older systems that have not been updated yet, customers can also be deleted via the Zammad console.

38.1 Deleting Users

Warning: All deletions are FINAL!

Once you click “Delete”, the action cannot be cancelled or undone.

Any time you delete a user, all their tickets will be deleted, as well. It is not possible to delete a user and still keep their tickets.

Note: The following records cannot be deleted:

- your own account
- the system’s last remaining administrator account

38.1.1 Step 1: Find a user / confirm deletion

There are three ways to access the user deletion dialog:

from the user’s profile

Fig. 2: Click **Action > Delete**.

in the “Manage > Users” Admin Panel

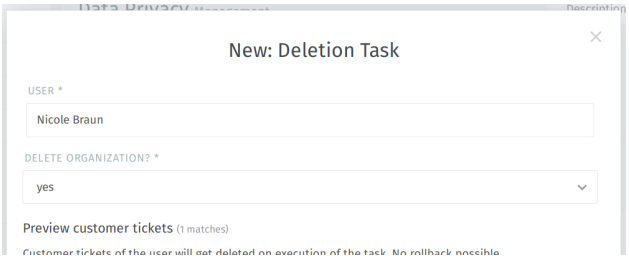
Fig. 3: Use the **Actions** menu for the target user.

in the “System > Data Privacy” Admin Panel

Fig. 4: Use the **New Deletion Task** button. Search for users by name or email address.

Hint: You can delete organizations, too.

If the customer you are deleting is the last user in their organization, a **Delete Organization?** option will be displayed in the user deletion dialog:

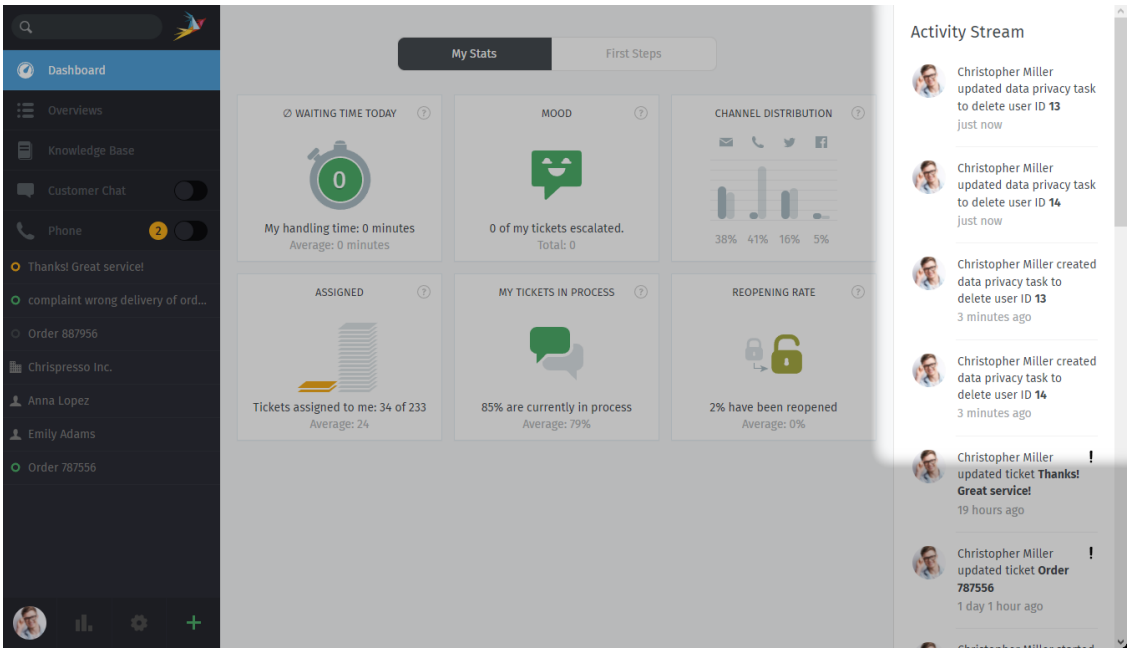


(If this option does not appear, make sure there are no pending deletion tasks for other customers from this organization.)

38.1.2 Step 2: Monitor deletion job status

It may take up to ten minutes for the system to process your request, so for each user you delete, a "deletion task" is added to the queue. You can keep an eye on the status of these tasks in two places:

in the Activity Stream



For each deleted user, the Activity Stream will be updated twice—once when the task is created, and once when it's complete.

Hint: These notifications are only visible to users with `admin.data_privacy` permissions.

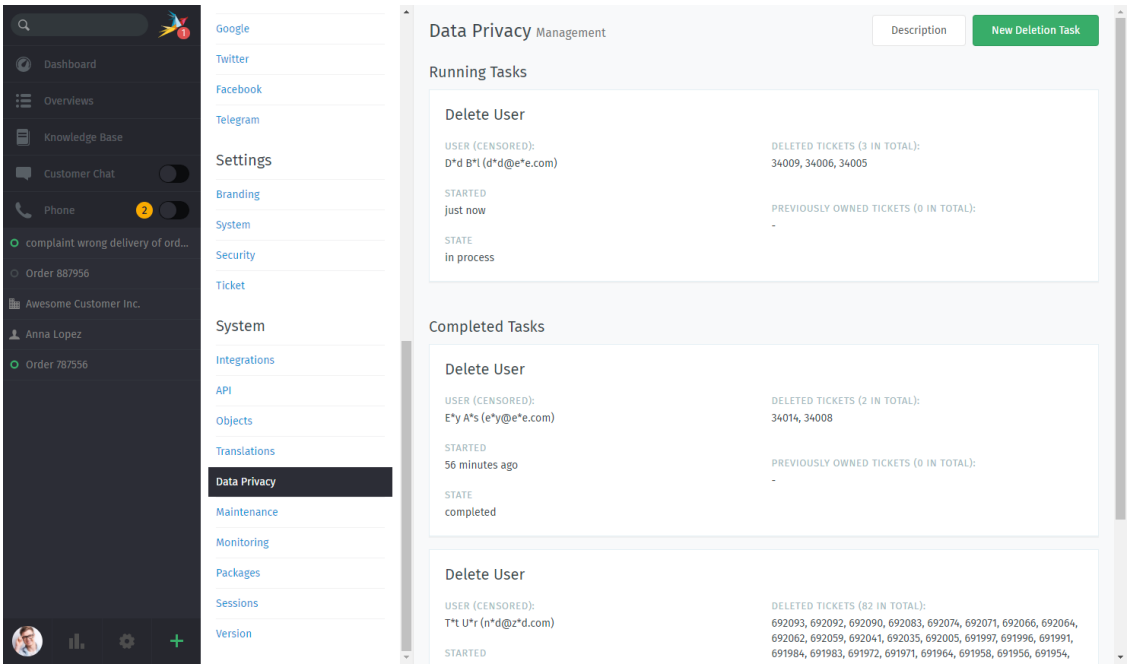
in the "System > Data Privacy" Admin Panel

38.2 Frequently Asked Questions

What happens if I receive an email from a deleted customer? Zammad automatically creates a new user account whenever it receives a message from an unrecognized email address, including deleted users. Deleted users are never blocked from creating new tickets.

In the unlikely event that you receive an email between the time that you click "Delete" and the system has processed your request, **that ticket will be automatically removed**. The ticket number for the lost ticket will be displayed in the Admin Panel under **System > Data Privacy > Completed Tasks > Delete User > Deleted Tickets**.

What about user information stored in internal notes or other messages? The deletion process removes **user accounts and associated tickets only**.



If there are references to a user's name or information stored elsewhere in the system, that information **will not be removed** because there is no way to safely determine if it actually describes the user in question.

I deleted an user and can still see a message they sent! Tickets can only belong to a single customer, but may contain messages ("articles") from many people. If you deleted a user but you're still seeing articles they sent, don't worry—those articles are for a ticket that belongs to someone else, and no longer contain any reference to the sender's identity.

I removed a customer, now my reporting is off! When removing users and their tickets, all references are removed. This also affects e.g. Reporting - these information are lost.

CHAPTER

THIRTYNINE

MAINTENANCE

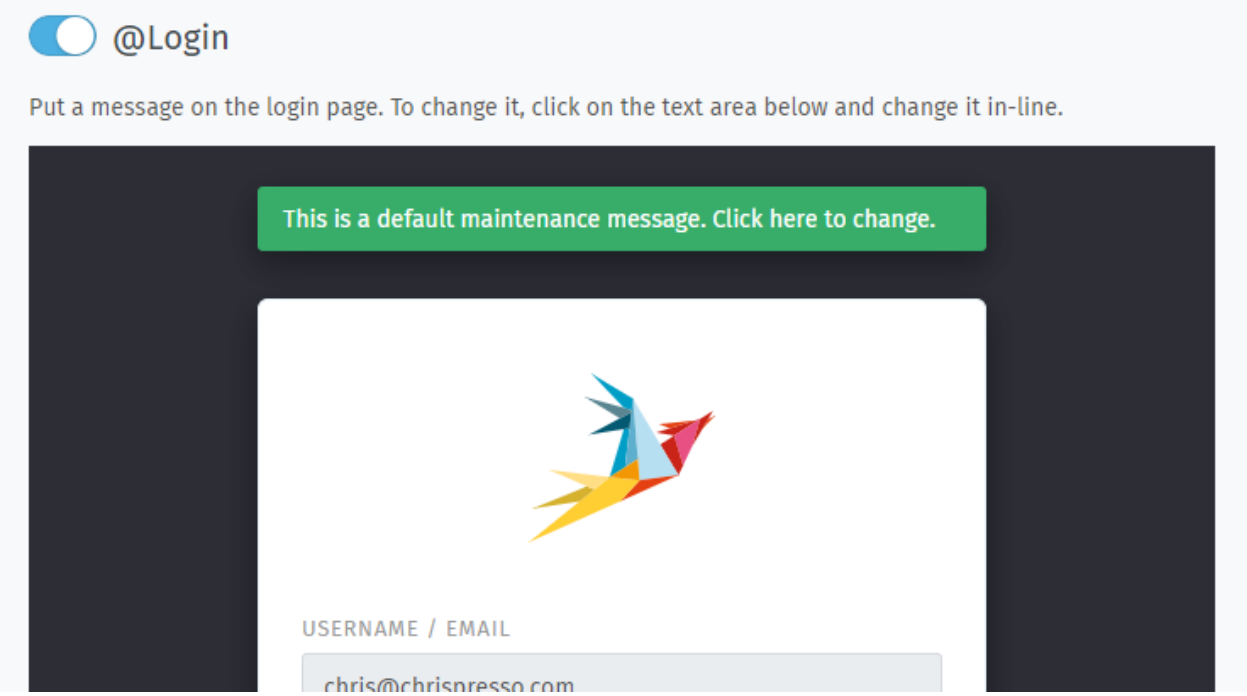
Zammad comes with a maintenance mode that you can use for e.g. updating the instance or changing settings while restricting availability and functions.

Mode The mode setting allows you to enable or disable the maintenance mode.

Defaults to *off*.

The maintenance mode will restrict access to administrative roles only. This means agents and customers are logged off.

@Login This setting allows you to provide a login message within a green banner above the login screen. Click into the green banner in the settings page to adjust your message. To activate the message, activate the *@Login* setting.



Message Send an informative message to **all active sessions**. This comes handy to inform your agents e.g. about mayor outages or to force reloads after configuration changes.

Title This is the messages title (slightly bigger than the normal message).

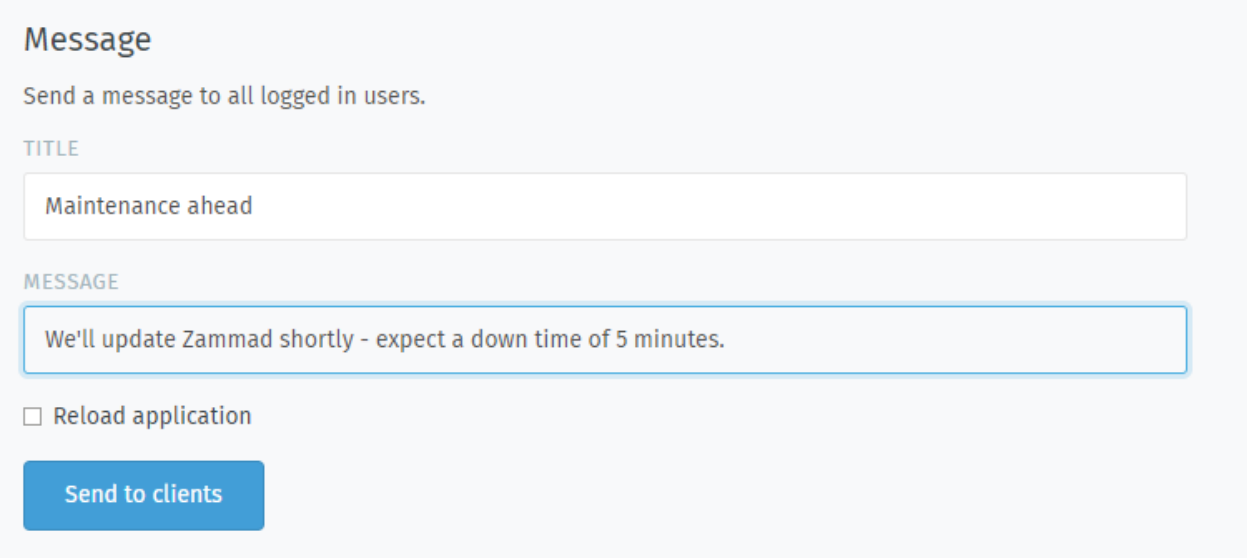
Message The text you want to provide to your logged in sessions.

Reload application Selecting this option will change the message acceptance button from *Close* (with nothing happening) to *Continue session*, which forces the application to reload.

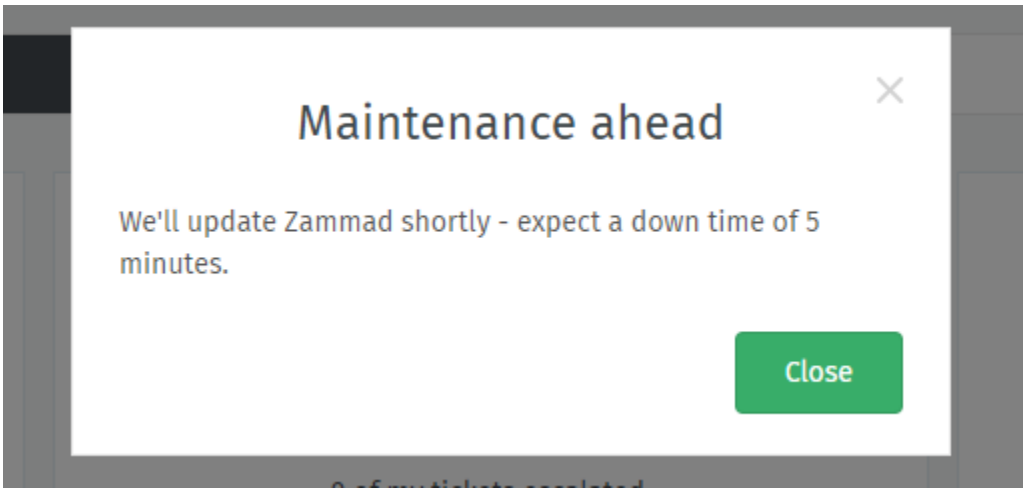
> **Warning:** If you have customers that are logged in to Zammad, they'll also be notified if they're active in that moment.

Example without reload application

Message setting within Zammad's admin settings without ticket reload application setting.



The modal all other active sessions will see upon pressing *Send to clients*.



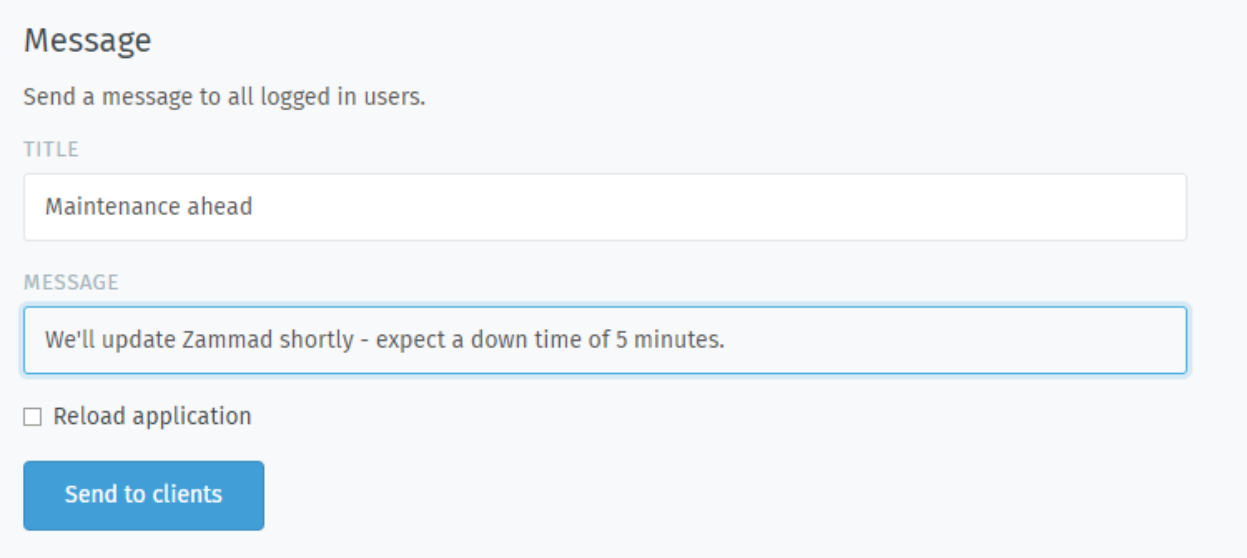
X

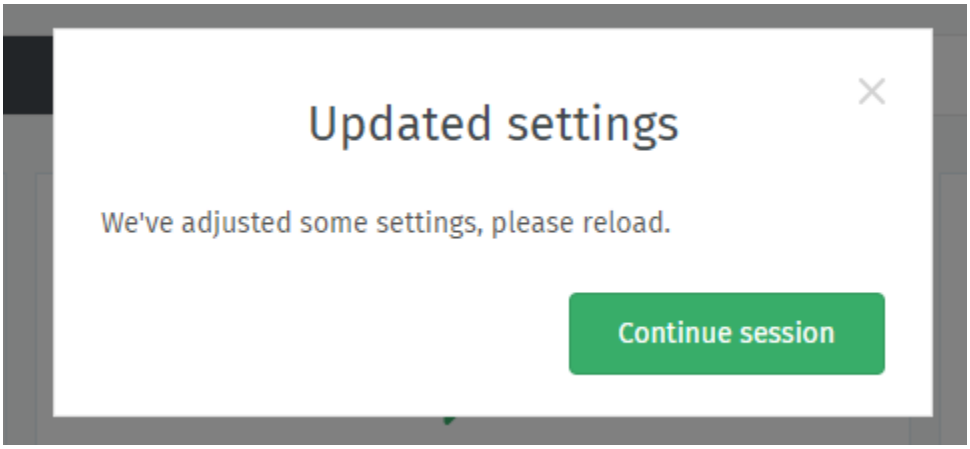
Example with reload application

Message setting within Zammad's admin settings with ticket reload application setting.

The modal all other active sessions will see upon pressing *Send to clients*.

X


Message
Send a message to all logged in users.
TITLE
Maintenance ahead
MESSAGE
We'll update Zammad shortly - expect a down time of 5 minutes.
Reload application
Send to clients


Updated settings
We've adjusted some settings, please reload.
Continue session

CHAPTER

FORTY

MONITORING

Please note: This is only available in self hosted instances, as we're monitoring hosted instances and fix problems.

On the monitoring page you can see the current health state of Zammad. This can be useful if you for example have the feeling that you don't receive emails anymore, you can take a look here before logging onto your Server.

Besides the optical state of an event, you can also reset the access token for this module and get the monitoring URL for a monitoring system of your choice.

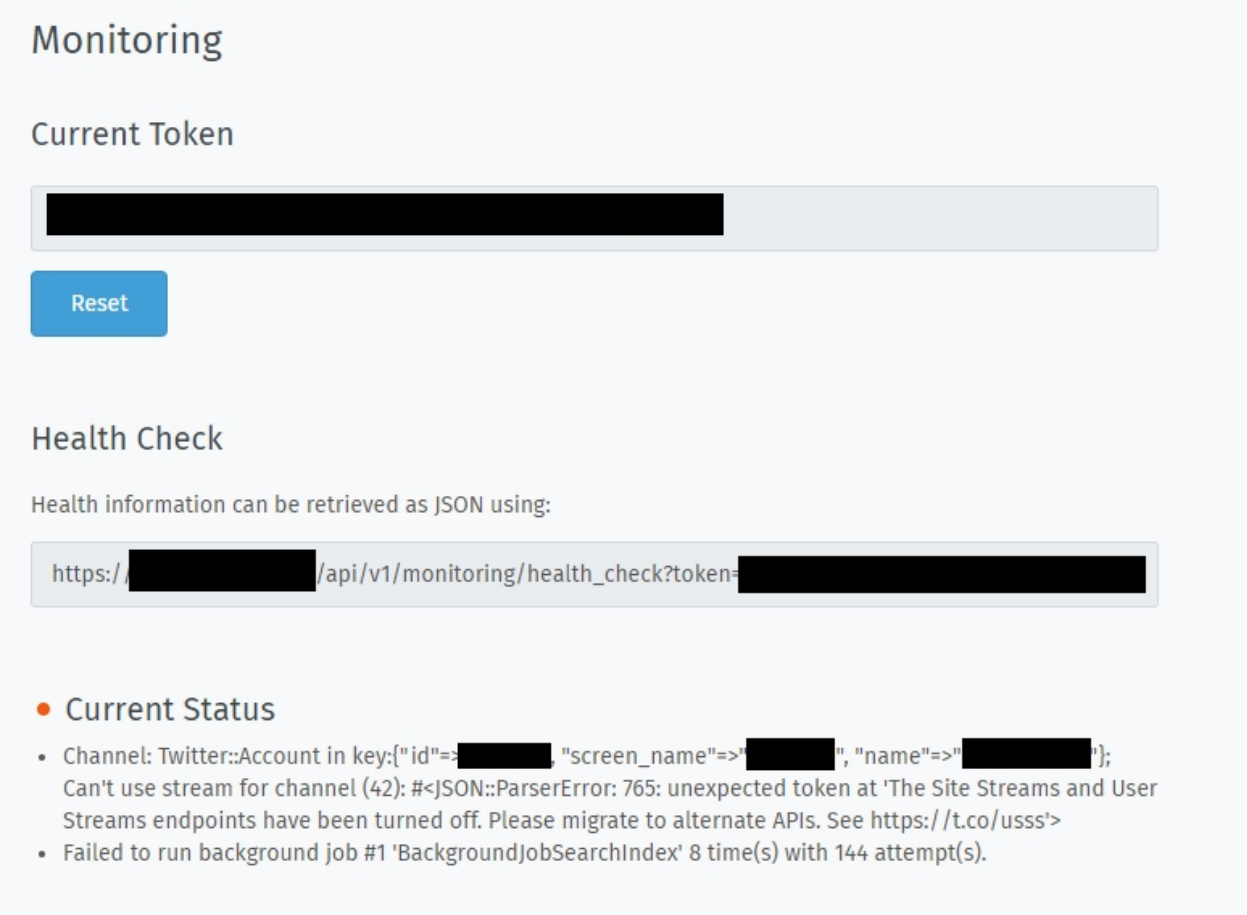
Example output to this can be:

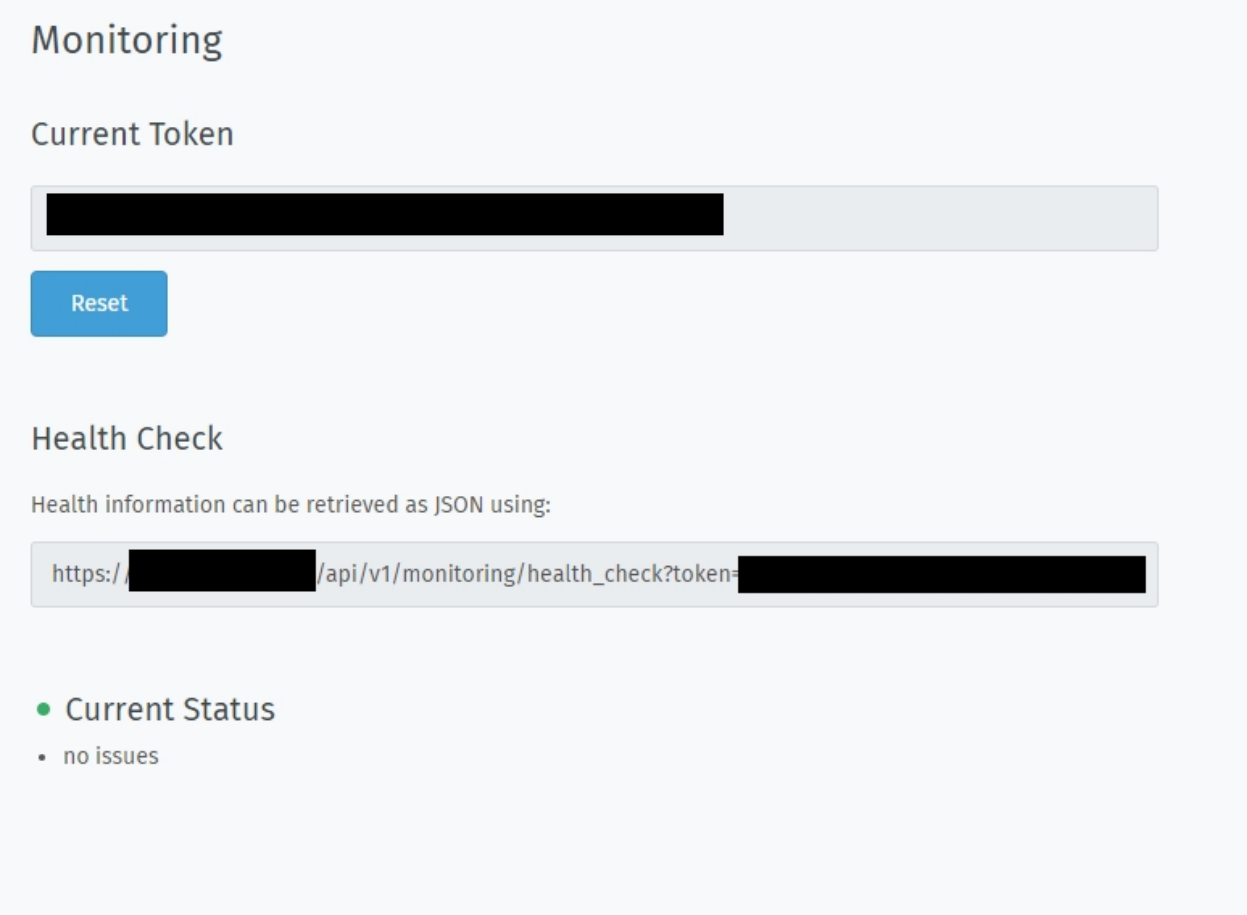
Everything is OK (refer to image 2 for interface example):

```
{"healthy"=>true, "message"=>"success", "token"=>"2432XXXXXXXXXXXXXXXXXXXXX1761"}
```

Zammad has issue (whatever nature, refer to image 2 for interface example):

```
{"healthy":false,"message":"Channel: Twitter::Account in key:{\"id\"=\XXXXXXXXXXXXXXXX,␣
↪\"screen_name\"=\u003e\"Name\", \"name\"=\u003e\"Somewhat name\"}; Can't use stream␣
↪for channel (42): #\u003cJSON::ParserError: 765: unexpected token at 'The Site Streams␣
↪and User Streams endpoints have been turned off. Please migrate to alternate APIs. See␣
↪https://t.co/usss'\u003e","issues":["Channel: Twitter::Account in key:{\"id\"=\XXXXXXX,
↪ \"screen_name\"=\u003e\"Name\", \"name\"=\u003e\"Somename\"}; Can't use stream for␣
↪channel (42): #\u003cJSON::ParserError: 765: unexpected token at 'The Site Streams and␣
↪User Streams endpoints have been turned off. Please migrate to alternate APIs. See␣
↪https://t.co/usss'\u003e"],"actions":[],"token":"OgitXXXXXXXXXXXXXXXXXXXXXNxo4ptCoQ"}
```


Monitoring
Current Token
[redacted]
Reset
Health Check
Health information can be retrieved as JSON using:
https:/[redacted]/api/v1/monitoring/health_check?token=[redacted]
Current Status
Channel: Twitter::Account in key:{"id"=>[redacted], "screen_name"=>"[redacted]", "name"=>"[redacted]"};
Can't use stream for channel (42): #<JSON::ParserError: 765: unexpected token at 'The Site Streams and User Streams endpoints have been turned off. Please migrate to alternate APIs. See https://t.co/usss'>
Failed to run background job #1 'BackgroundJobSearchIndex' 8 time(s) with 144 attempt(s).


Monitoring
Current Token
[redacted]
Reset
Health Check
Health information can be retrieved as JSON using:
https:/[redacted]/api/v1/monitoring/health_check?token=[redacted]
Current Status
no issues

CHAPTER

FORTYONE

PACKAGES

That’s the package management-area.

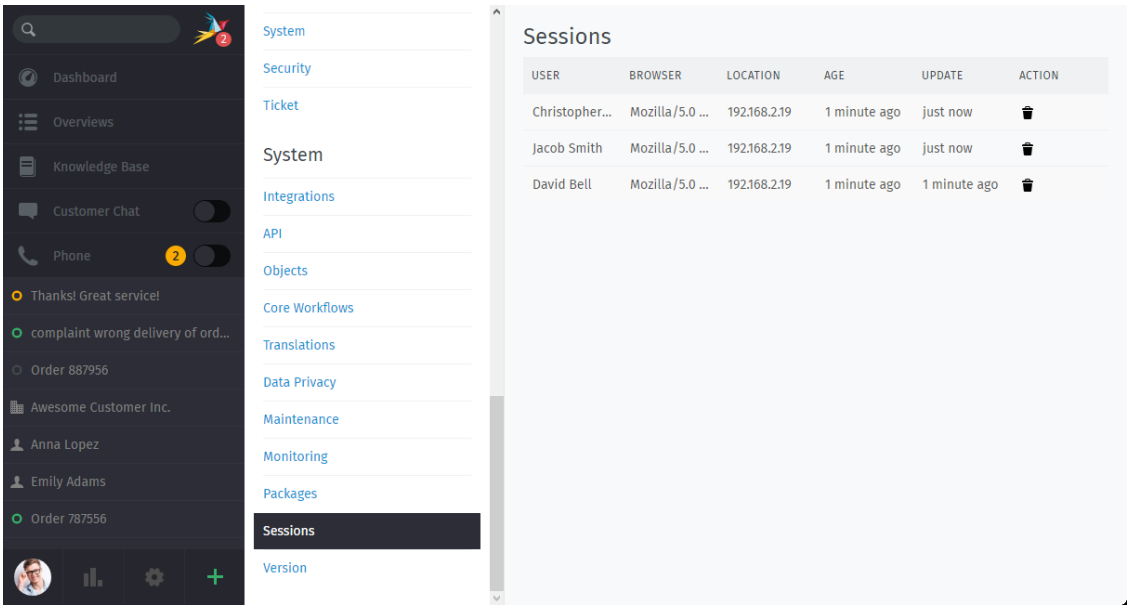
Individual add-ons for Zammad can be installed and managed here.

CHAPTER

FORTYTWO

SESSIONS

Sessions management allows reviewing currently known user sessions and allows to end these.



Hint: This page indirectly is affected by *Session Timeout* configurations from security settings.

Zammad will provide the following information:

User The user account this session entry belongs to. It can be normal if a user has several sessions in the list. This can be due to changed browser information or if you use e.g. single sign on methods where the user does not use the log off button.

Browser The browser agent communicated to Zammad.

Location The anticipated location based on the users IP address. Depending on the IP address the result shown may differ. This option depends on *Services*.

The following information can be shown:

IP address Either if you set *Geo IP services* to inactive or you're using internal IP address which do not provide location information.

Country (or Country + City) If *Geo IP services* is enabled only. Depends on how detailed the available IP address information are.

> **Note:** Depending on how long the address is assigned to a specific country the result may differ. Results can be inaccurate - this technically is not an error.

Age Provides the time when the session initially has been created.

Update Provides the time the user last used this session to open Zammad. This timestamp is only updated if the user e.g. reloads, not during normal work on tickets.

> **Hint:** This timestamp is being used for the session timeout.

Action Use the delete button to remove a single session on behalf of your user.

CHAPTER

FORTYTHREE

VERSION

Shows which version is currently being used on your Zammad-instance.

CHAPTER

FORTYFOUR

COMPOSER SETTINGS

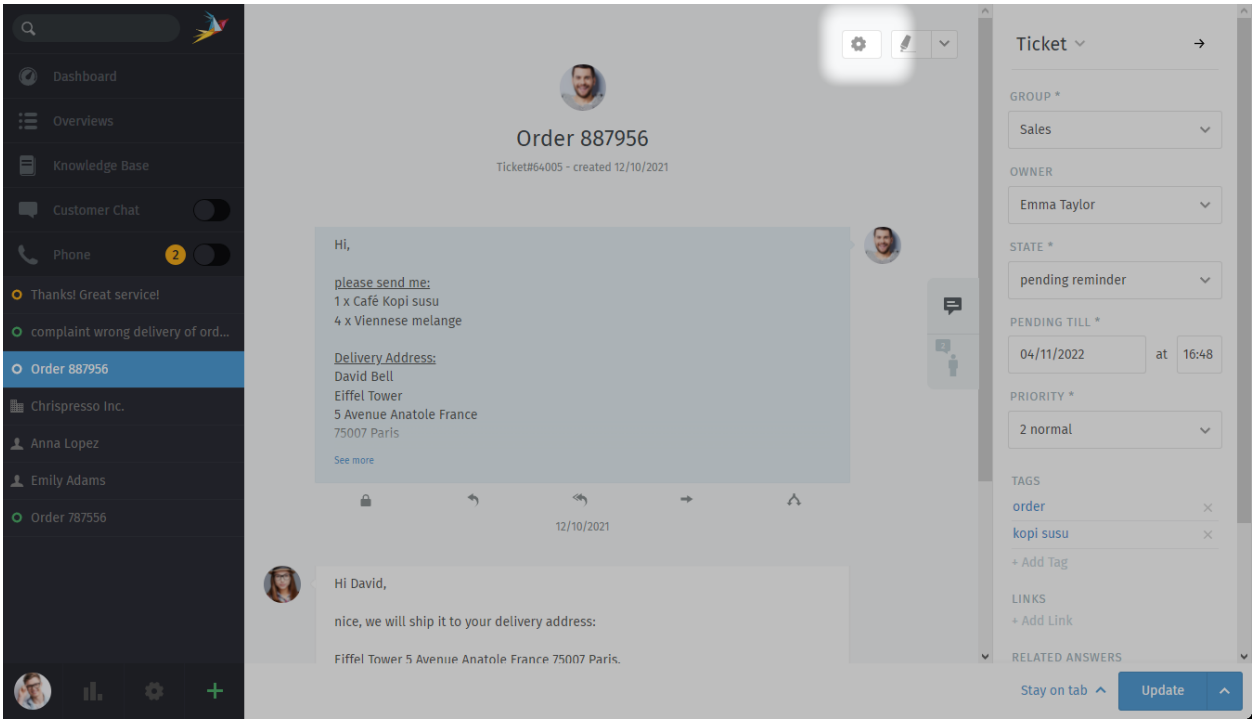
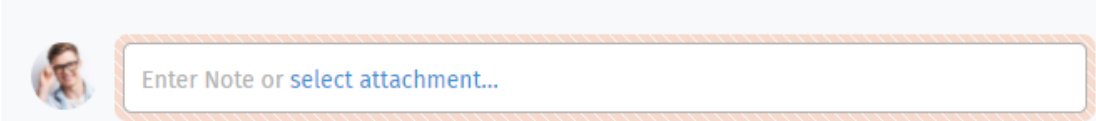


Fig. 1: Use the Composer Settings to change the behavior of the new message editor.

Note: These settings **apply on all tickets and to all users across the entire system**.

Note - default visibility (default: `internal`) This setting decides what the default visbility of note articles is. This affects **only** notes (default article on ticket answering). The visibility of phone- and email article notes is not affected.



Article - visibility confirmation dialog This setting allows administrators to enforce a confirmation dialog to agents when they're trying to change the article visibility from internal to public.

This setting affects all article types within UI, no matter if the article exists already or it's still a writing draft.

Email - subject field (default: `no`) When setting this option to `yes`, Zammad will also display the subject field when answering via email articles. It doesn't matter if you click on `reply` or switch to email article manually.

> **Warning:** Please note that if set to `no`, Zammad will automatically use the tickets title as subject!
>
> The subject can differ between title and mail subject if choosing `yes`.

Email - full quote (default: `no`) Setting this option to `yes` will always add the content of the answered article as quotation below your signature.

Note: This does not affect the "mark and quote" functionality, if you mark a text with this setting enabled, we'll use the marked text as quote instead.

Email - quote header (default: `yes`) If you don't want Zammad to add the date, time and name or the article you're quoting, set this to `no`.

Example: `On Thursday, June 27, 2019, 3:37:11 PM, Jacob Smith wrote:`

Twitter - tweet initials (default: `yes`) When set to yes, this will add `/CM` (first character of first- and last name) to the bottom of every tweet answer you create. This only affects tickets that come from the Twitter Channel.