Zammad

20.05.2021

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 . . . 6
- 1.5 Reference Guide: User Details . . . 8

2 Groups **13**
- 2.1 Group Settings . . . 13
- 2.2 Group Access Levels . . . 14

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 Import of organizations via CSV file . . . 32

5 Overviews **35**

6 Text Templates **41**
- 6.1 Text modules . . . 41
- 6.2 Delete or clone text modules . . . 44
- 6.3 Import of text modules via CSV file . . . 44
- 6.4 Ticket templates for new tickets . . . 44

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

8 Tags **49**
- 8.1 How to tag . . . 49
- 8.2 Tag-Overviews . . . 49
- 8.3 Tag Management Area . . . 52

9 Calendars **53**

10 SLAs **57**

11 Triggers **61**
11.1 Learn by example . . . 61
11.2 How do they work? . . . 65
11.3 Webhooks . . . 67
11.4 Limitations . . . 73
11.5 System Notifications . . . 74

12 Scheduler **75**
12.1 Add a new scheduler . . . 76

13 Report-Profiles **79**
13.1 Further information about the reporting: . . . 80

14 Time Accounting **83**

15 Knowledge Base **87**
15.1 Features . . . 88
15.2 Setup . . . 88

16 Web **93**

17 Forms **95**
17.1 Potential Spam-Issue . . . 98

18 Email **99**
18.1 Accounts . . . 99
18.2 Filters . . . 107
18.3 Signatures . . . 109
18.4 Settings . . . 112
18.5 Extra Options for Self-Hosted Users . . . 116

19 Chat **121**
19.1 How to improve Support by Chat? . . . 121
19.2 Our answer: The Zammad Chat Widget . . . 121
19.3 Configuration of the Chat widget . . . 122

20 Google **127**
20.1 Accounts . . . 127
20.2 Filters . . . 136
20.3 Signatures . . . 137
20.4 Settings . . . 140

21 Microsoft 365 **145**
21.1 Accounts . . . 145
21.2 Filters . . . 153
21.3 Signatures . . . 154
21.4 Settings . . . 157

22 Twitter **161**
22.1 1. Apply for a Twitter Developer account . . . 162
22.2 2. Create a new Twitter app for Zammad . . . 163
22.3 3. Set your new app's permissions . . . 163
22.4 4. Generate a new access token & secret . . . 166
22.5 5. Set up a dev environment . . . 166

22.6 6. Add your new Twitter app in Zammad . . . 168
22.7 7. Add your Twitter account in Zammad . . . 168
22.8 8. Configure filters . . . 168

23 Facebook **171**

24 Telegram **185**
24.1 Register a Telegram bot app . . . 185
24.2 Configure Zammad as Telegram bot . . . 187
24.3 Start using your new channel . . . 188

25 Branding **195**

26 System **197**
26.1 Service . . . 197
26.2 Frontend . . . 197

27 Security **199**
27.1 Base . . . 199
27.2 Password . . . 199
27.3 Third-Party Applications . . . 200

28 Ticket **223**
28.1 Base . . . 223
28.2 Number . . . 224
28.3 Auto-Assignment . . . 225

29 Integrations **227**
29.1 Integrations for phone systems . . . 227
29.2 Integrations for authentication and customers . . . 234
29.3 Integrations for Monitoring Systems . . . 241
29.4 Other Integrations . . . 250

30 Objects **259**
30.1 Object types . . . 260
30.2 URL fields (Link-Template) . . . 261
30.3 Object permissions . . . 261
30.4 Updating database after adding or editing objects . . . 263

31 Variables **265**
31.1 Variable Categories . . . 265
31.2 Wait, what about custom objects? . . . 275

32 Translations **277**

33 Data Privacy **279**
33.1 Deleting Users . . . 280
33.2 Frequently Asked Questions . . . 281

34 Maintenance **283**
34.1 Mode . . . 283
34.2 @Login . . . 283

35 Monitoring **287**

36 Packages **291**

37 Sessions 293

38 Version 295

39 Composer Settings 297

KAPITEL 1

Users

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

Abb. 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

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

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

Deleting users

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

Warnung: Deleting a customer destroys all their associated tickets!

To learn more, see *Data Privacy*.

Filtering the user list

Abb. 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.

Bemerkung: 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.

Taking over a user's session

Abb. 5: 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.

Gefahr: 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.)

Hinweis: 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.

Hinweis: 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*.

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.

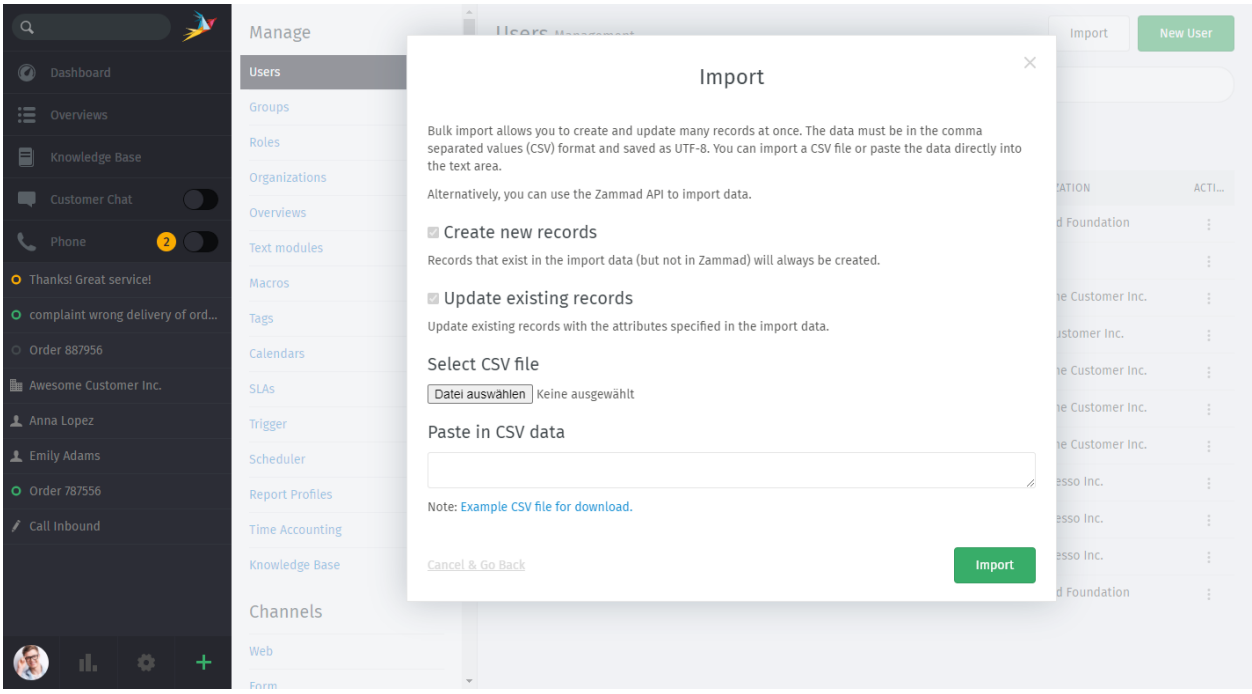


Abb. 6: Use the **Import** button to open the CSV import dialog.

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 others 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:

Bemerkung: 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!

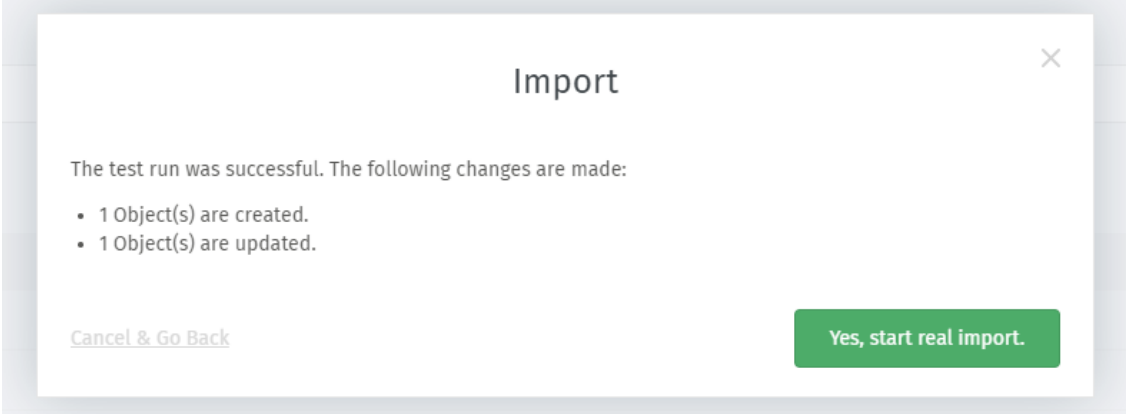


Abb. 7: CSV import always begins with a preview / test run.

1.3 LDAP / Active Directory

With our LDAP integration, you can easily use existing authentication systems without having to update more than one source. Also, e.g. password policies are ensured by your LDAP source - Zammad will always contact your LDAP server for authentications.

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

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

On this wizard step, you can also define the wanted LDAP-group-to-Zammad-role mapping.

Bemerkung: Please note that nested groups are currently not supported by Zammad.

If needed, you can also change the user filter for your LDAP query. The option „Users without assigned LDAP groups“ will by default assign the *customer* role (default sign-up role) to any LDAP user, that doesn't match to the above role mapping. After pressing Continue, Zammad will check if the configuration is okay. You can then enable LDAP and start your first sync. The sync will then run hourly - if you need to change mappings, you can change the configuration at any time.

LDAP Mapping

User

LDAP ATTRIBUTE	ZAMMAD ATTRIBUTE	ACTION
givenname (e.g., All)	Firstname	Remove
sn (e.g., Exudos)	Lastname	Remove
samaccountname (e.g., Admi	Login	Remove
		Remove

+ Add

Roles

LDAP GROUP	ZAMMAD ROLE	ACTION
cn=zammad-group,ou=zamm	Agent	Remove

+ Add

Expert

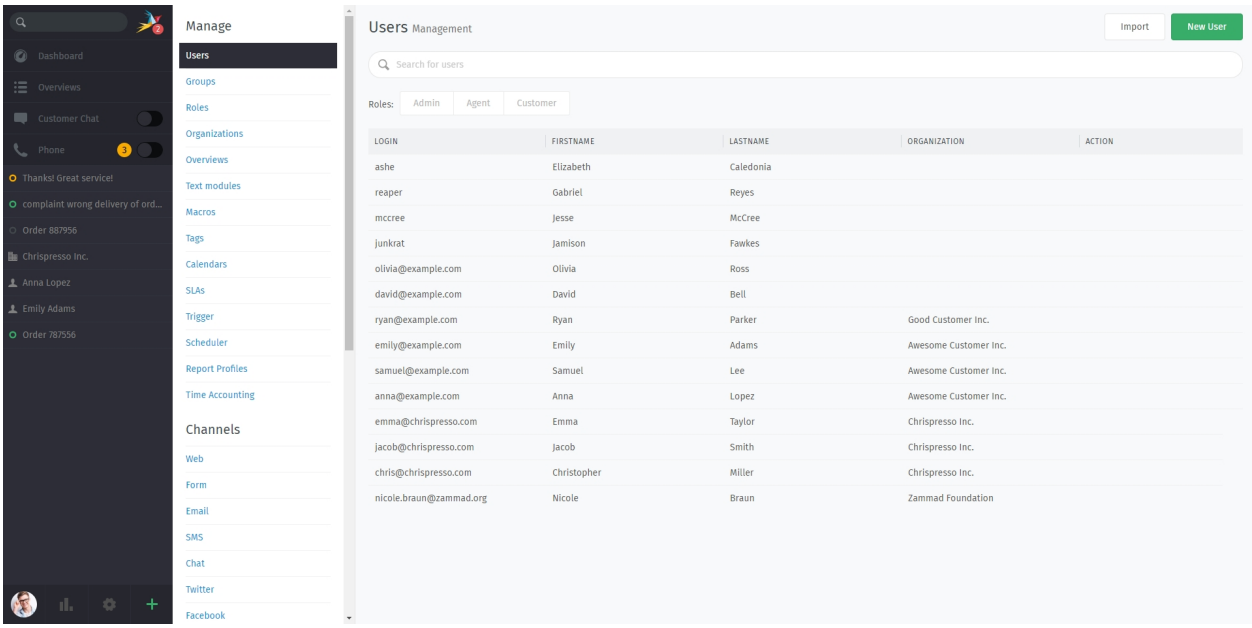
NAME	VALUE
User filter	(&(objectClass=user)(samaccountname=*)(!(samaccountname=
Users without assigned L DAP groups	Assign signup roles

Go Back

Continue

Bemerkung: You can use user filters to limit the search results Zammad gets. Also, you can hide deactivated Active Directory accounts (the users will be set to inactive within Zammad). You can find further information for Active Directory Filters at the Website from Microsoft.

After the sync has finished, you can find the new LDAP users under „Users“. Zammad integrates them just normal users, the difference is the login name.



1.4 Exchange

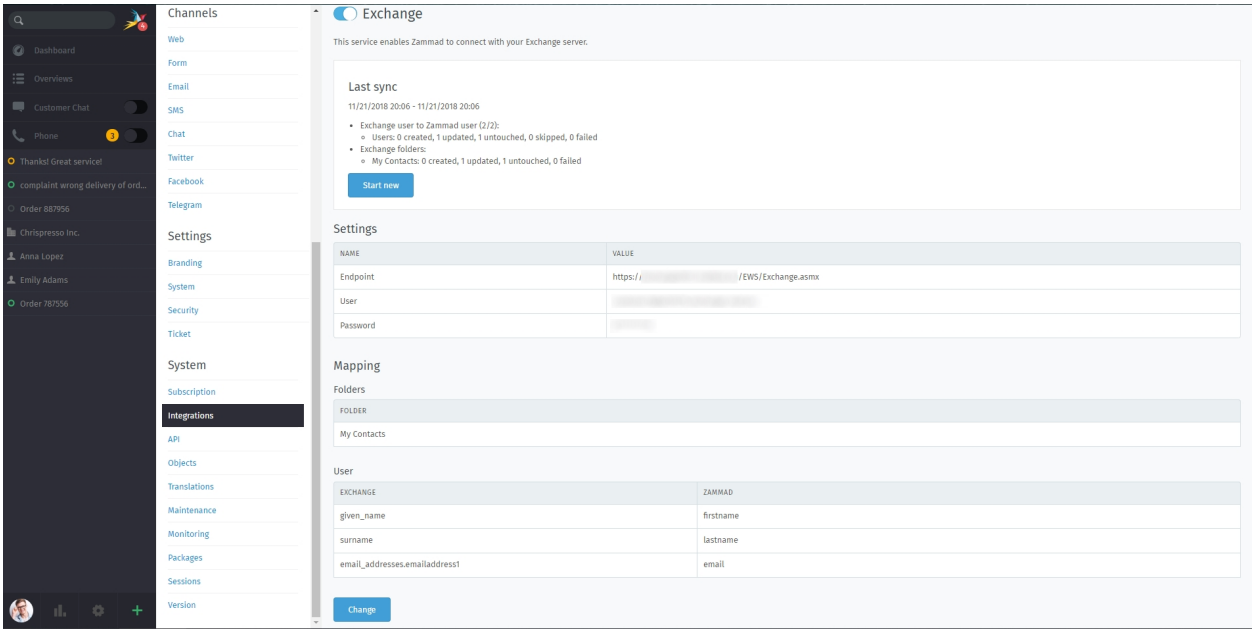
With our Exchange integration, you can easily use existing address books without having to update more than one source.

Bemerkung: 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!).

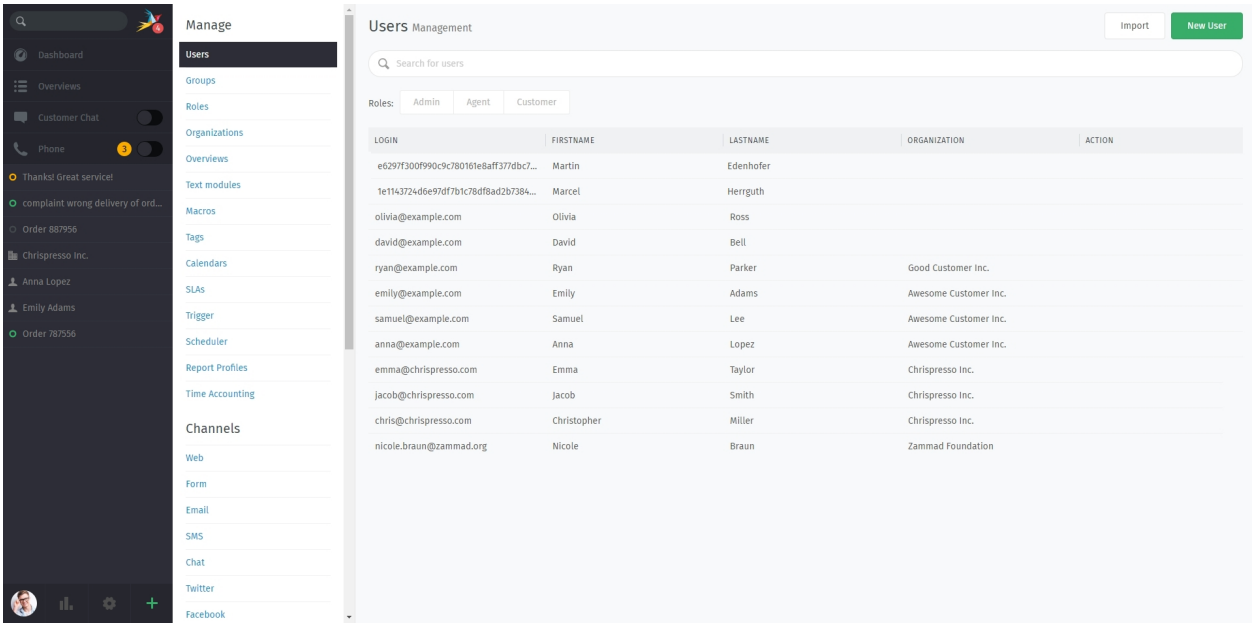
Bemerkung: Please refrain from syncing all addresses, as the results may not be what you expect (Exchange collects huge amounts of addresses). A central address book of your company to sync makes more sense, as you can ensure that Zammad gets only the data you need and want.

After pressing Continue, Zammad will check if the configuration is okay. You can then enable LDAP 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.

Bemerkung: 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.



Bemerkung: Customers get their own user accounts, too?

Yes! Unlike 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.

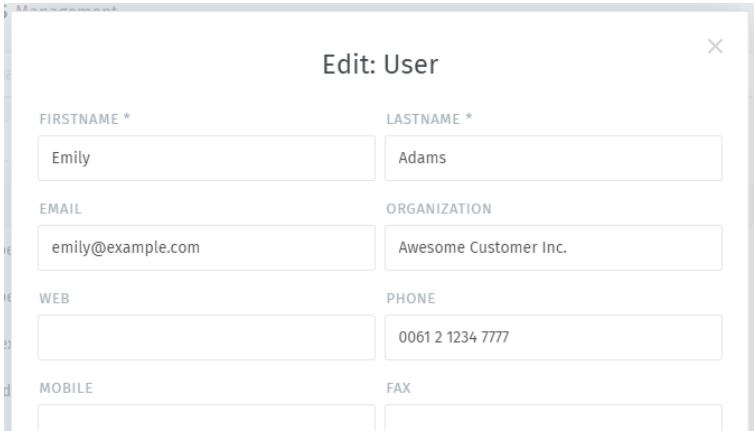


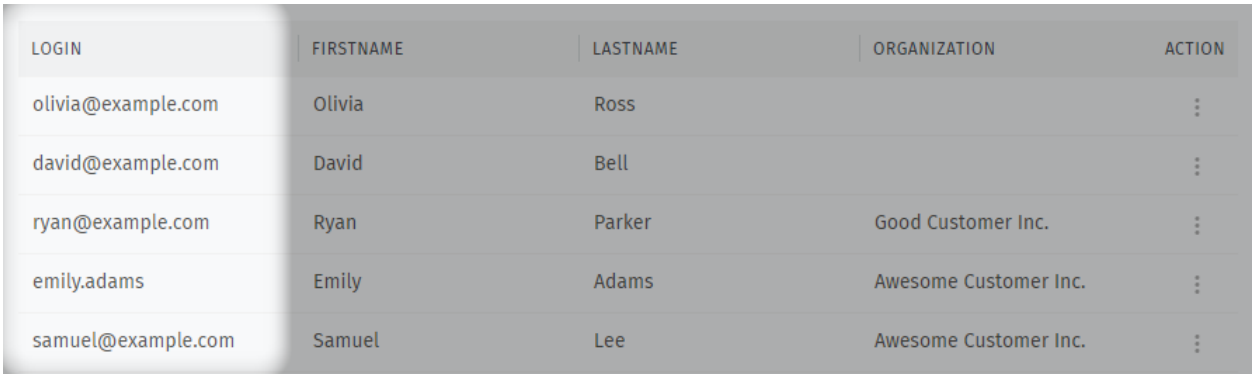
Abb. 8: User details can be set in the **New/Edit User** dialog.

Bemerkung: 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.**

Bemerkung:



LOGIN	FIRSTNAME	LASTNAME	ORGANIZATION	ACTION
olivia@example.com	Olivia	Ross		
david@example.com	David	Bell		
ryan@example.com	Ryan	Parker	Good Customer Inc.	
emily.adams	Emily	Adams	Awesome Customer Inc.	
samuel@example.com	Samuel	Lee	Awesome Customer Inc.	

Abb. 9: 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.

Hinweis: 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.

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.

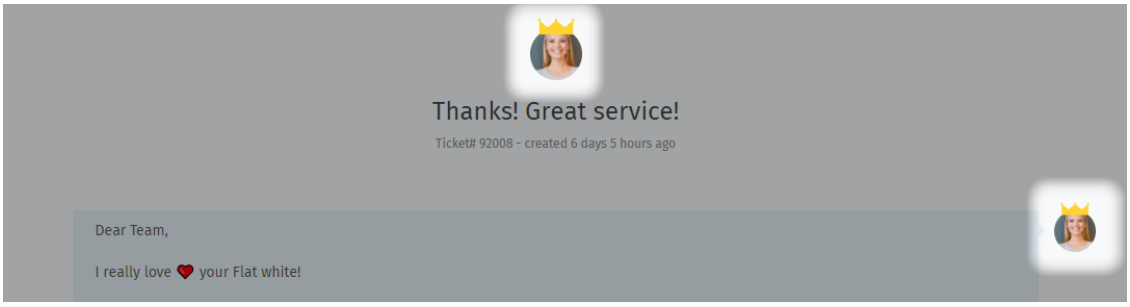


Abb. 10: VIPs are displayed with a crown above their avatars.

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

Hinweis: 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:

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.

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

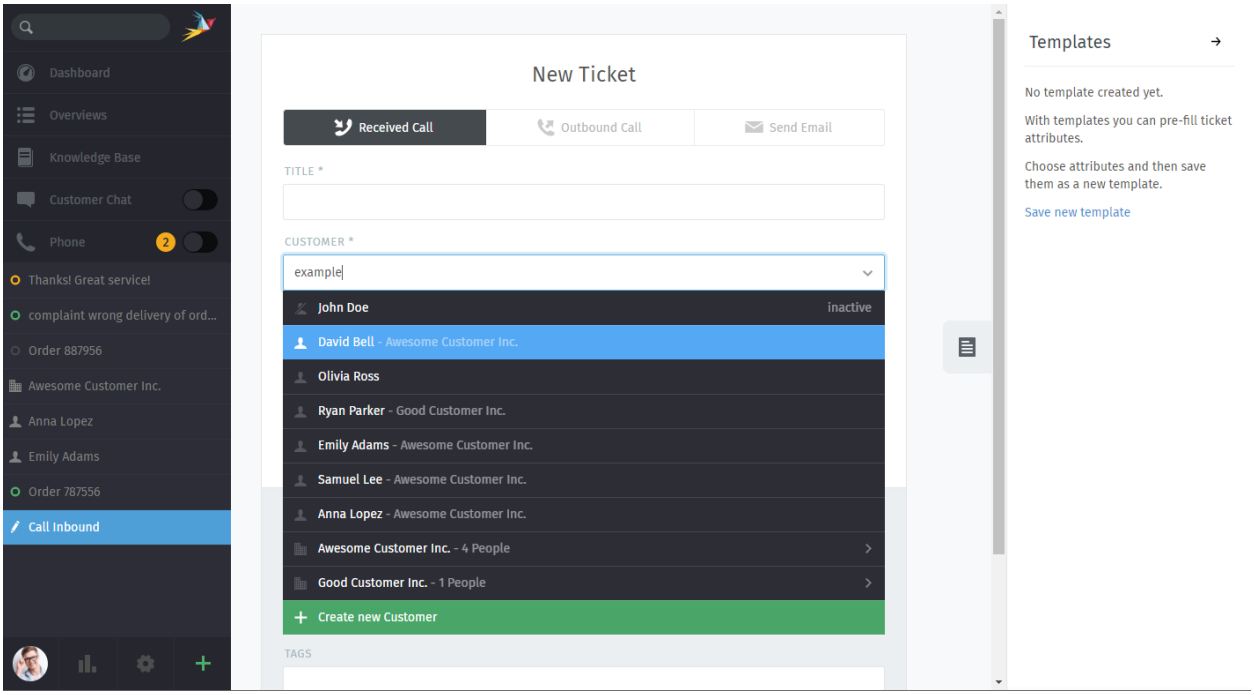


Abb. 11: A slashed-out icon indicates an inactive user. In other cases, inactive users are greyed out.

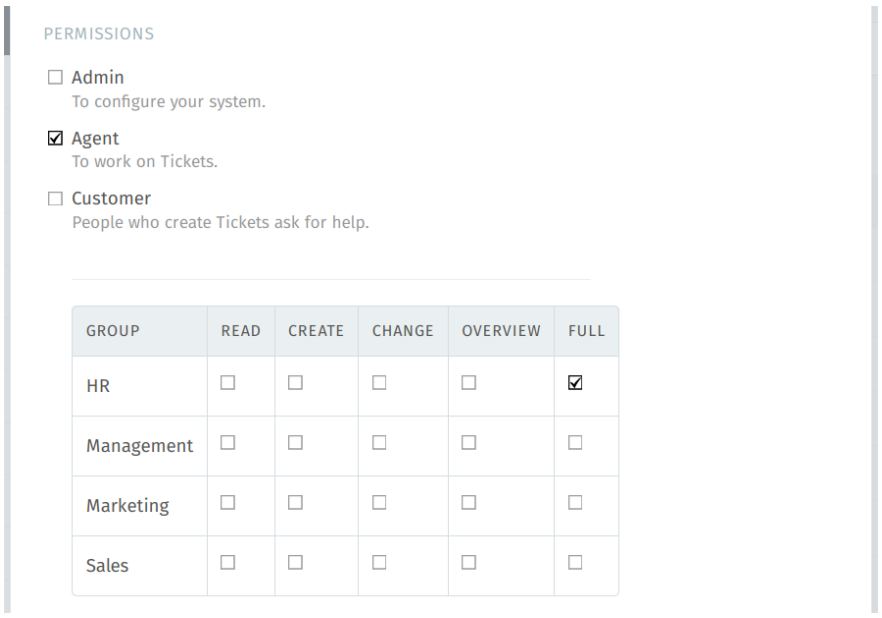


Abb. 12: **Top:** A user's roles decide what kinds of actions they can perform and which *groups* they belong to. **Bottom:** Group assignments can alternately be set on a per-user basis.

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

KAPITEL 2

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*.

Hinweis: 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.

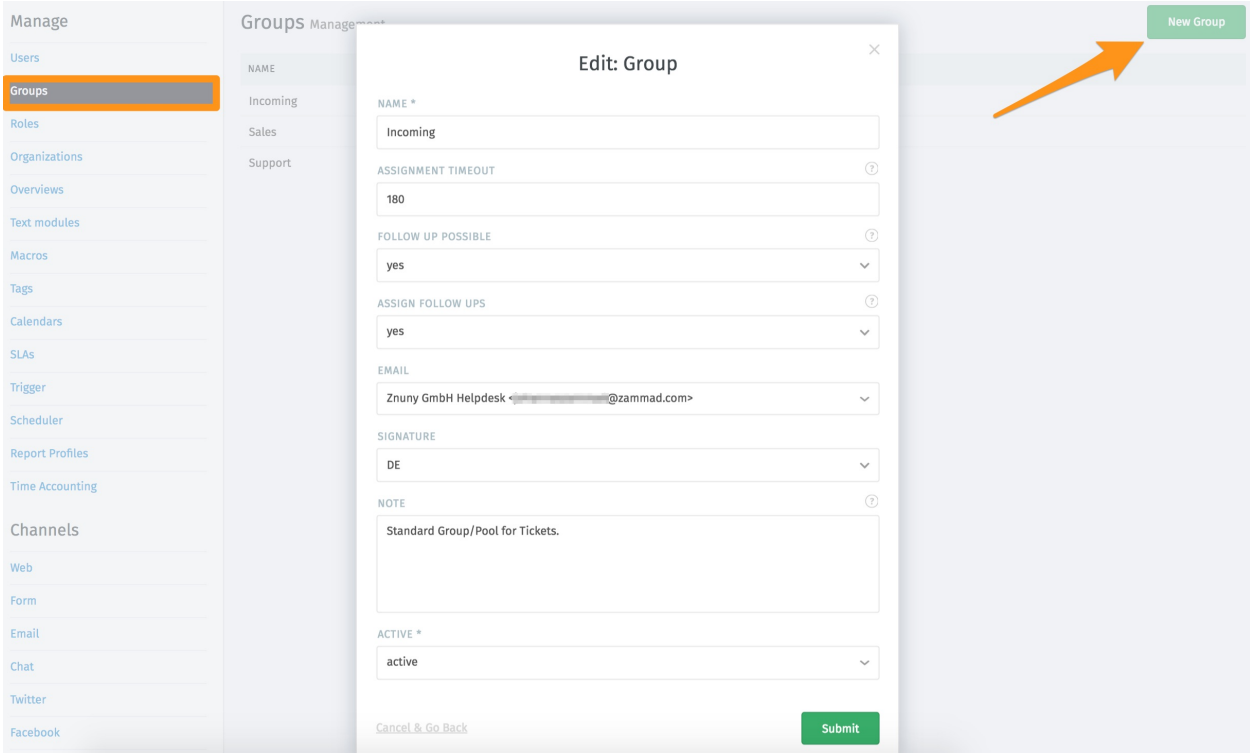
There you can edit the following settings:

- *Name*: choose a name for the group.
- *Assignment timeout*: the time in minutes after which the ticket's state will revert back to *unassigned* after the assigned agent hasn't worked on the ticket.
- *Follow up possible*: configure what happens when a customer replies to a closed ticket:
  - *yes*: the ticket will be reopened.

  - *do not reopen ticket but create new ticket*: the ticket will remain closed, and Zammad will instead create a new ticket for the customer's reply.

- *Assign follow ups*: configure whether a closed ticket that has been reopened due to a customer's reply should remain assigned to the last agent:
  - *yes*: the ticket will remain to the last agent who owned it.
  - *no*: Zammad will unassign the ticket.
- *Email*: choose which email address will be used as the sending address (*From* header) when agents reply to tickets in this group. Email addresses can be configured in Channels → Email → Accounts.
- *Signature*: choose which *signature* to use when replying to tickets in this group.
- *Note*: an internal note about the group that is only visible to people who can access the group management area.
- *Active*: choose whether the group is enabled or not. Groups cannot be deleted, they can only be set to inactive.

Eventually it should look something like this:



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:

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

READ …view tickets

CREATE …create new tickets

CHANGE …modify existing tickets

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

Abb. 1: Use the group access table to grant per-group privileges.

OVERVIEW ... see ticket overviews (but not ticket details)

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

Bemerkung: 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*

Abb. 2: Simply set your access levels right on the target user.

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

☐ ticket
Ticket Interface

☐ agent
Access to Agent Tickets based on Group Access

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

☐ customer
Access to Customer Tickets based on current_user and organization

Abb. 3: First, set your access levels on a role...

Abb. 4: . . . then, add that role to the target user.

Bemerkung: **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"

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.

Hinweis: 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	☑	☐	☑	☑	☐

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

KAPITEL 3

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.

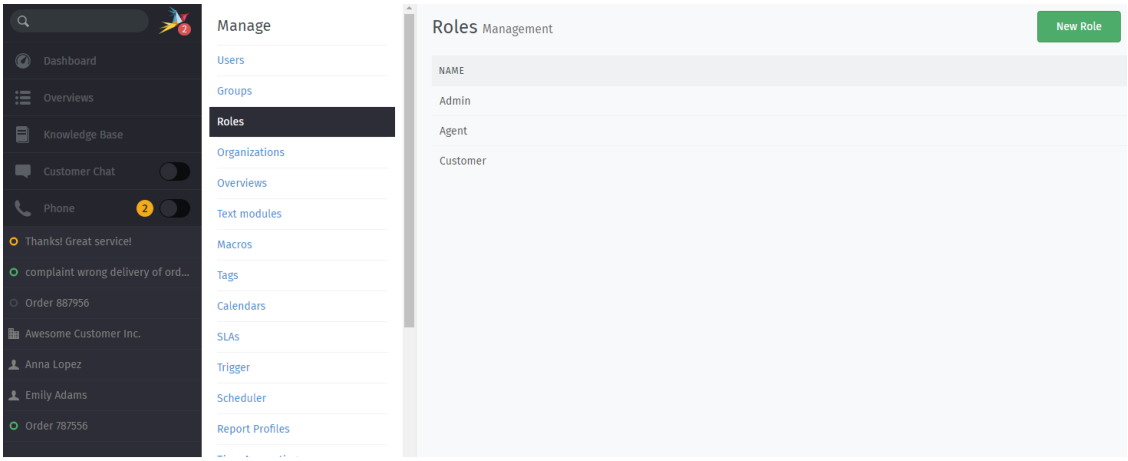


Abb. 1: Assign user privileges in the Admin Panel, under **Manage > Roles**.

Bemerkung: What's New in v3.5

Now, 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.

Tipp: 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?

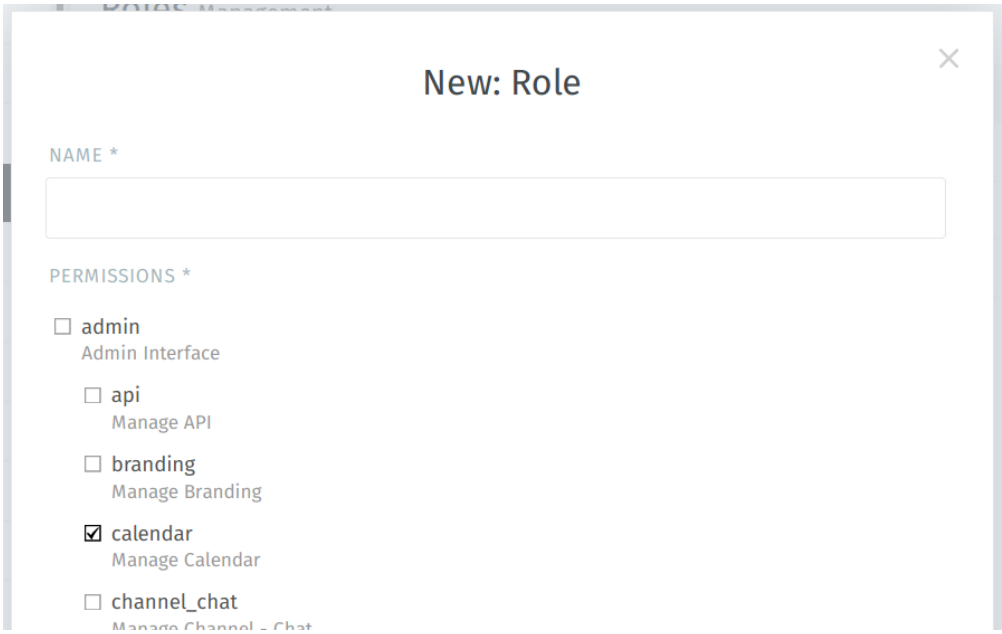


Abb. 2: The `admin.calendar` permission gives you access to the **Manage > Calendars** admin panel.

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

Bemerkung: **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.

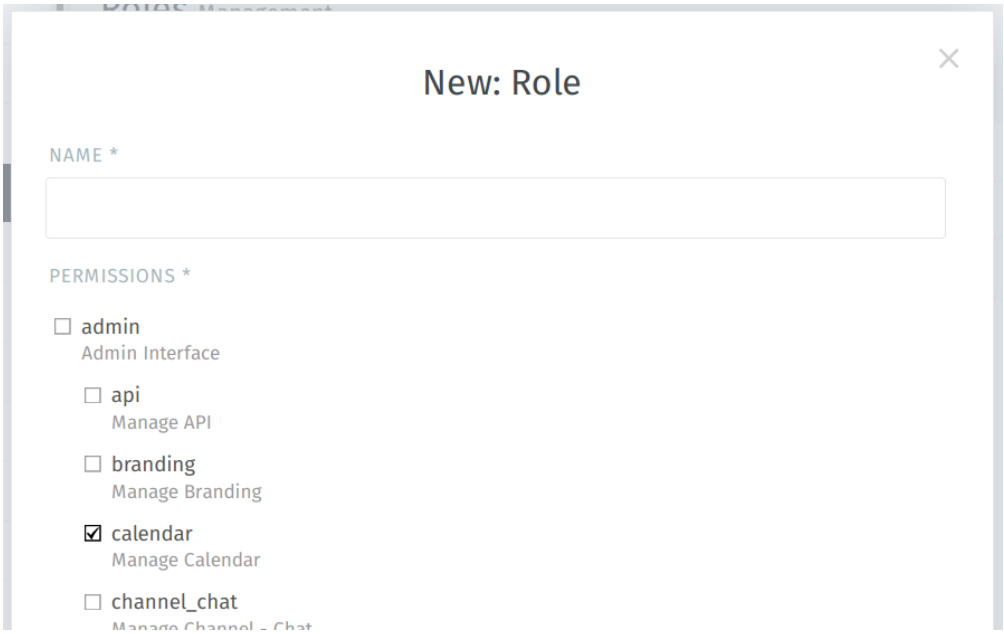


Abb. 3: Admin permissions are shown at the top of the New Role dialog…

`admin.api` System > API

`admin.branding` *Settings > Branding*

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

`admin.channel_chat` *Channels > Chat*

Hinweis: **Trying to grant access to send messages in live chats?**

Use `chat.agent` instead.

`admin.channel_email` *Channels > Email*

Bemerkung: There is no specific permission for the Google channel yet. This is an open feature request.

`admin.channel_facebook` *Channels > Facebook*

Hinweis: **Trying to grant access to view/update tickets from Facebook?**

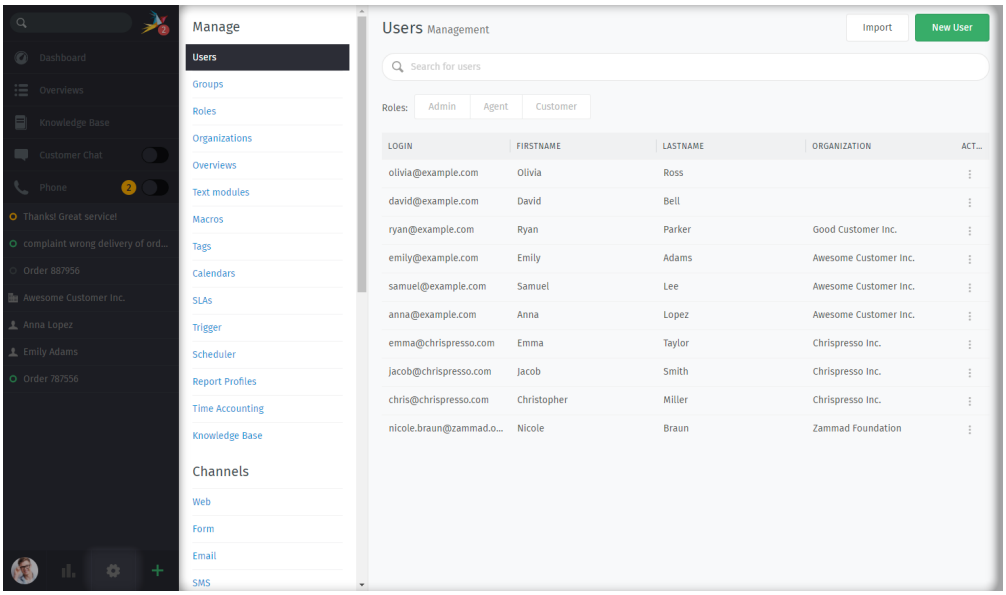


Abb. 4: . . . and give users access to the pages of the Admin Panel.

That's in *Group Access Levels*.

---

admin.channel_formular *Channels > Form*

admin.channel_sms Channels > SMS

admin.channel_telegram *Channels > Telegram*

Hinweis: Trying to grant access to view/update tickets from Telegram?

That's in *Group Access Levels*.

---

admin.channel_twitter *Channels > Twitter*

Hinweis: Trying to grant access to view/update tickets from Twitter?

That's in *Group Access Levels*.

---

admin.channel_web *Channels > Web*

admin.data_privacy *System > Data Privacy*

> **Gefahr:** 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*

Hinweis: 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*

Bemerkung: 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*

Bemerkung: 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*

Hinweis: 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.text_module *Manage > Text Modules*

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

admin.time_accounting *Manage > Time Accounting*

Hinweis: 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*

Bemerkung: 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 agents' accounts

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

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

3.2.2 Agent Permissions

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

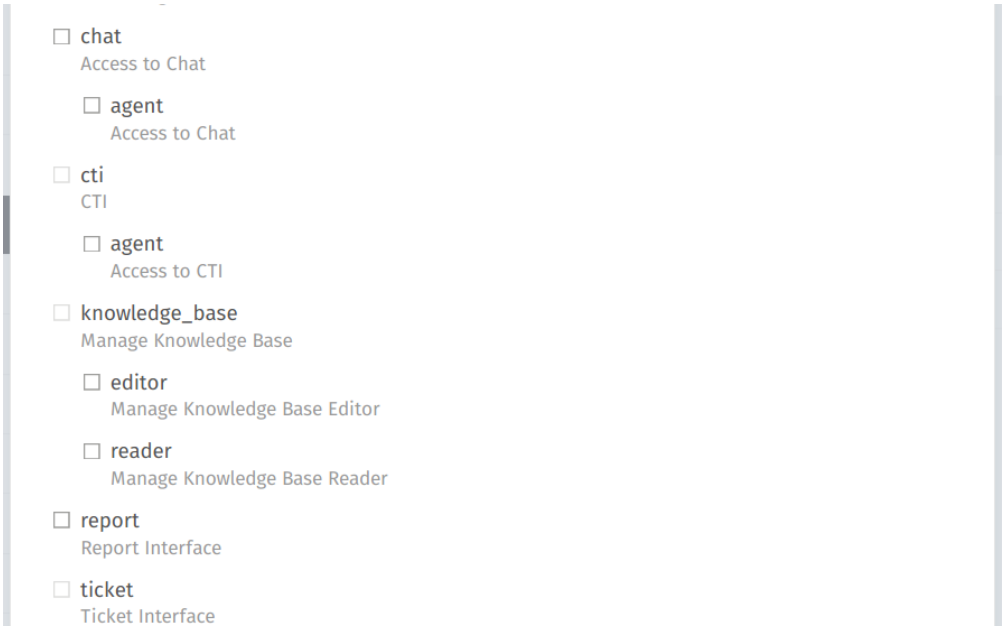


Abb. 5: Agent permissions are shown in the middle of the New Role dialog…

chat.agent Customer Chat

Hinweis: Requires configuration of *Chat Channel*

cti.agent Provides access to Caller Log

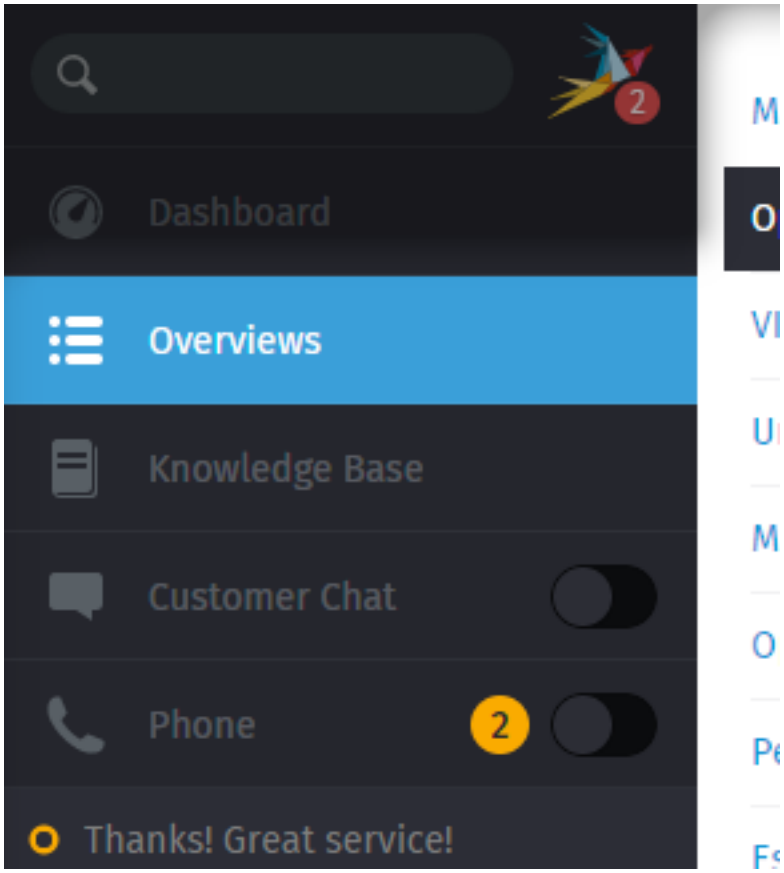


Abb. 6: . . . and give users access to new sidebar tabs for communicating with customers.

Hinweis: Requires configuration of either integrations

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

knowledge_base Knowledge Base

`knowledge_base.editor` create/edit privileges

Hinweis: Editor permissions always include reader permissions.

`knowledge_base.reader` read privileges for internal content

Hinweis: Public articles are always visible.

`report` *Reporting*

Warnung: 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**!

Bemerkung: 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

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

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*.

Hinweis: Point of technicality

As of Zammad 3.5, you can assign both agent and customer roles to the same user—but 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`.

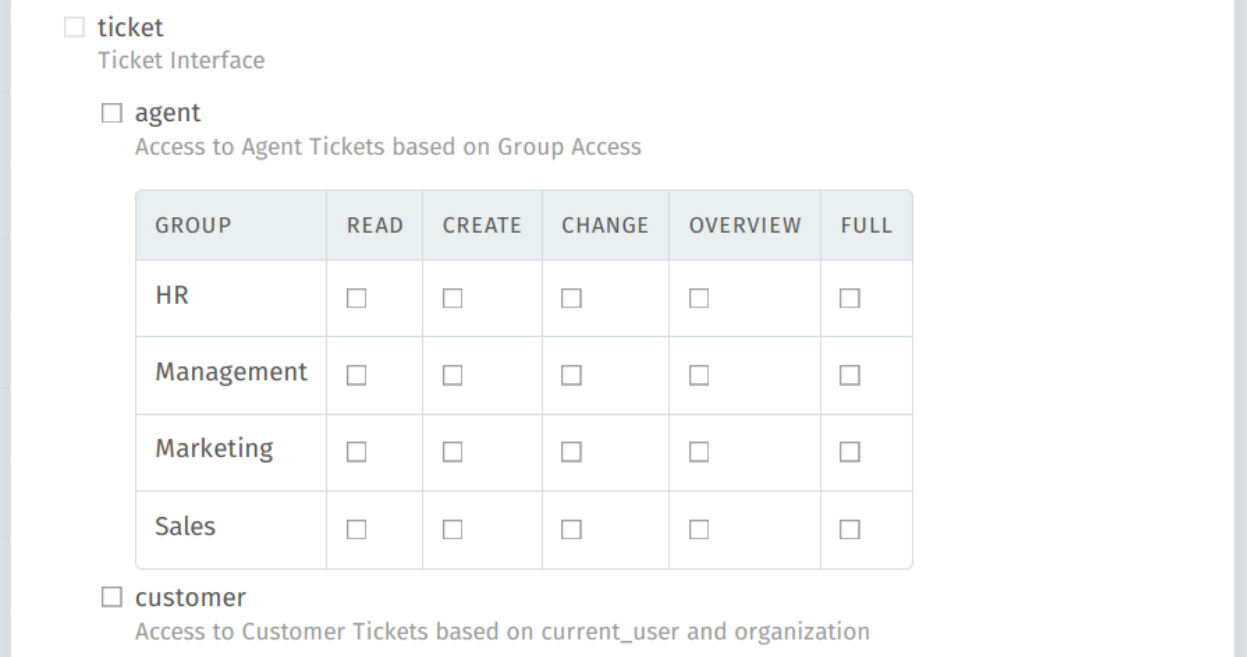


Abb. 7: The group access table is shown when there is **more than one active group** in the system.

3.2.3 User Preferences Permissions

Bemerkung: **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.

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

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

Bemerkung: **Security Tip**

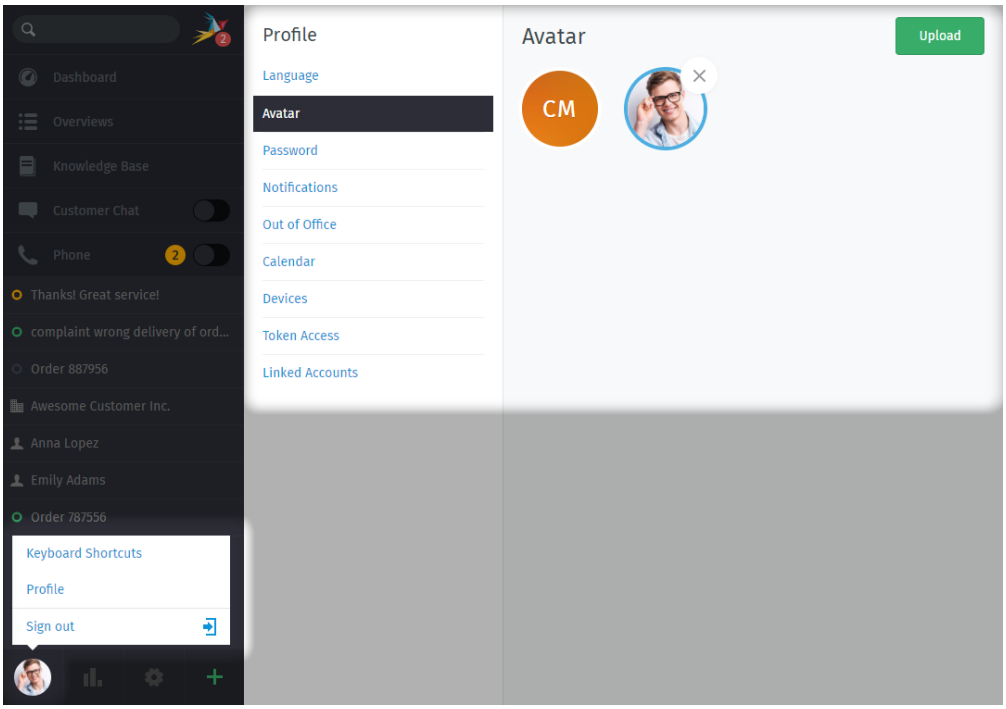


Abb. 9: . . . and give users access to the pages of their User Profile.

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

Bemerkung: Security Tip

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*

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

`user_preferences.notifications` Configure ticket notification settings

Bemerkung: 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

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

`user_preferences.password` Change account password

> **Warnung: 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

Bemerkung: 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).

> **Warnung: This role should never provide admin/agent permissions.**

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

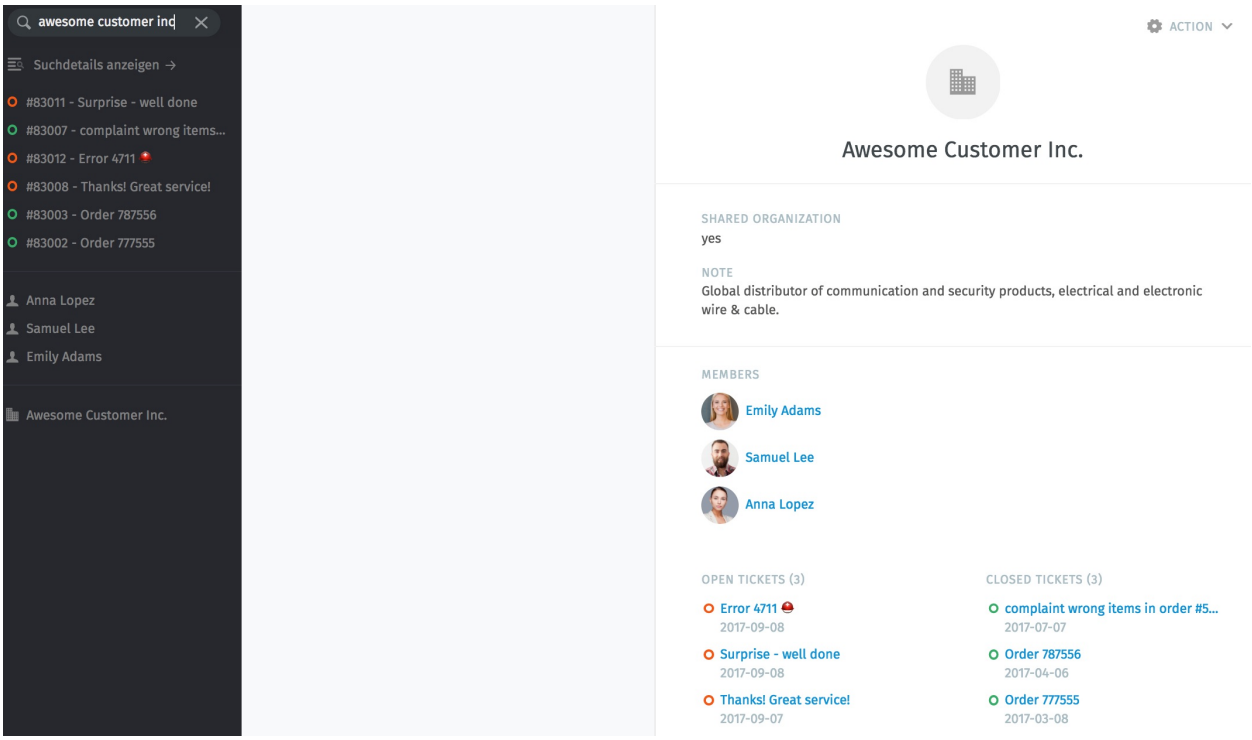
KAPITEL 4

Organizations

Organizations allow you to group customers. This has two important advantages, among other things.

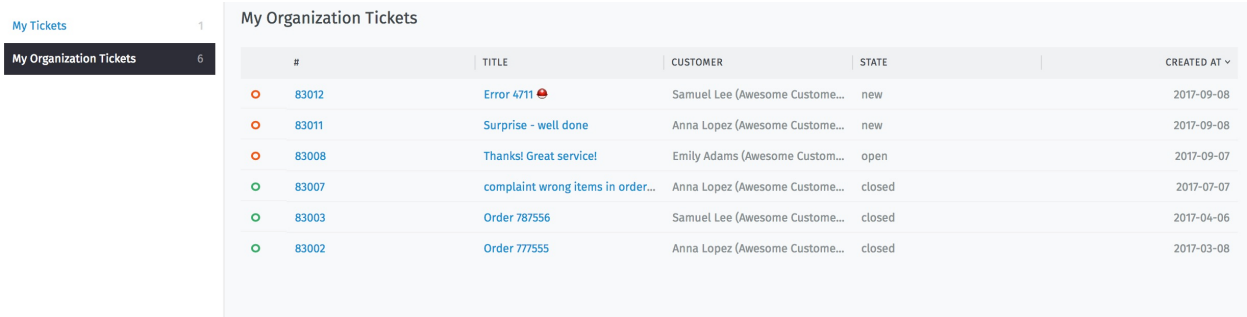
1.) As an agent you not only have the overview of a customer's tickets, but also an overview of the entire organization.

For example, by searching for the organization, all the tickets for the organization are displayed and they can be opened easily by a click. This overview appears as follows:

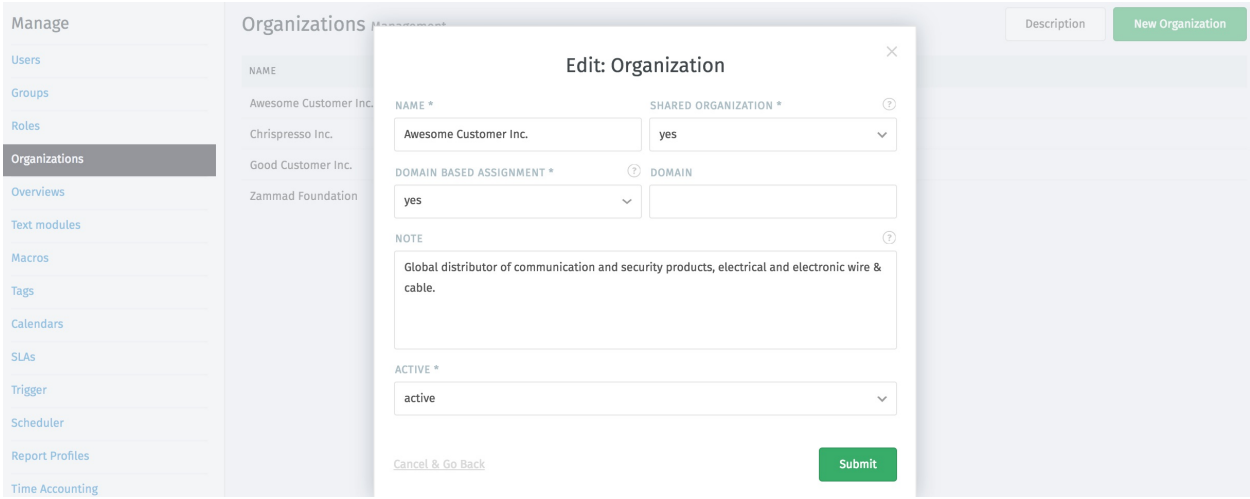


2.) As a customer, you can view and edit your colleagues' tickets (if the organization is a „shared organization", you can set this as a parameter for each organization. See the Edit-Mask).

For example: For a customer user who has only created one ticket himself, but whose entire organization has created 6, the overview would look like this:



In the organizations management area (Admin Interface –> Manage –> Organizations) you can manually add, edit or delete existing organizations. That's the Edit-Mask:



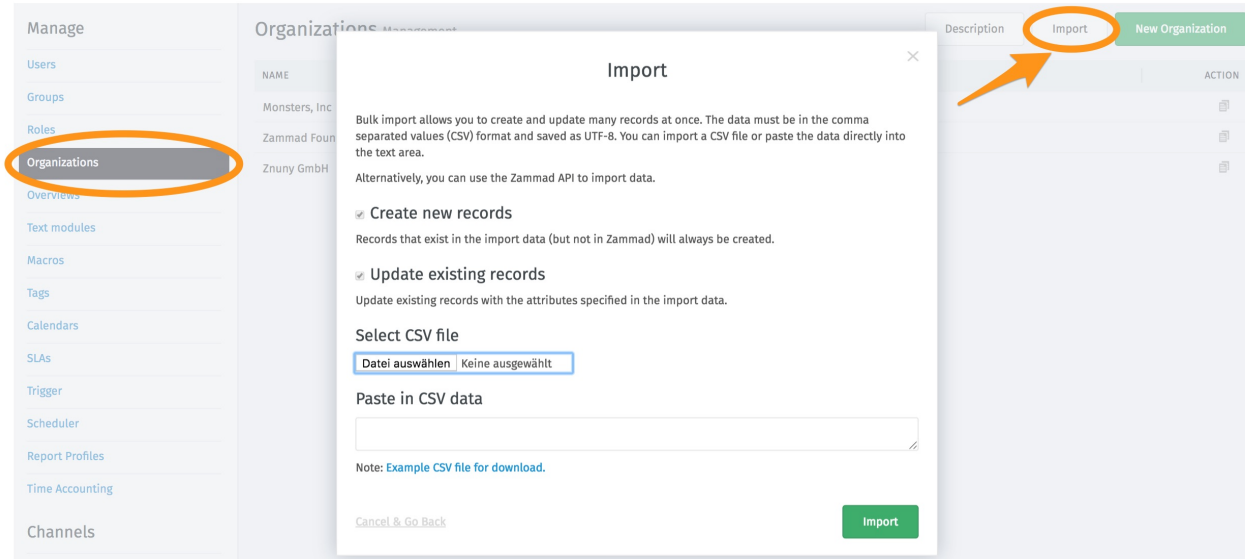
Within the organization the following things can be set:

- if it's a shared organization (All customers who are assigned to this organization can view and edit the group tickets)
- if the assignment is domain based (assign users based on domain)
- note
- if it's active or inactive

4.1 Import of organizations 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
Scheduler
Report Profiles
Time Accounting
Channels
Organizations
NAME
Monsters, Inc
Zammad Foun
Znuny GmbH
Description
Import
New Organization
ACTION
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.
Select CSV file
Datei auswählen
Keine ausgewählt
Paste in CSV data
Note: Example CSV file for download.
Cancel & Go Back
Import

KAPITEL 5

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.

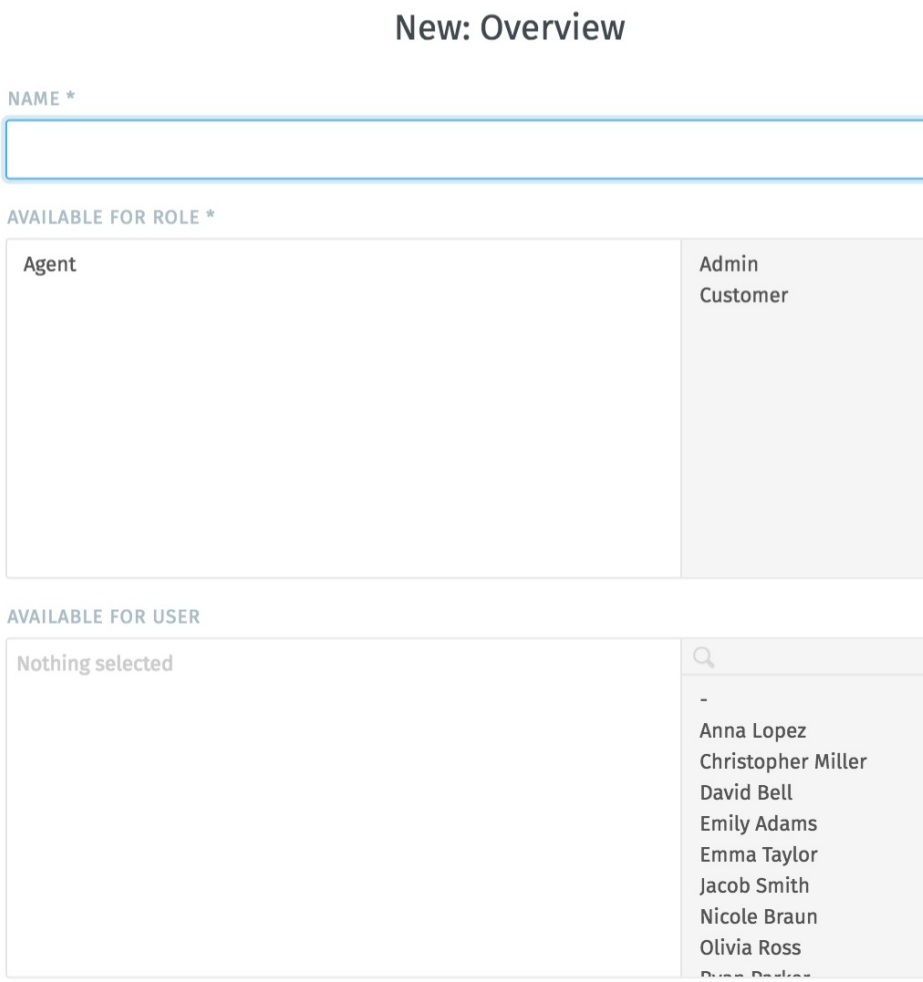
In the Overview Management Area you can add new overviews, edit or delete them.

Warnung: 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.

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

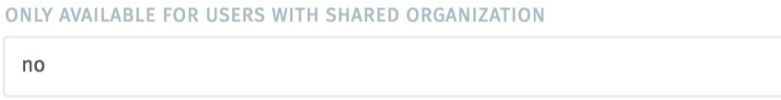
The following attributes can be set when creating an overview:



Available for role / Available for user

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

With these settings you can choose whether to make the overviews available to an entire group of people (by selecting the role) or to specific users. The entries in both fields apply. This means that you can also select individual users in addition to sharing the overview for all role members.



Only available for users with shared organization

Hinweis: Shared organization is a setting in the organisations-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…

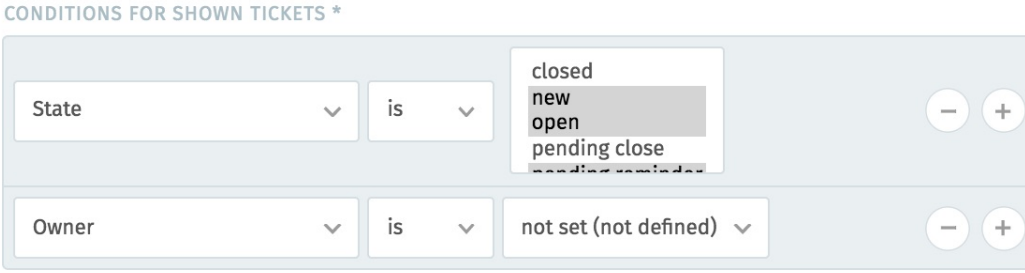
Bemerkung: Users also refers to the customer role in this case.

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.

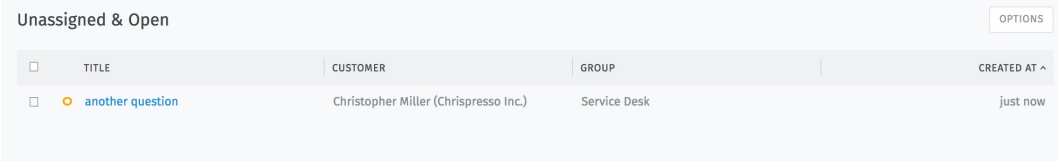
For example: Agent A is on vacation and Agent B will take care of his tickets. Then an overview can be set up, which only shows Agent B all new tickets from Agent A for this period of time, without having to search for them separately.

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



Conditions for shown tickets

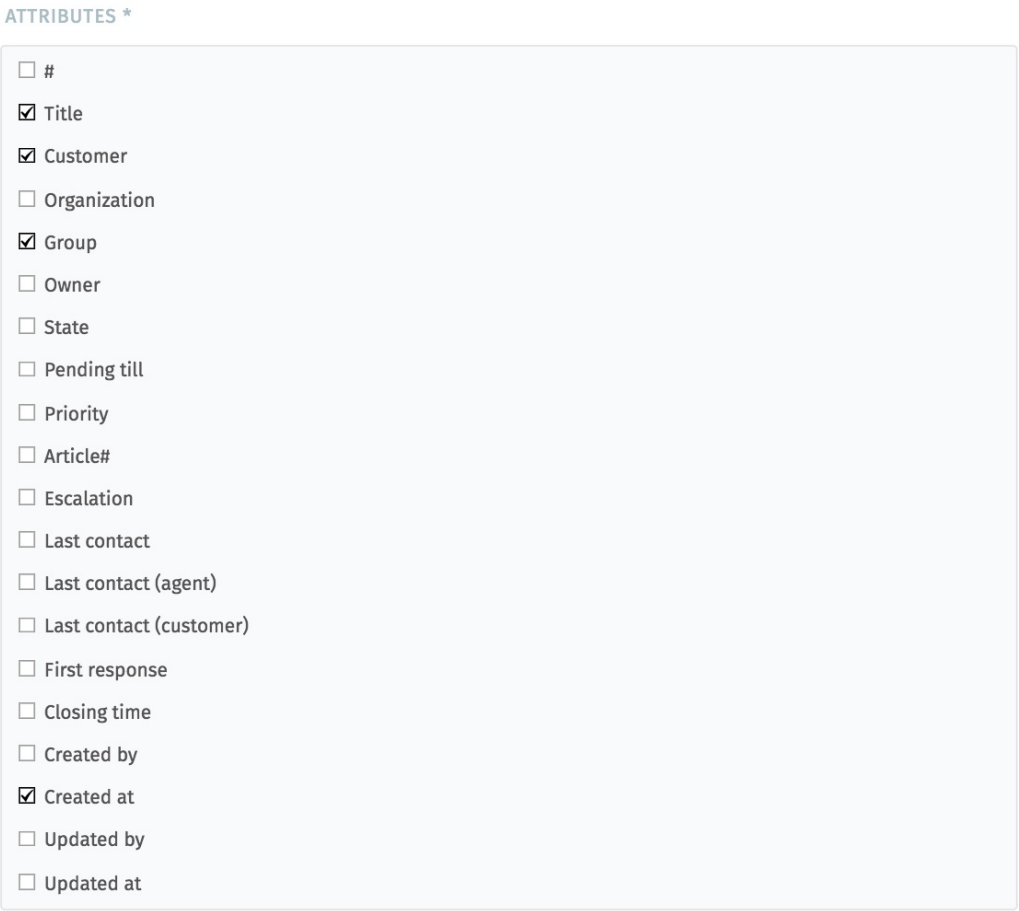
What conditions should the listed tickets contain? (it is like a filter) You can add more than one condition. In the preview you have the possibility to double check if your entry of the conditions makes sense by directly displaying tickets that match your filtering.



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).

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

Ordering, grouping and active

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

- direction: The direction of the order
- group by: Should the tickets be displayed again grouped by a specific attribute within the list?
- active: Set them active or inactive

KAPITEL 6

Text Templates

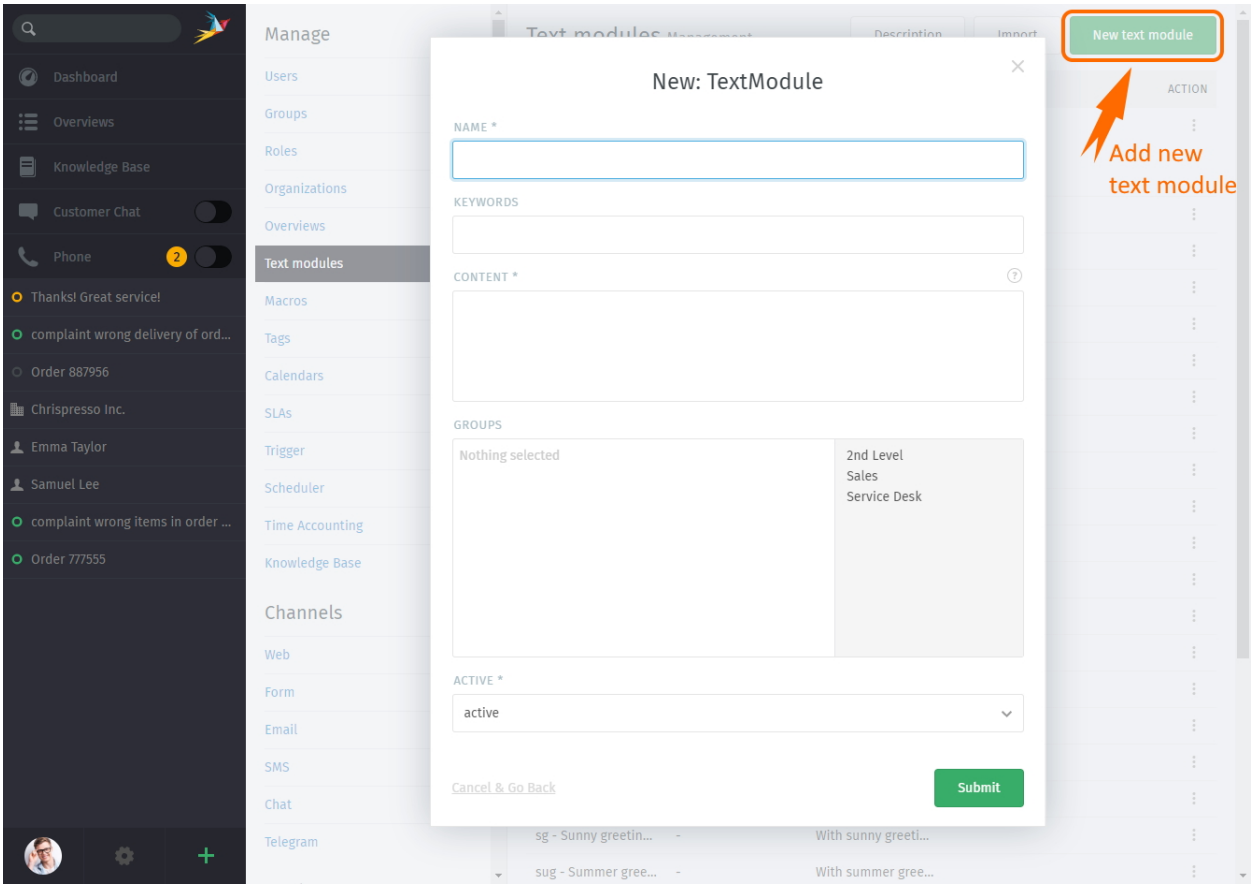
Warnung: With Zammad 3.1 we introduced group based text modules. This documentation contains information that might no longer apply to Zammad versions earlier 3.1.

Create text templates to spend less time writing responses. You can create text passages (Text modules) or even entire response templates.

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

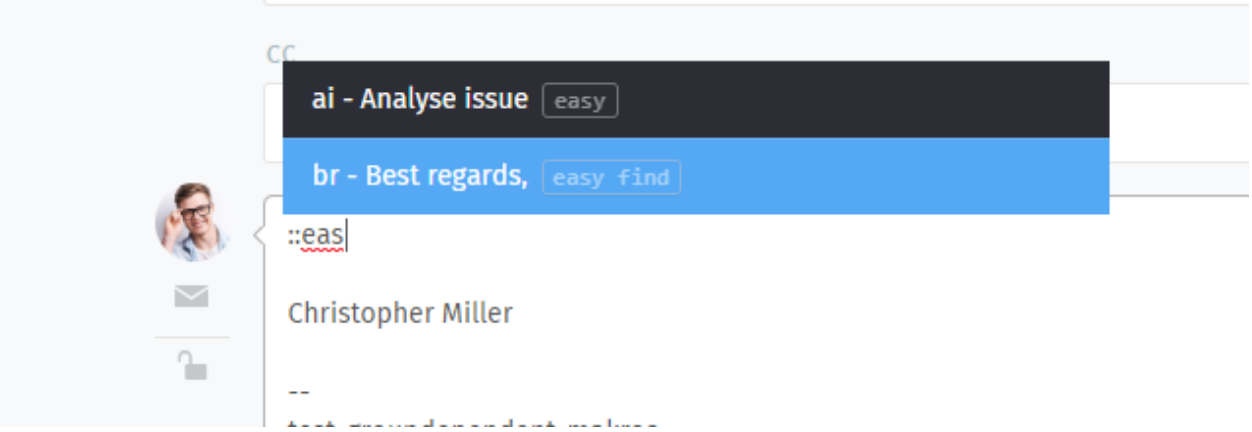
6.1 Text modules

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.

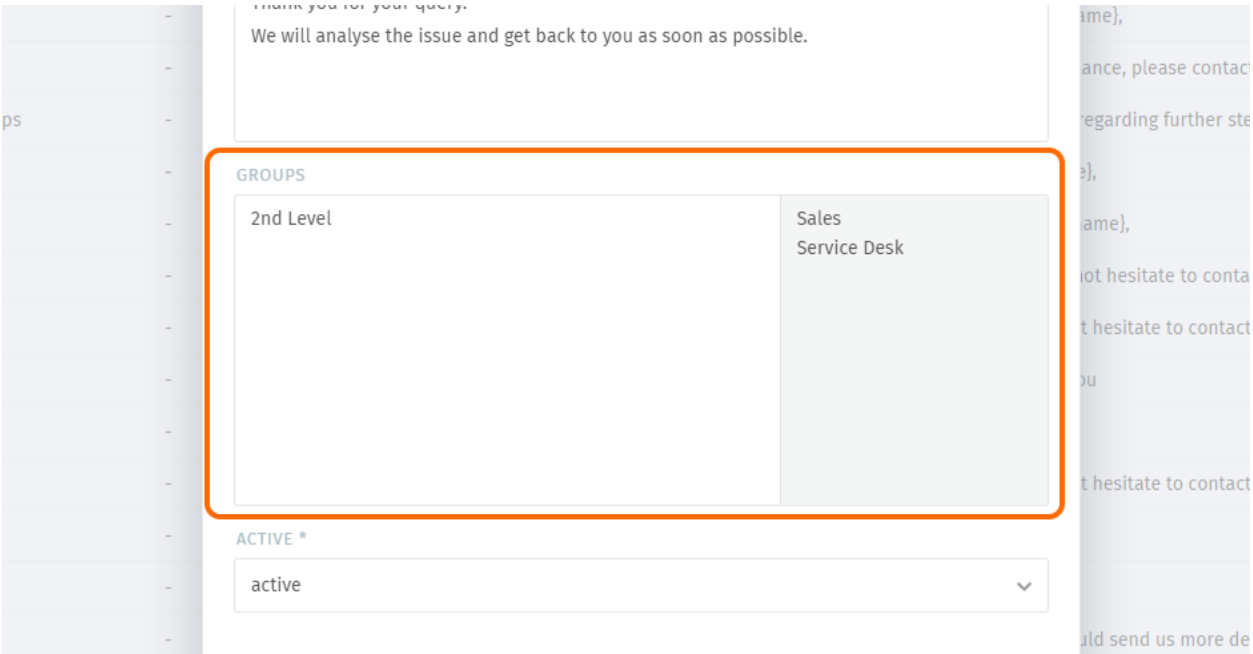
Creating keywords makes it easier to find the right text module.



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.



Bemerkung: You can find more information on how to use text modules on our User Documentation.

Tipp: 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.

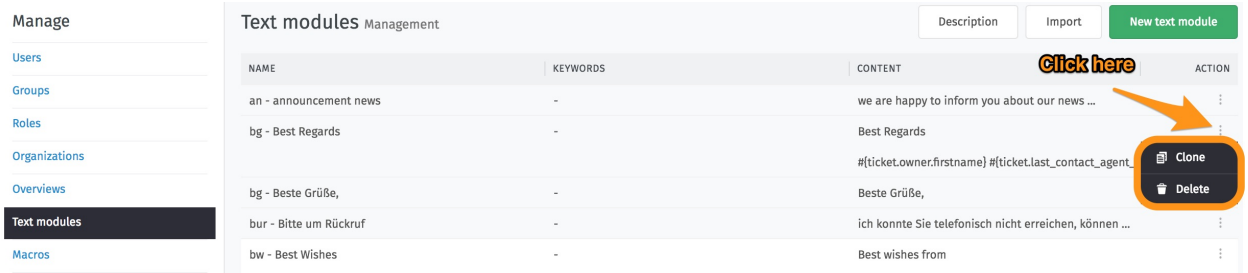
Examples of snippets are:

```
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.

6.2 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:

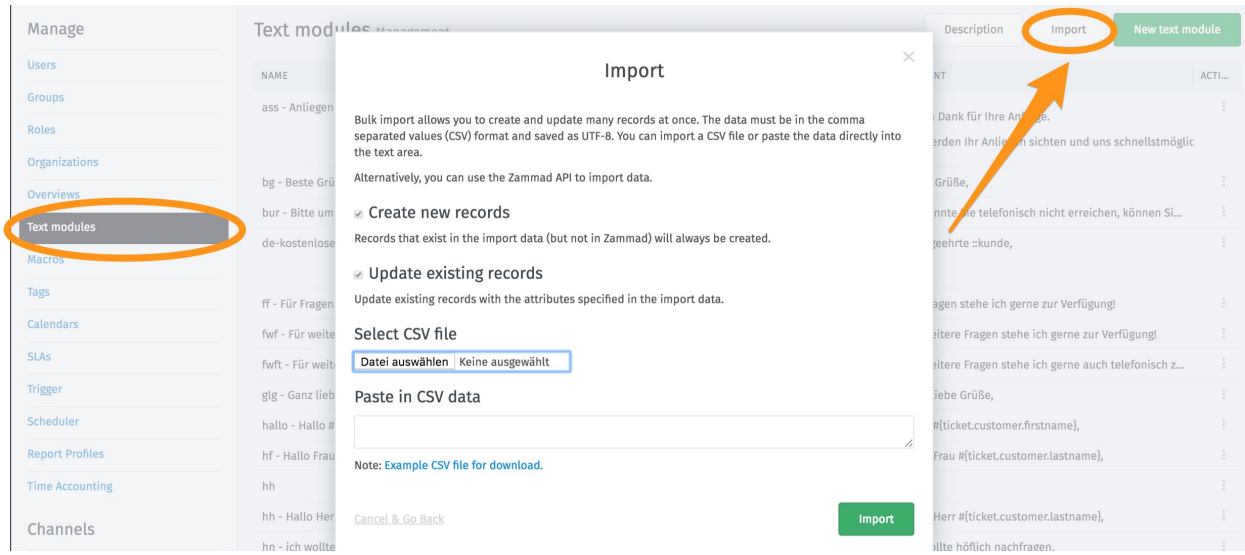


When cloning, text modules with all attributes are duplicated and can be edited later.

6.3 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.



6.4 Ticket templates for new tickets

Ticket templates in new tickets can be created by any agent. Therefore you will find information inside of our User-Documentation .

KAPITEL 7

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:

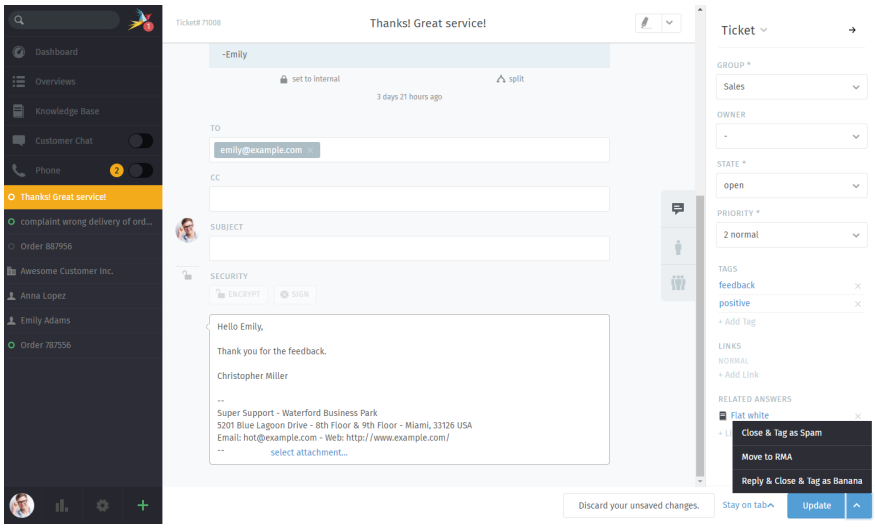


Abb. 1: Choose from the macros **Close & Tag as Spam**, **Move to RMA**, and **Reply & Close & Tag as Banana**.

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.

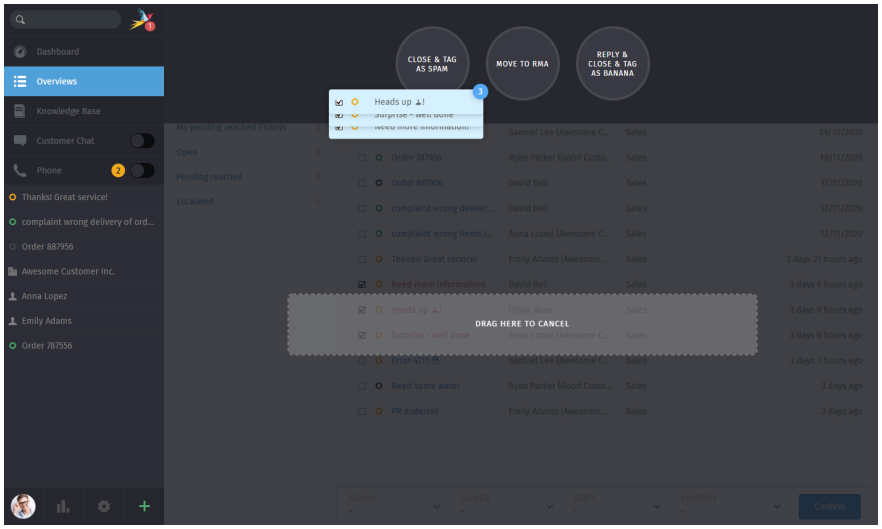
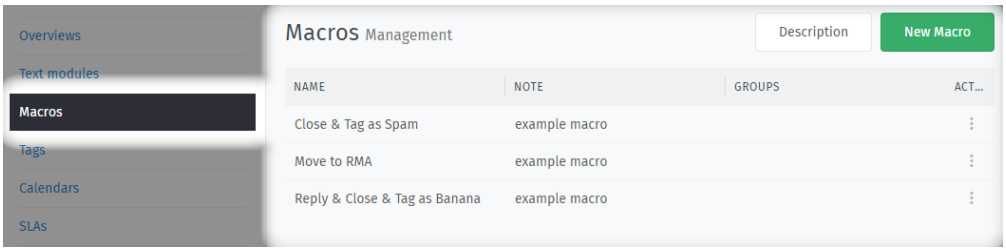


Abb. 2: Select your desired tickets, then click-and-drag to apply a macro to all of them.



Hinweis: 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

   Tags add `spam`

   Owner current user

   Tipp: 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

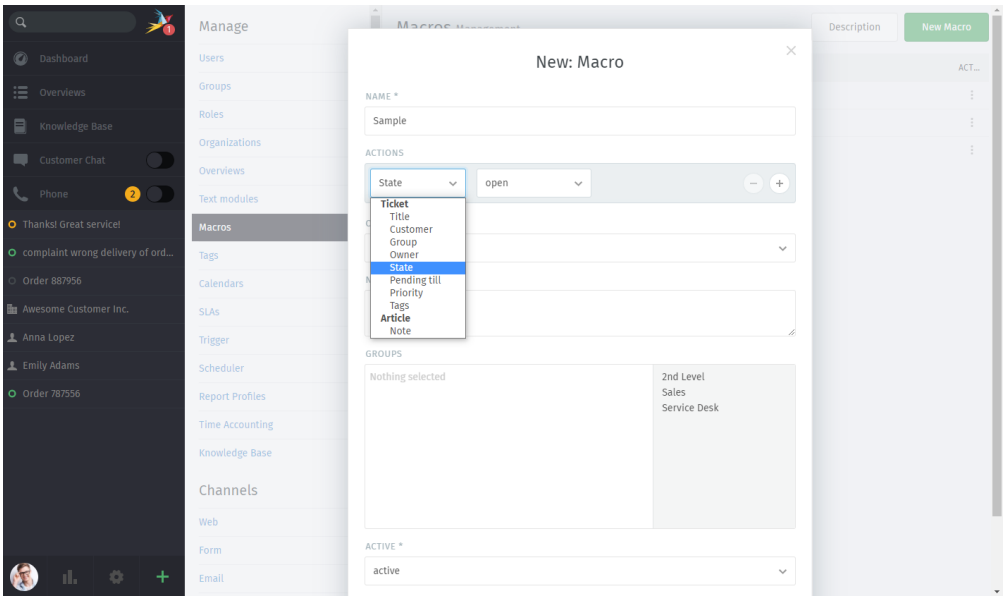
   Owner current user

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

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

> **Warnung:** 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**.

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

KAPITEL 8

Tags

With tags, tickets can be categorized and marked.

8.1 How to tag

All agents can tag tickets. You can find further information about tagging of tickets on our User Documentation.

8.2 Tag-Overviews

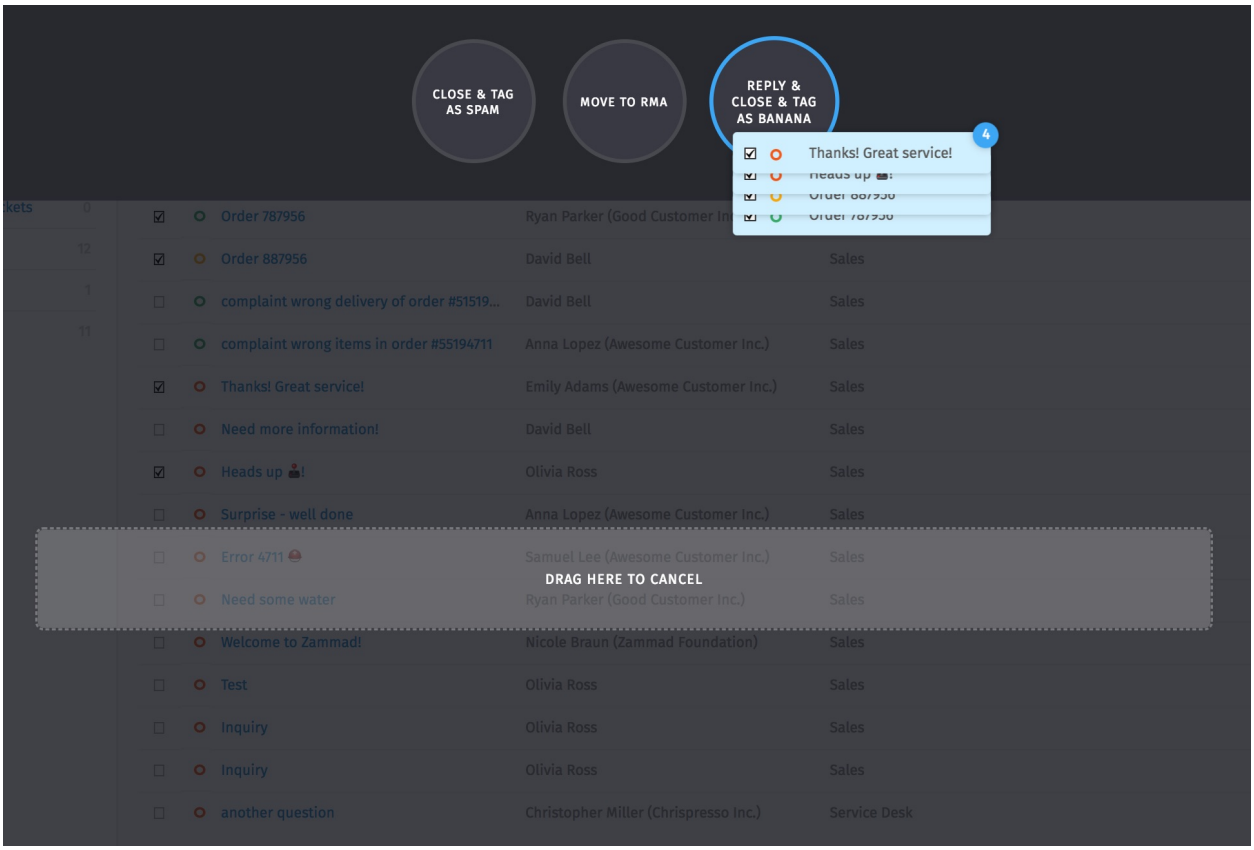
You can list all the marked tickets by tag in different ways:

1. by providing an appropriate overview

Create a new overview and select „Tag contains . . . “ as condition. You can find further information in *Overviews*

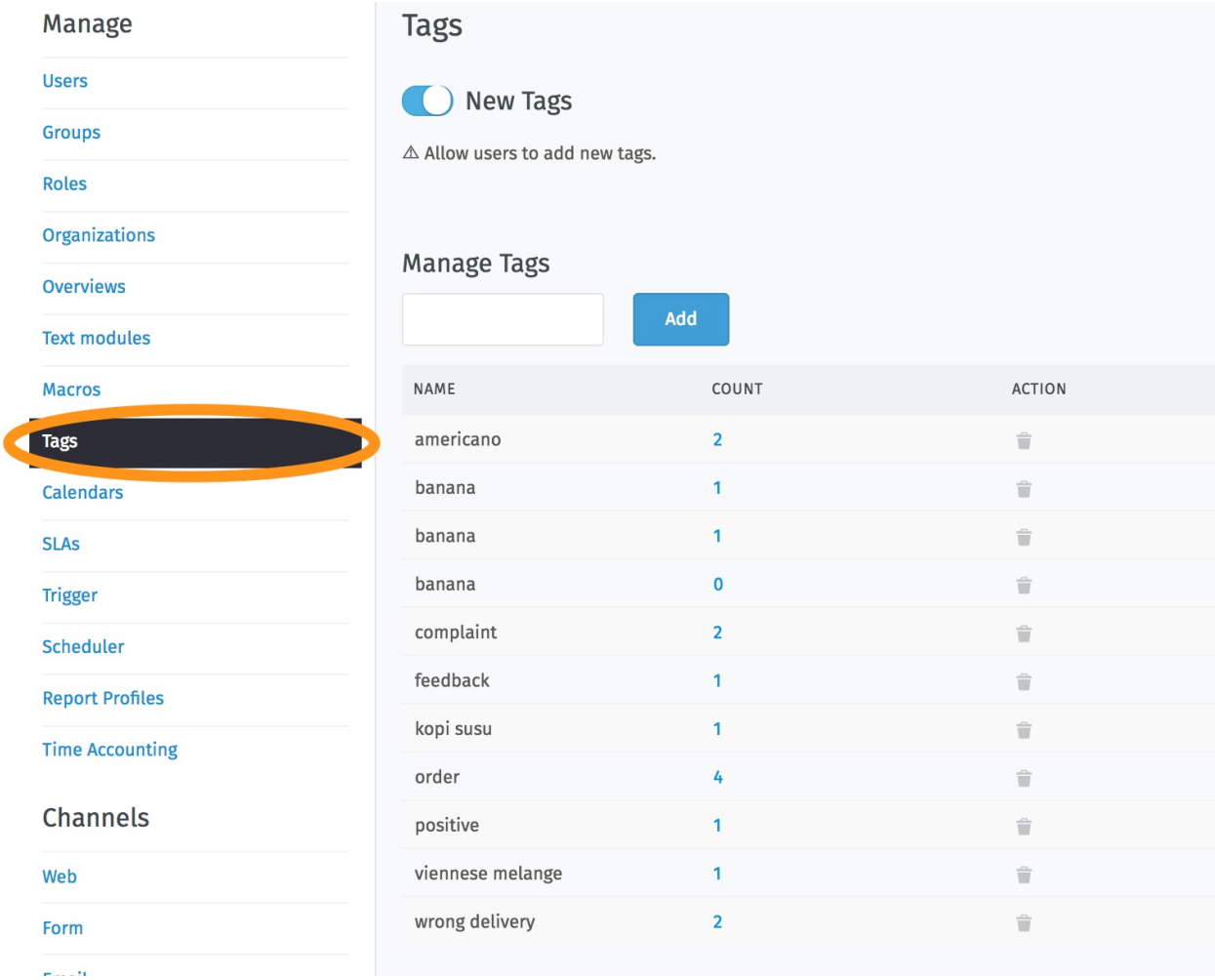
2. by a search-query

Enter e. g.: `tag:banana` 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:

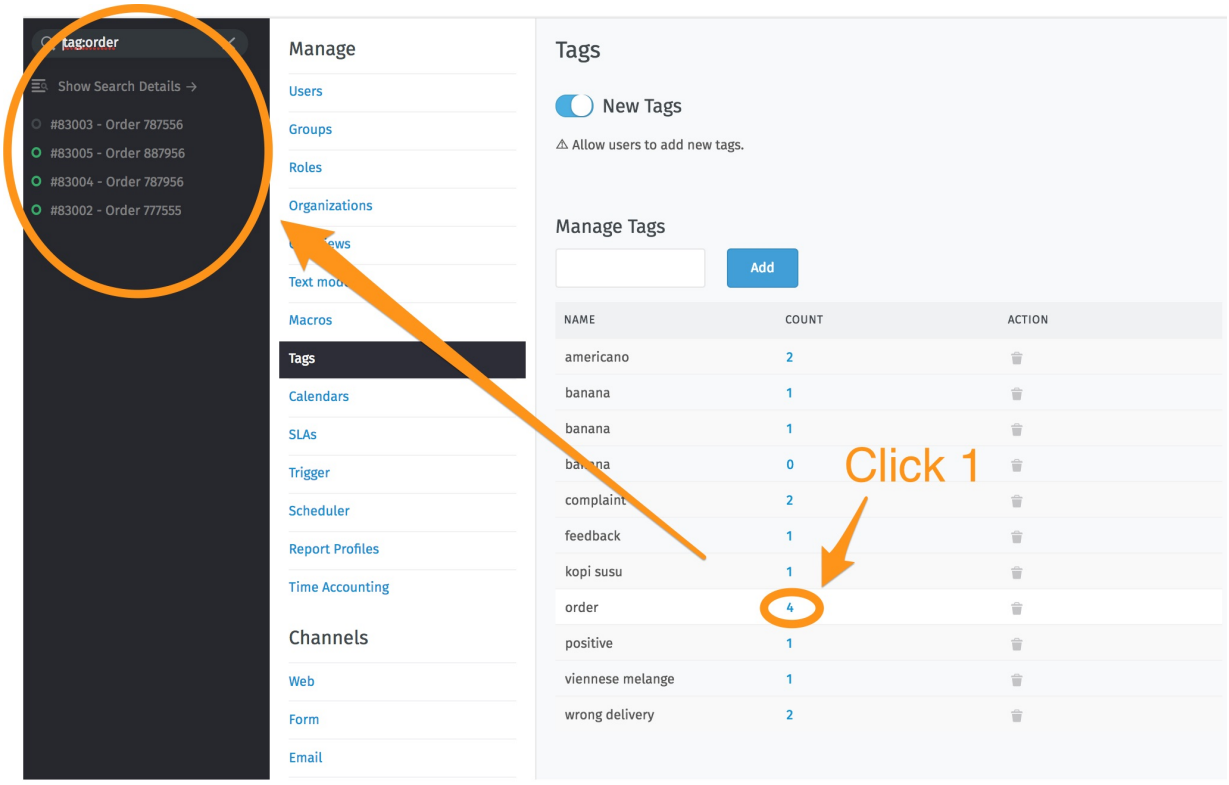


3. by clicking on the tag in the Tag Management Area

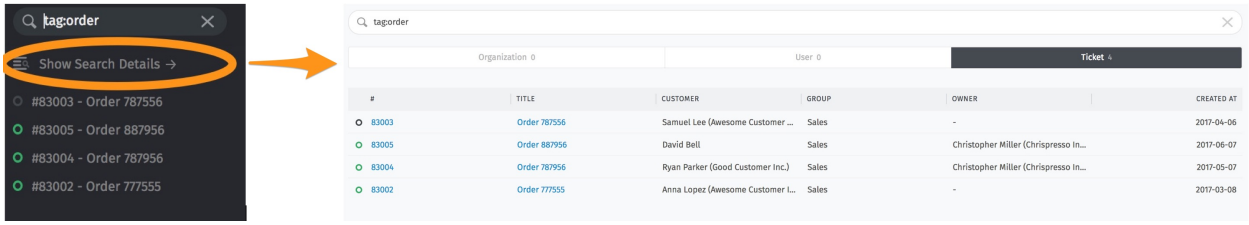
In the Tag Management you can find an overview of existing tags used in the system (and a counter how often they are used):



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“:



8.3 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.

KAPITEL 9

Calendars

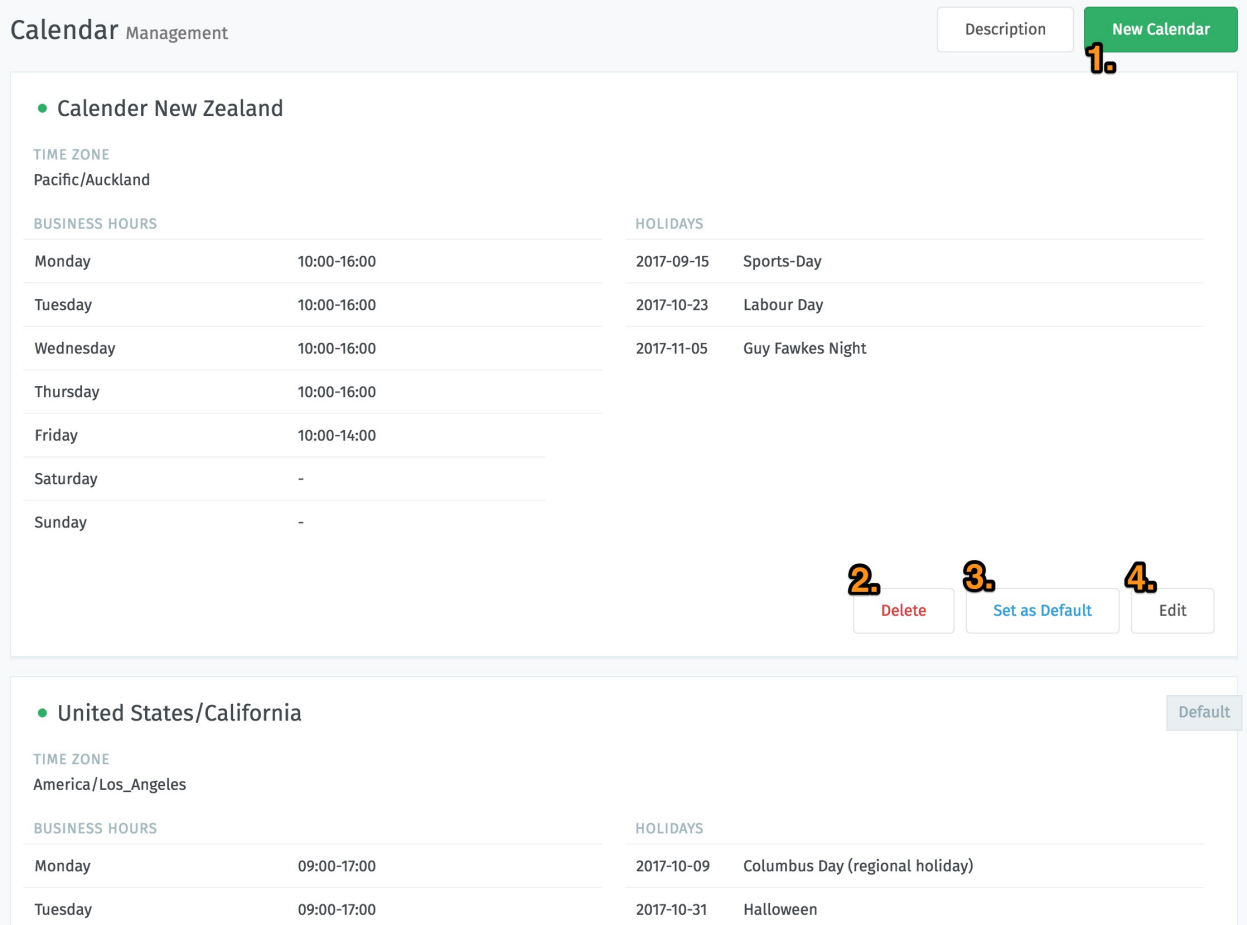
A calendar is required to:

- 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.

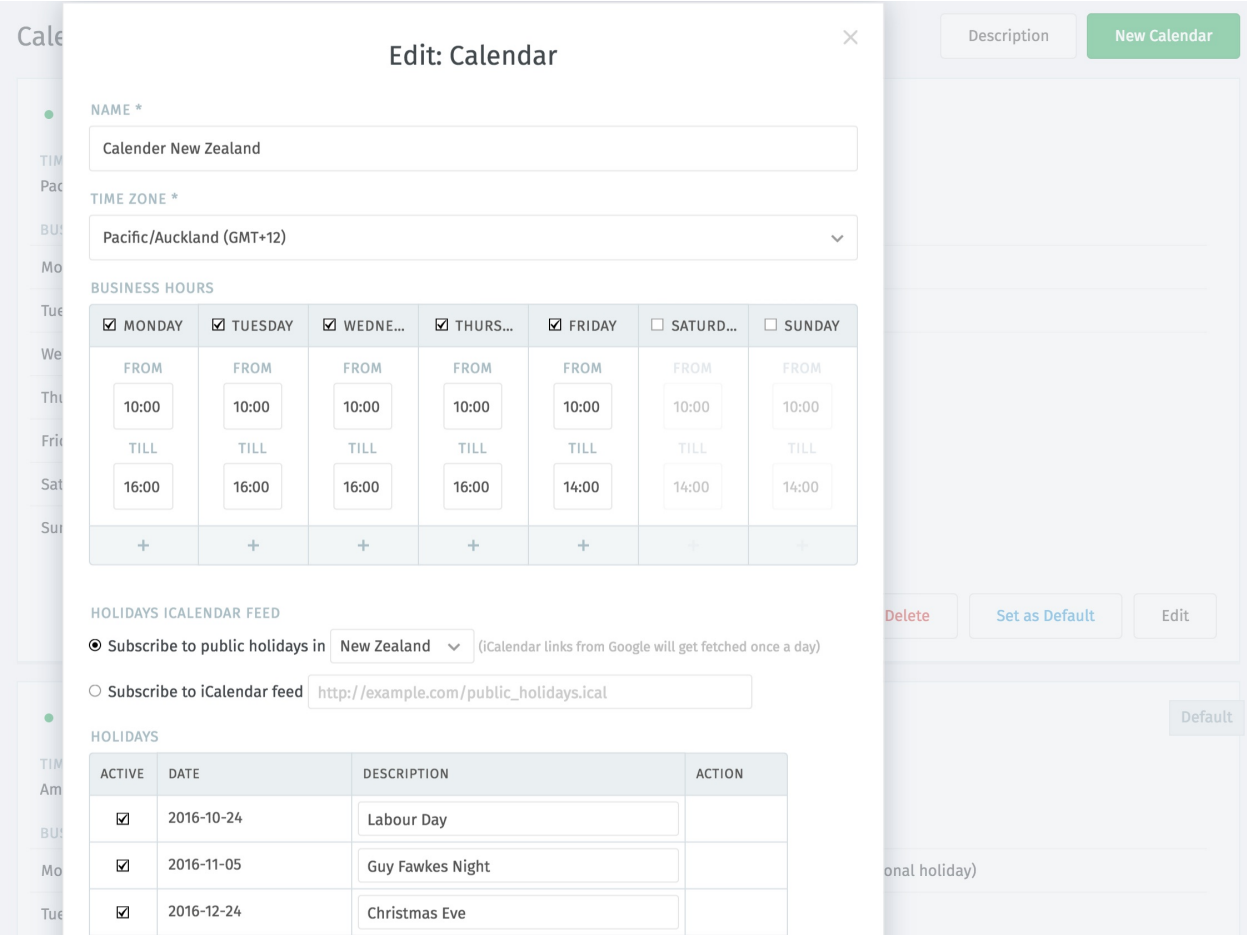
If you have customers for which you have to comply with different customer business hours, you can create several calendars. The allocation to the customer tickets can be adjusted via the SLAs.

And this is what it looks like:



All created calendars are displayed in the overview.

1. Here you can create a new calendar if agents or customers belong to another time zone.
2. Just push the delete-button to delete this specific calender - all SLAs assigned to this calendar are automatically assigned to the default calendar.
3. Pressing this button sets this calendar as the default calendar for the entire system.
4. Via this button you get to the edit-mask (same mask as in 1.):



–> 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.

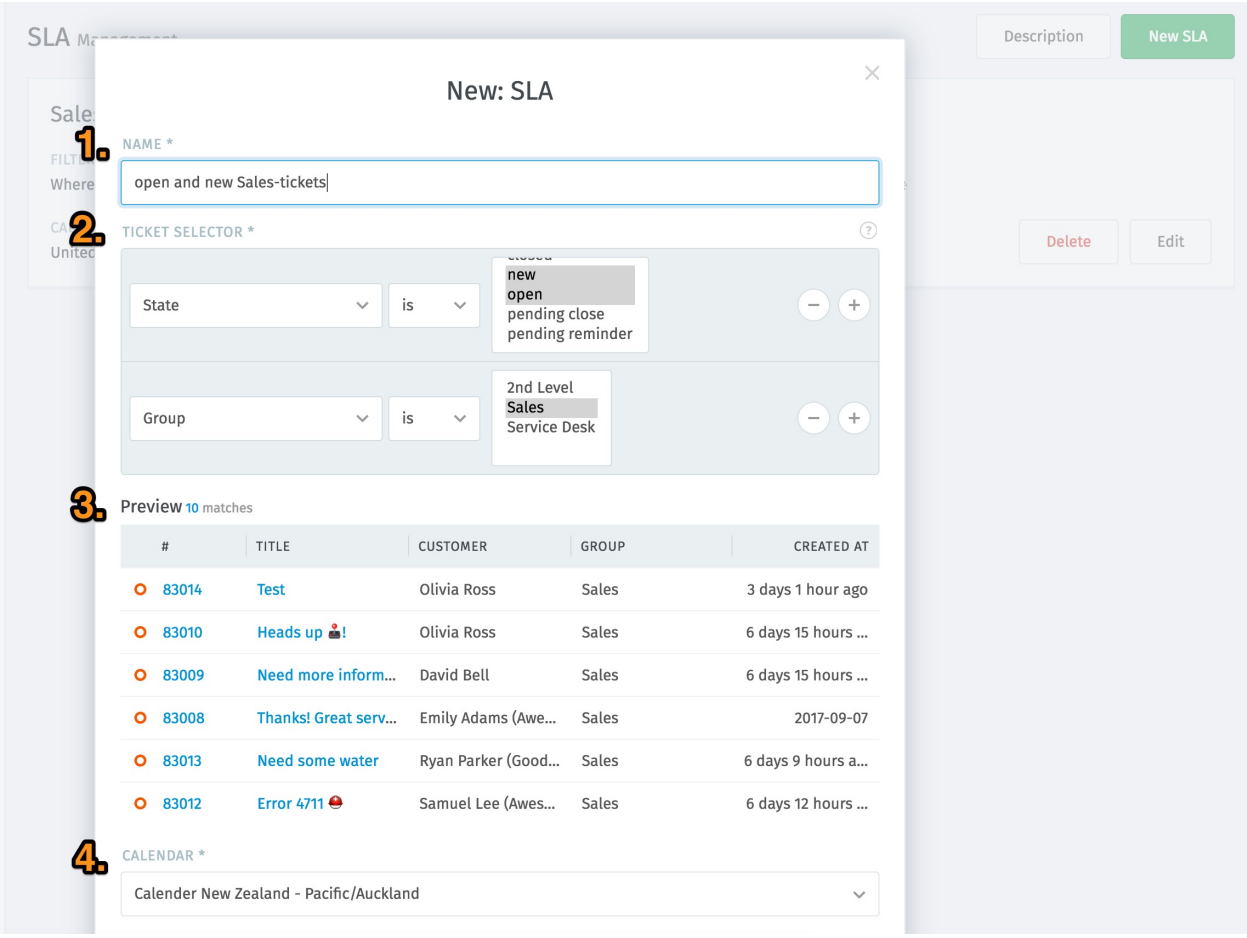
KAPITEL 10

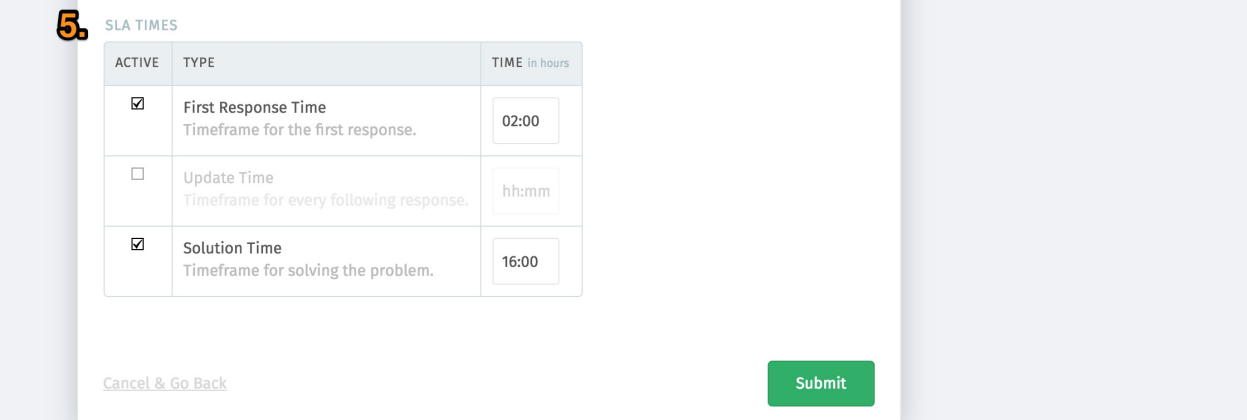
SLAs

Service levels and the respective agreements (service level agreements, SLAs) document quality pledges for IT services. SLAs are recorded and administered in here.

A calendar is required to calculate escalations or evaluations based on business hours. Define a „standard“ calendar which is valid throughout the system. Only in the specified business hours, escalation notifications are sent to agents. If you have customers for which you need to comply with different business hours, you can create multiple calendars. The customer tickets are allocated via the SLAs.

That’s how it works:





1. Give it a distinctive name
2. Specify the ticket groups for which the SLA is to apply (these can also be arbitrarily combined and thus specified)
3. In the preview you see the selection of the tickets and doublecheck wheather those are correct
4. Choose the business-calender
5. Define the SLA-Times:

   First Response Timeframe for the first response (external call, email)

Update Time Timeframe for every following response (external call, Email)

Solution Time Timeframe for solving the problem (status: closed)

It is up to you if you set one, two or all three times. When the SLA time is reached, the ticket escalates. Now all agents get notifications, which have stored the corresponding setting in their own settings profile -> notifications. The information that a ticket is escalated can be selected in the triggers as an attribute, whereby a desired action can be followed.

KAPITEL 11

Triggers

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

Hinweis: 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:

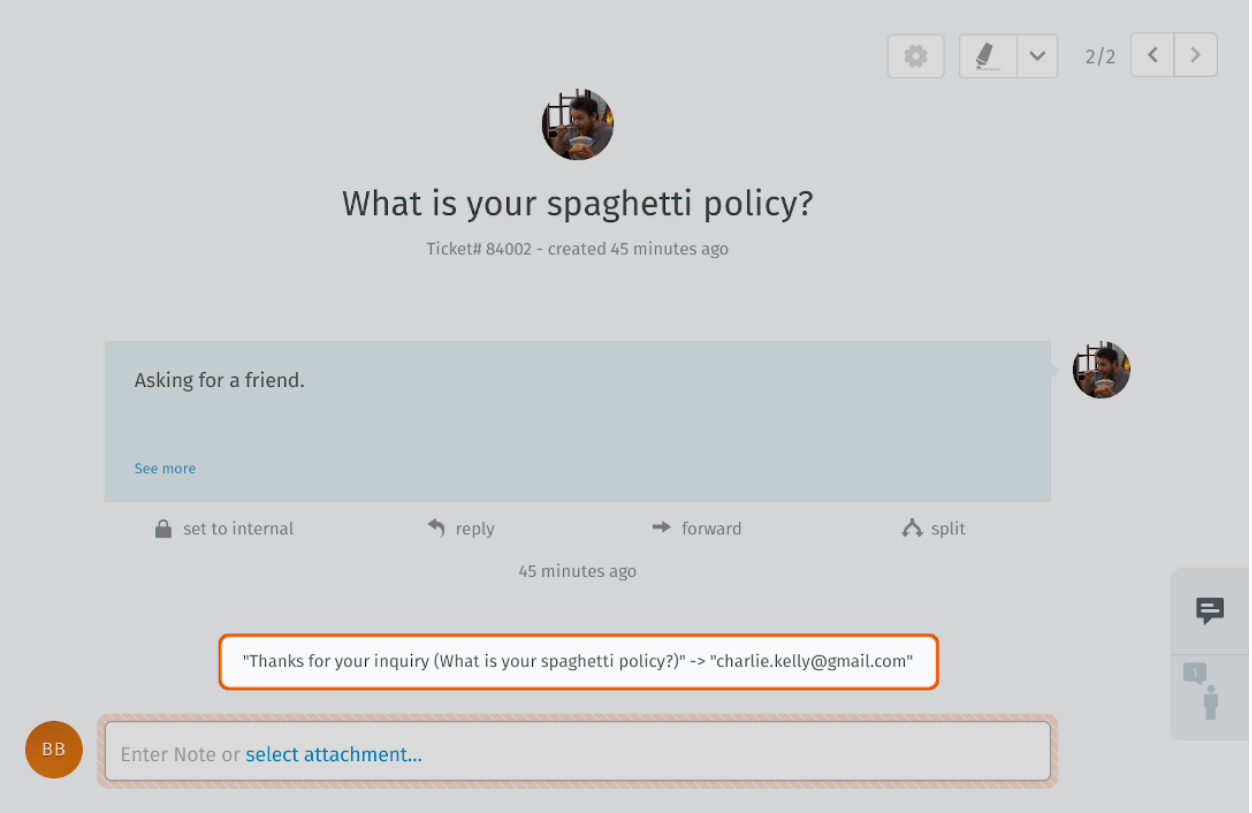
11.1 Learn by example

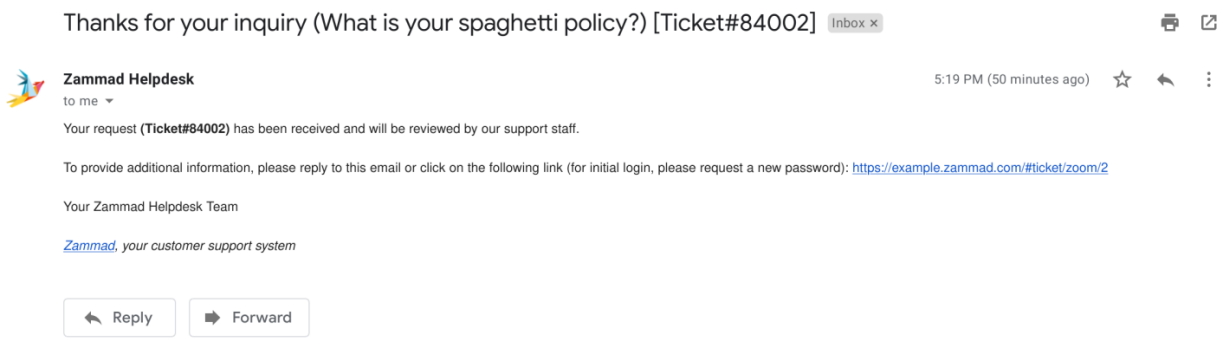
To get you up and running quickly, here are some examples of the kinds of automation tasks you can set up with triggers.

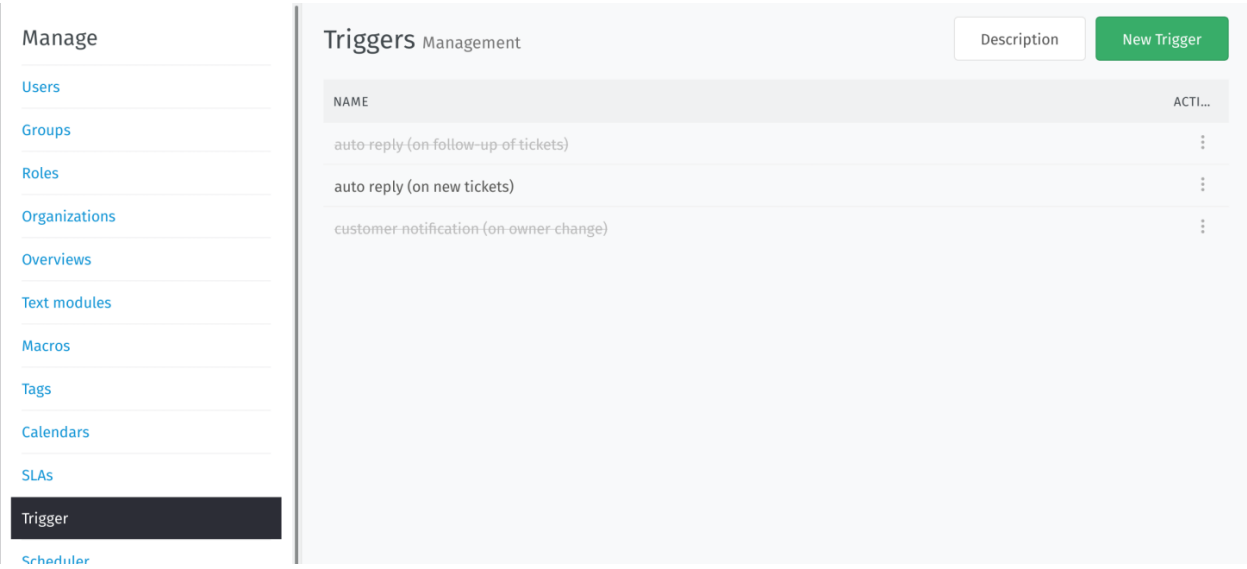
Hinweis: If they don't make sense to you, don't worry—just skip ahead to *How do they work?* to learn about all the options in detail, then come back here to see them in action.

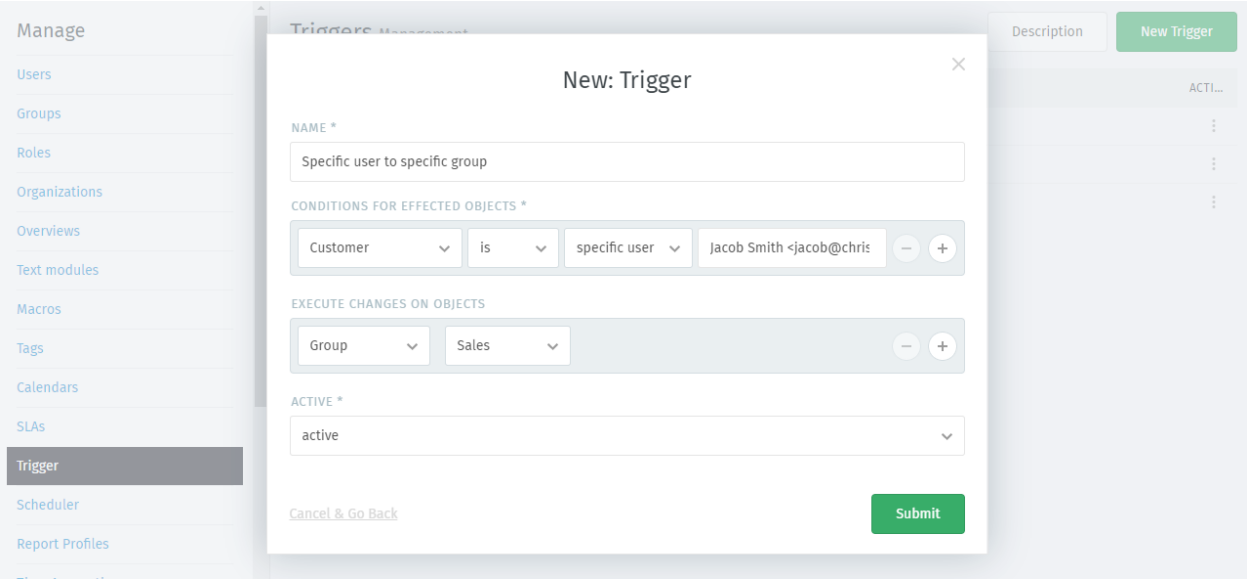
1. Any time Jacob Smith creates a ticket, assign it to the Sales group:
2. Emma Taylor is responsible for all sales internally, so if a new ticket has the word "order" in the subject, assign it to her and make sure it's set with a high priority:
3. Send an auto-reply email to *anyone* who responds to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

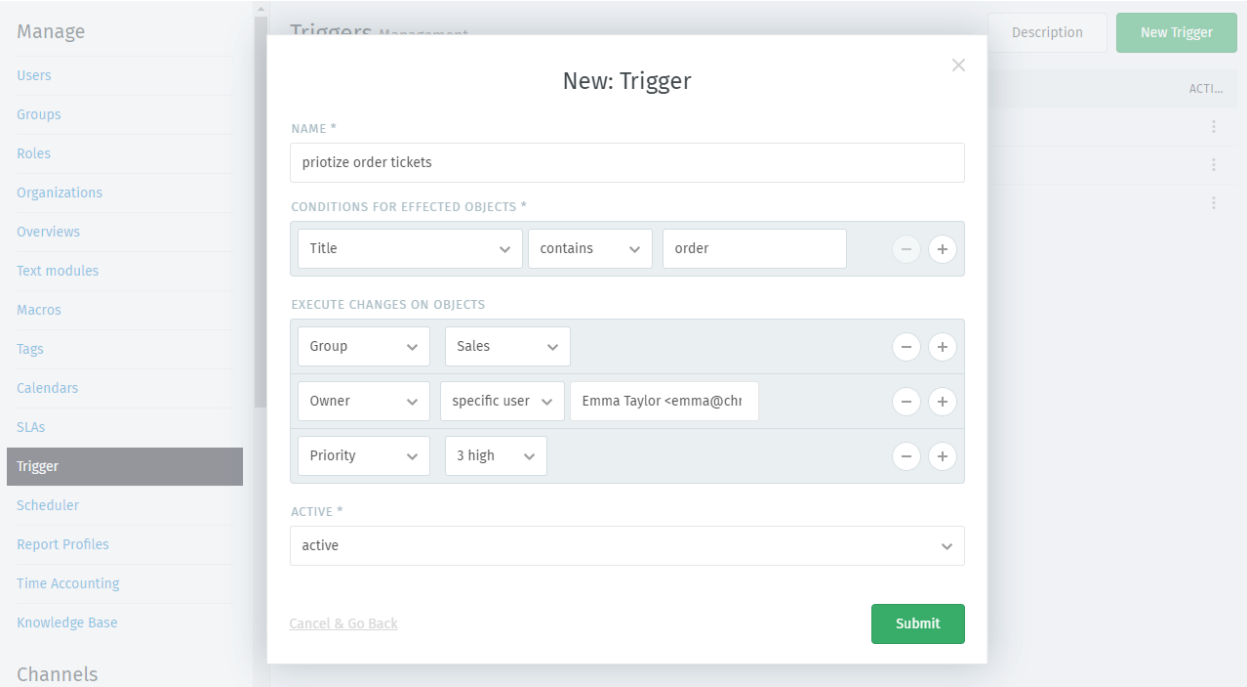
Bemerkung: Not all automated messages come from triggers!

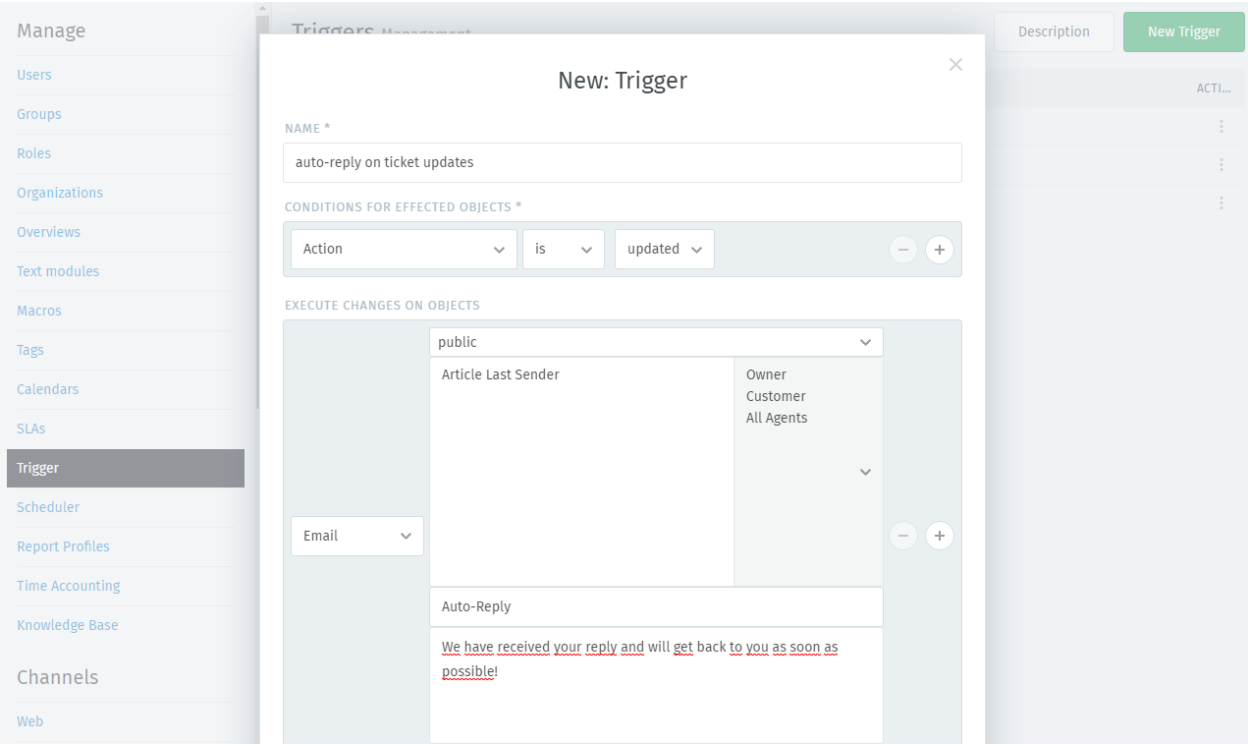

2/2
What is your spaghetti policy?
Ticket# 84002 - created 45 minutes ago
Asking for a friend.
See more
set to internal
reply
forward
split
45 minutes ago
"Thanks for your inquiry (What is your spaghetti policy?)" -> "charlie.kelly@gmail.com"
BB
Enter Note or select attachment...


Thanks for your inquiry (What is your spaghetti policy?) [Ticket#84002]
Inbox ×
Zammad Helpdesk
to me
5:19 PM (50 minutes ago)
Your request (Ticket#84002) has been received and will be reviewed by our support staff.
To provide additional information, please reply to this email or click on the following link (for initial login, please request a new password): https://example.zammad.com/#ticket/zoom/2
Your Zammad Helpdesk Team
Zammad, your customer support system
Reply
Forward


Manage
Users
Groups
Roles
Organizations
Overviews
Text modules
Macros
Tags
Calendars
SLAs
Trigger
Triggers Management
Description
New Trigger
NAME
ACTI...
auto reply (on follow-up of tickets)
auto reply (on new tickets)
customer notification (on owner change)


Manage
Users
Groups
Roles
Organizations
Overviews
Text modules
Macros
Tags
Calendars
SLAs
Trigger
Scheduler
Report Profiles
Description
New Trigger
New: Trigger
NAME *
Specific user to specific group
CONDITIONS FOR EFFECTED OBJECTS *
Customer
is
specific user
Jacob Smith <jacob@chris
EXECUTE CHANGES ON OBJECTS
Group
Sales
ACTIVE *
active
Cancel & Go Back
Submit


Manage
Users
Groups
Roles
Organizations
Overviews
Text modules
Macros
Tags
Calendars
SLAs
Trigger
Scheduler
Report Profiles
Time Accounting
Knowledge Base
Channels
Description
New Trigger
ACTI...
New: Trigger
NAME *
priotize order tickets
CONDITIONS FOR EFFECTED OBJECTS *
Title
contains
order
EXECUTE CHANGES ON OBJECTS
Group
Sales
Owner
specific user
Emma Taylor <emma@chr
Priority
3 high
ACTIVE *
active
Cancel & Go Back
Submit


Manage
Users
Groups
Roles
Organizations
Overviews
Text modules
Macros
Tags
Calendars
SLAs
Trigger
Scheduler
Report Profiles
Time Accounting
Knowledge Base
Channels
Web
Description
New Trigger
ACTI...
New: Trigger
NAME *
auto-reply on ticket updates
CONDITIONS FOR EFFECTED OBJECTS *
Action
is
updated
EXECUTE CHANGES ON OBJECTS
public
Article Last Sender
Owner
Customer
All Agents
Email
Auto-Reply
We have received your reply and will get back to you as soon as possible!

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*.

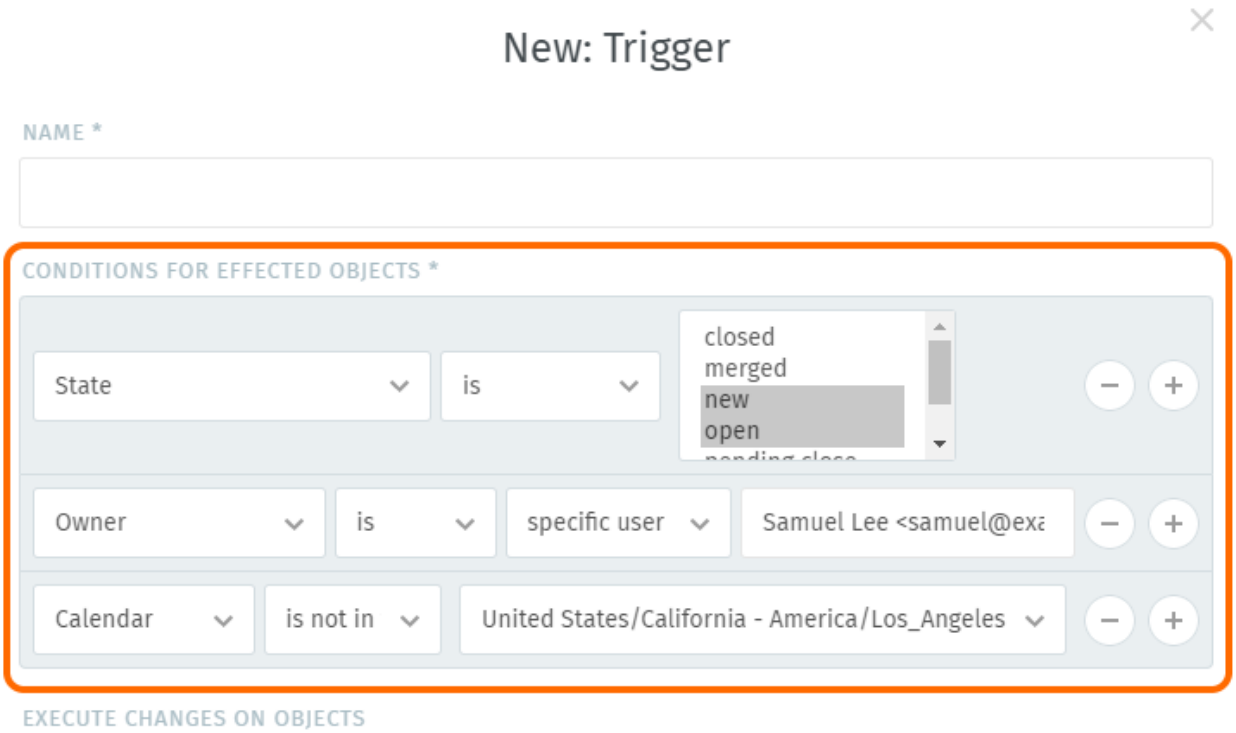
11.2 How do they work?

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.

11.2.1 Conditions

When creating a trigger, define your conditions here:



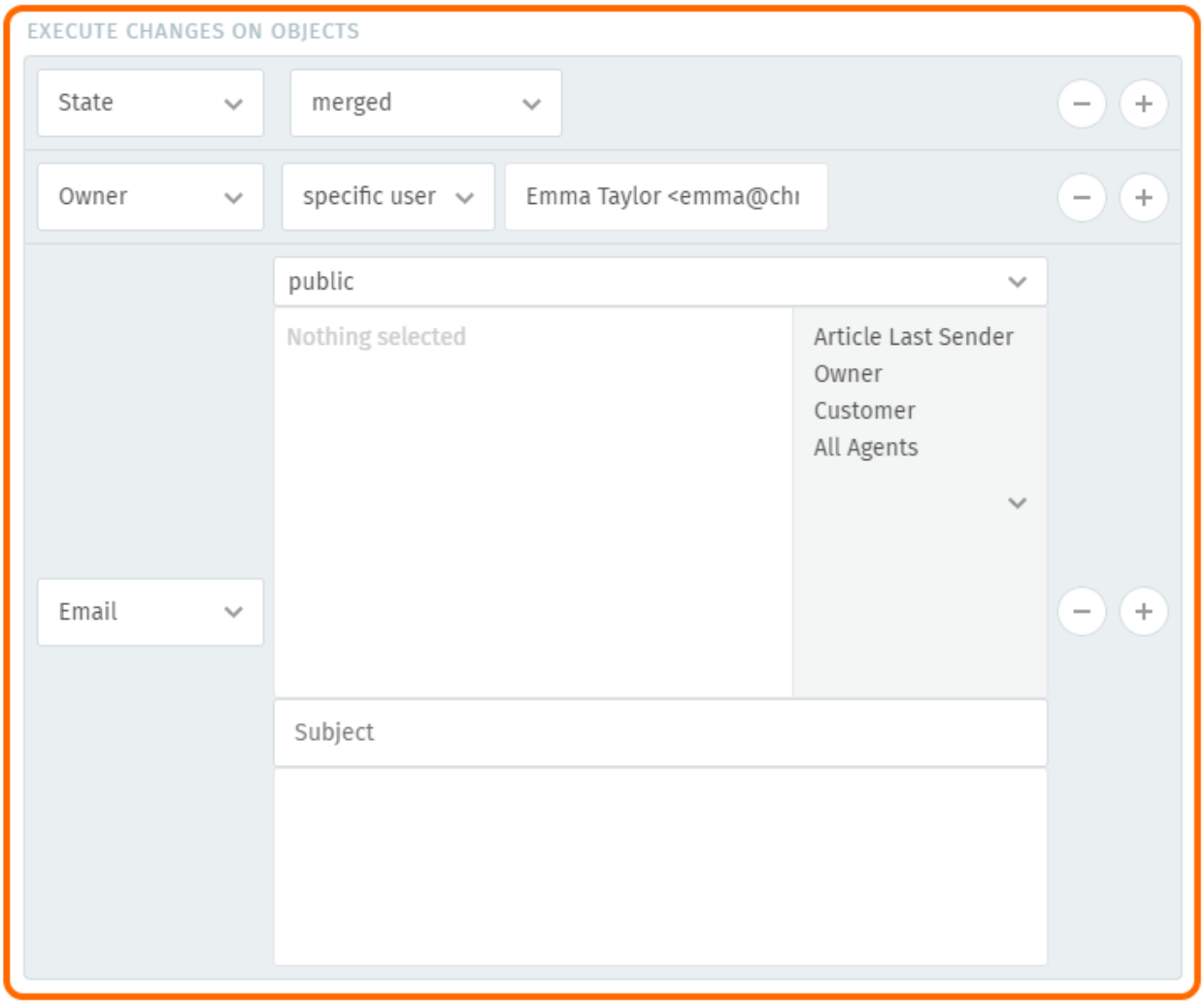
Trigger conditions are **and-selectors** and thus all conditions must apply as configured for the trigger to fire. You can configure triggers to fire based on the properties of:

- **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.)

11.2.2 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** or **Escalation at**) can be specified in *absolute* or *relative* terms.
- **Send an email or SMS** Either to the customer, the agent who owns the ticket, or every agent in the system.
- ***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)

Bemerkung: 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.

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

11.3 Webhooks

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

Bemerkung: 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.

11.3.1 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

11.3.2 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)

11.3.3 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",
      "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,
```


(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)


```
      "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,
      "domain_assignment": false,
      "id": 3,
      "member_ids": [
        8,
        6,
        7
      ],
      "members": [
        "emily@example.com",
```


(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)


```
        "anna@example.com",
        "samuel@example.com"
      ],
      "name": "Awesome Customer Inc.",
      "note": "Global distributor of communication and security products, electrical␣
↪and electronic wire &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": "-",
      "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",
```


(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)


```
    "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": {
          "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": "-",
```


(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)


```
      "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,
    "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": [
```


(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)


```
        "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
  }
}
```


Bemerkung:

- 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`

11.4 Limitations

It's important to understand when a trigger can be used and when it's better to use e.g. Zammads *Scheduler* or postmaster filter.

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 Zammads *Scheduler*.

11.5 System Notifications

Bemerkung: 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.

11.5.1 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.

11.5.2 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.

KAPITEL 12

Scheduler

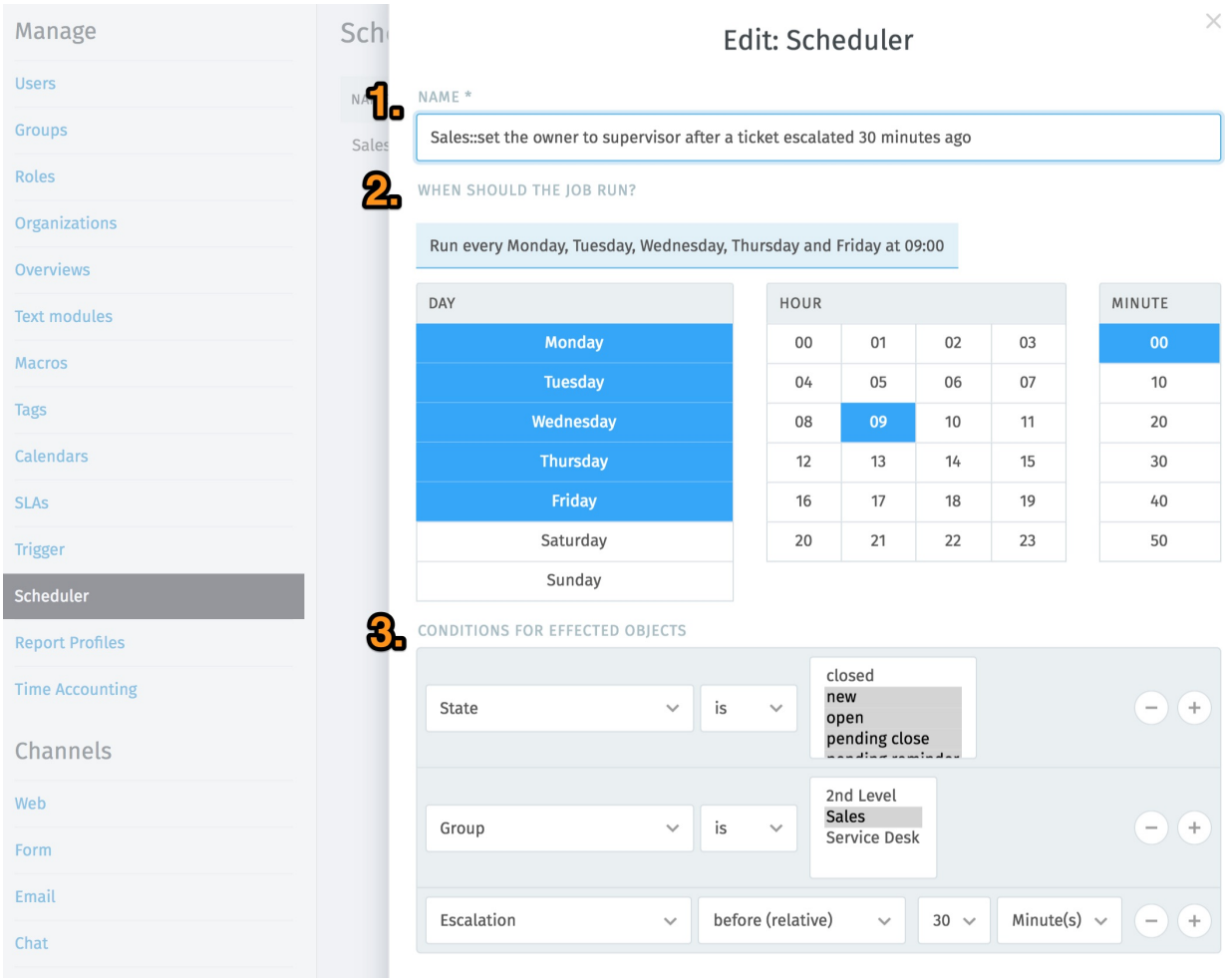
The scheduler performs time-based automated actions. You can set up your own schedulers, configure at which points in time they should run, set up conditions to determine which tickets they should affect, and then configure the actions that you want to be executed on these tickets.

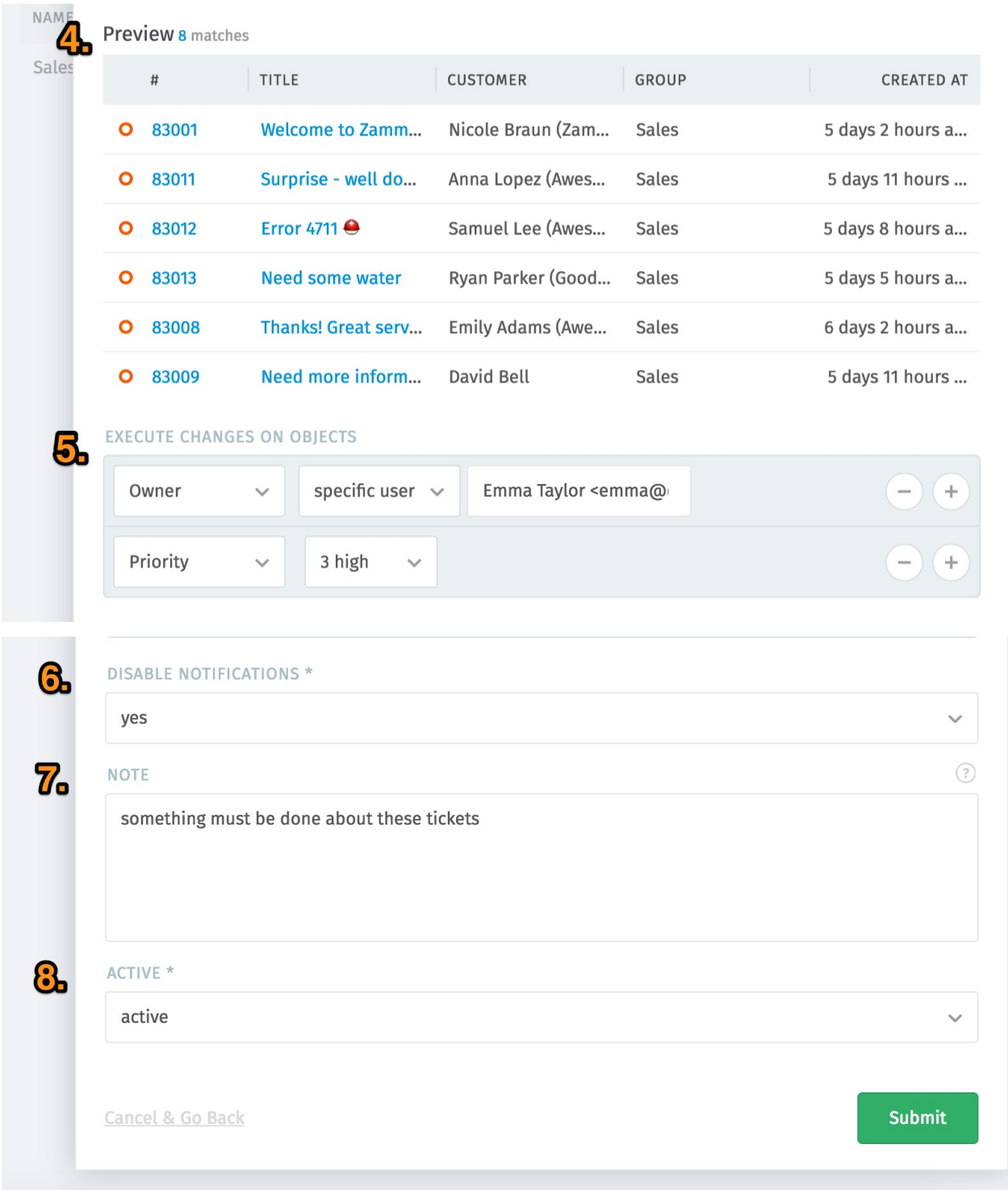
Bemerkung: Schedulers with *Action: Delete* are currently the only way in the Zammad front end to permanently delete tickets. This limitation is intentional as Zammad is designed to be revision-proof. A possible use case for such a scheduler is to delete spam tickets some time after creation (e.g. 30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

Warnung: 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.

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

12.1 Add a new scheduler





1. *Name*: choose a name for the scheduler.
2. *When should the job run*: choose the points in time using UTC timezone when the scheduler should run.
3. *Conditions for affected objects*: determine the ticket attributes (conditions) to limit on which tickets the actions configured in step 5 are to be performed.

4. *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.
5. *Execute changes on objects*: determine the changes to be made to the ticket.
6. *Disable notifications*: by default, actions triggered by schedulers won't send notifications. You can override this here by setting this to *no*.
7. *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.
8. *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. (*UTC*), all tickets which:

- are not closed 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.

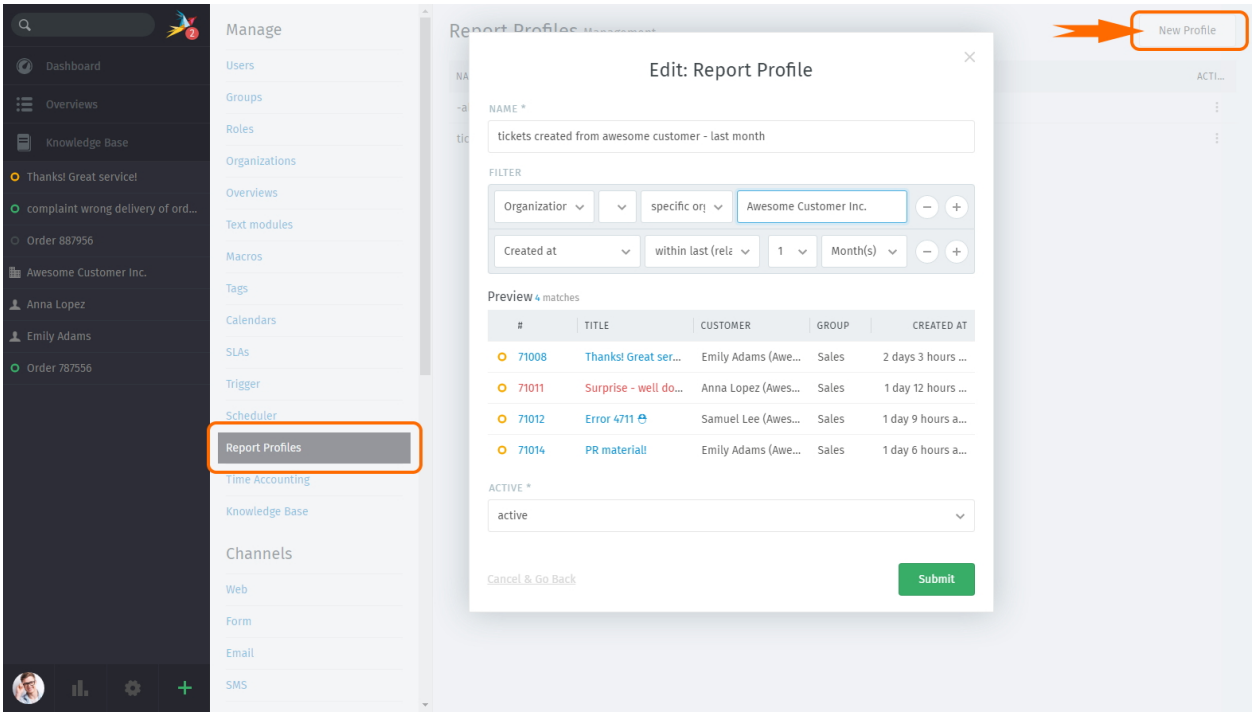
As a supervisor in the Sales group, this enables Emma to intercept and process escalated tickets.

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

KAPITEL 13

Report-Profiles

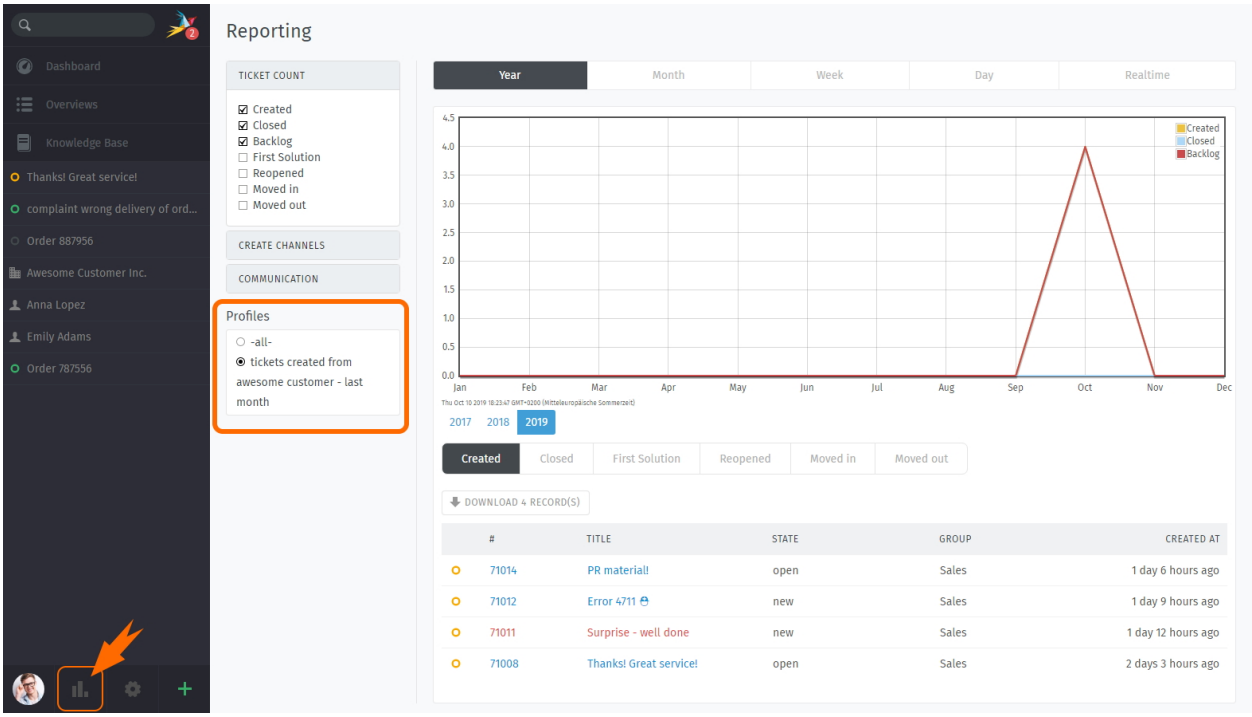
Report profiles are used to restrict / filter report-results. The idea of the profiles is to limit the number of tickets and determine the type of tickets you want to analyze. You can create any number of profiles in the Admin Interface in the „Report Profile“ area. The edit-mask looks like this:



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

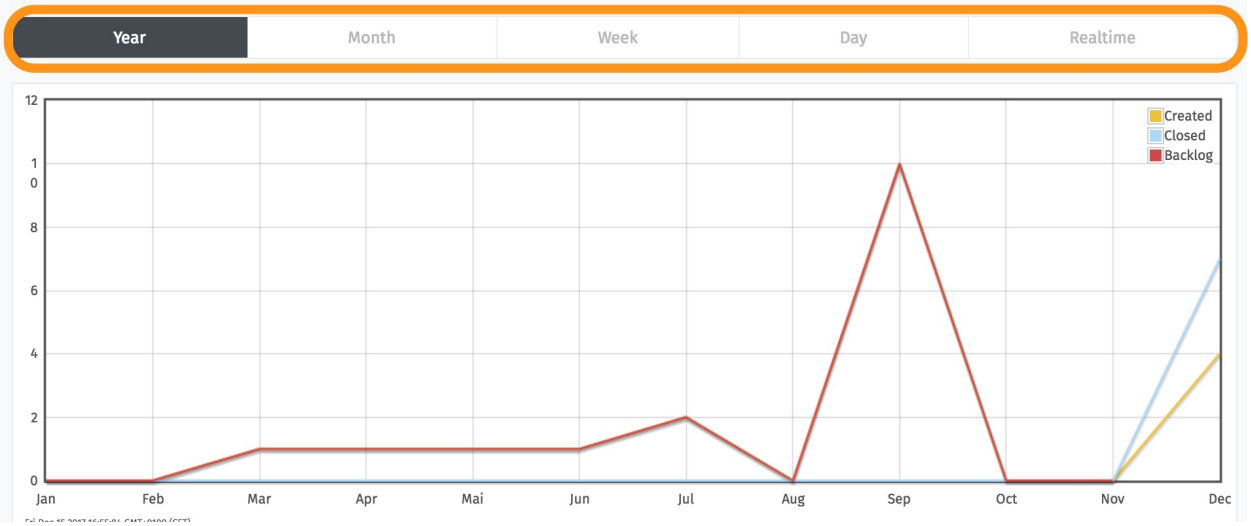
The filters can be combined with each other as desired. The filters build on each other, which means that they are further restricted per additional attribute.

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

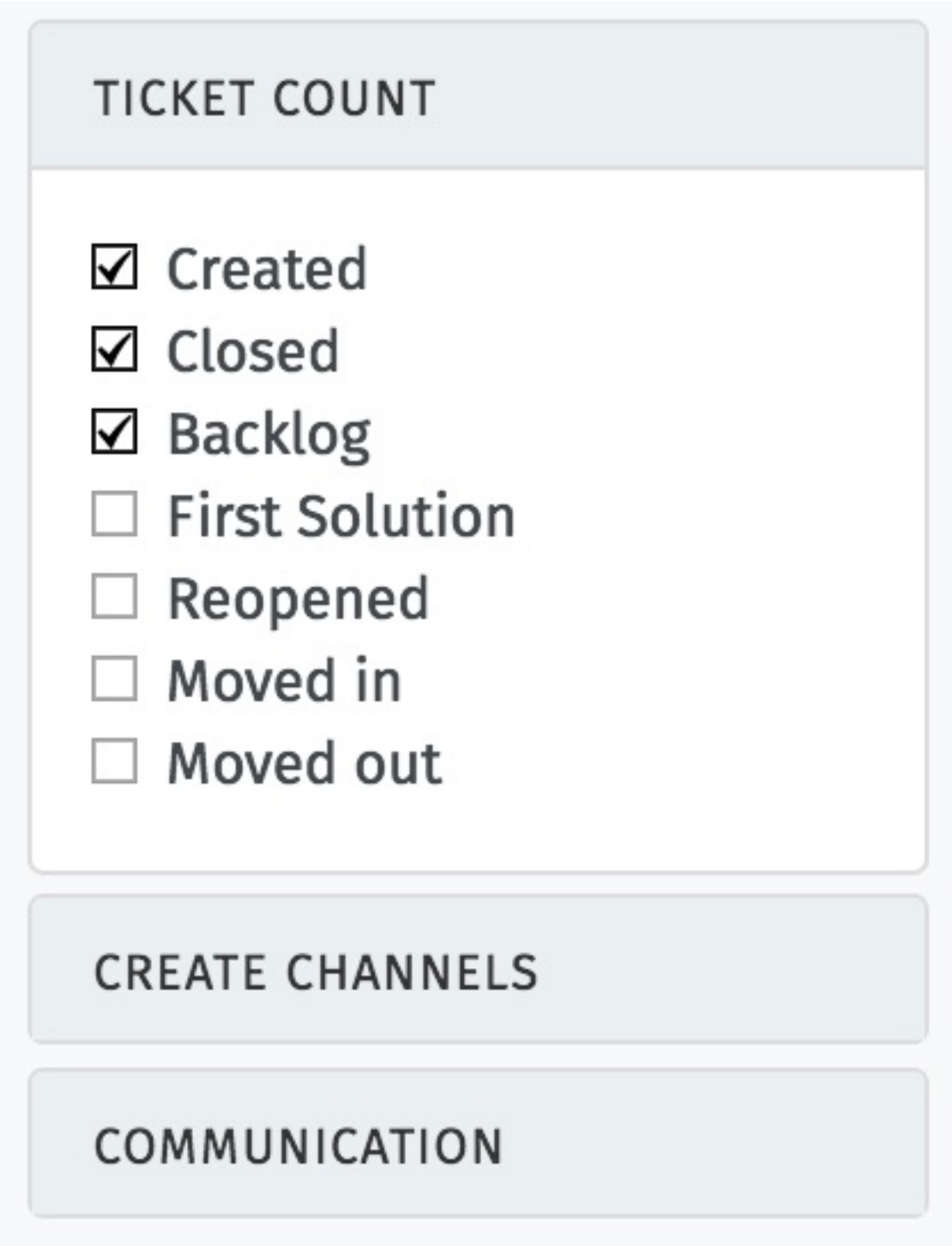


13.1 Further information about the reporting:

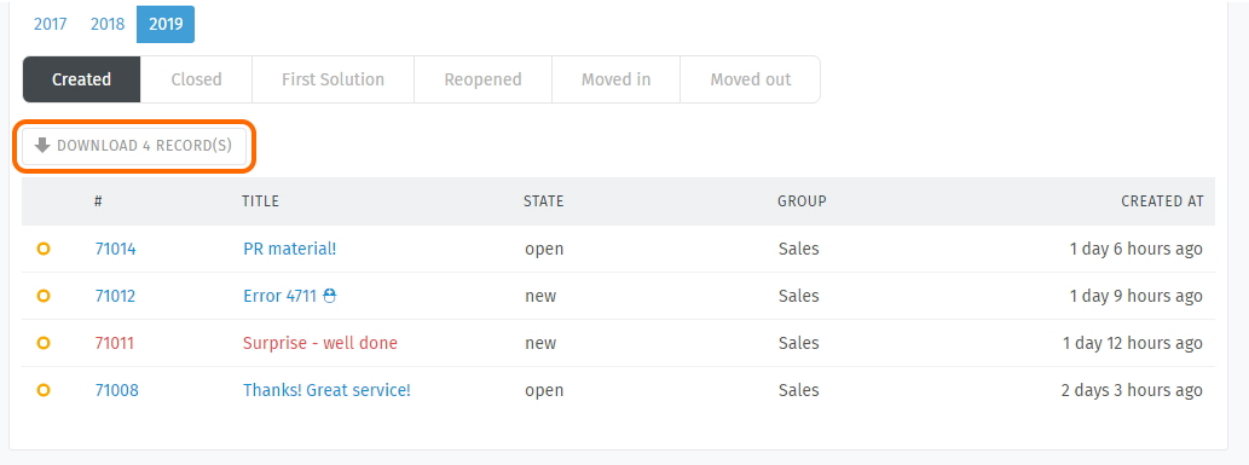
The time period and time interval can be changed with one click (the graphic adapts itself directly):



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:

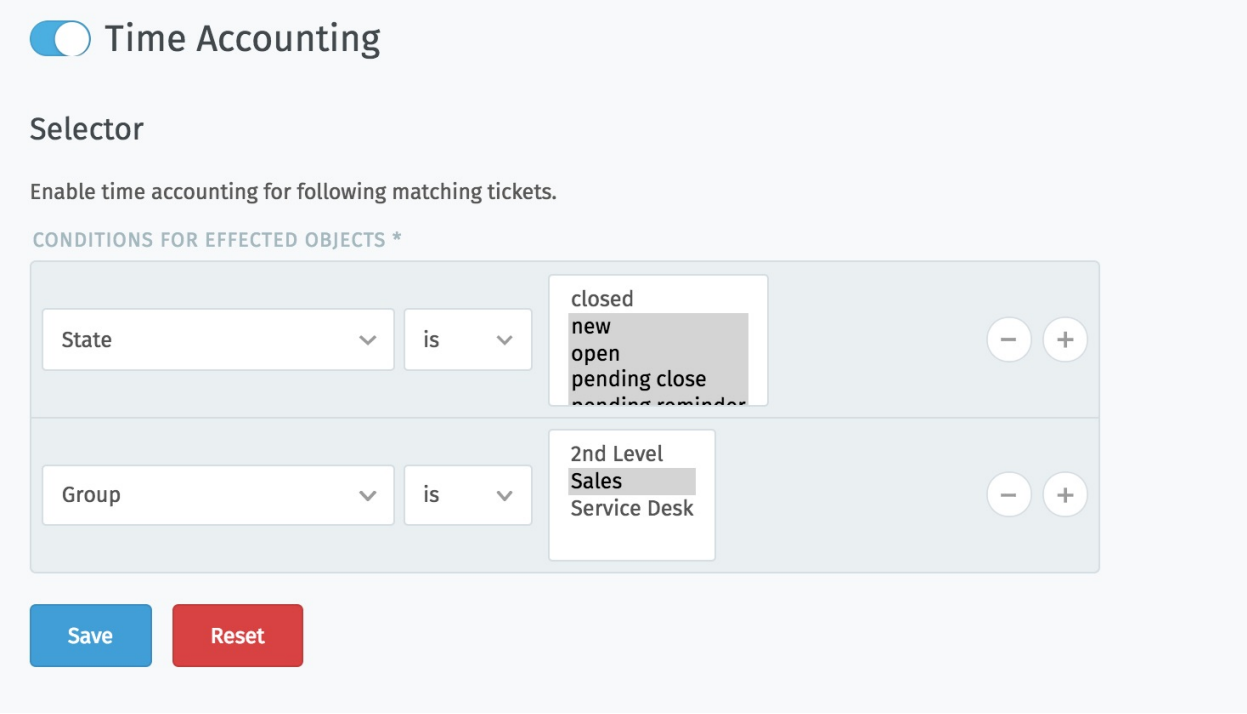

2017
2018
2019
Created
Closed
First Solution
Reopened
Moved in
Moved out
DOWNLOAD 4 RECORD(S)
TITLE STATE GROUP CREATED AT
71014 PR material! open Sales 1 day 6 hours ago
71012 Error 4711 new Sales 1 day 9 hours ago
71011 Surprise - well done new Sales 1 day 12 hours ago
71008 Thanks! Great service! open Sales 2 days 3 hours ago

KAPITEL 14

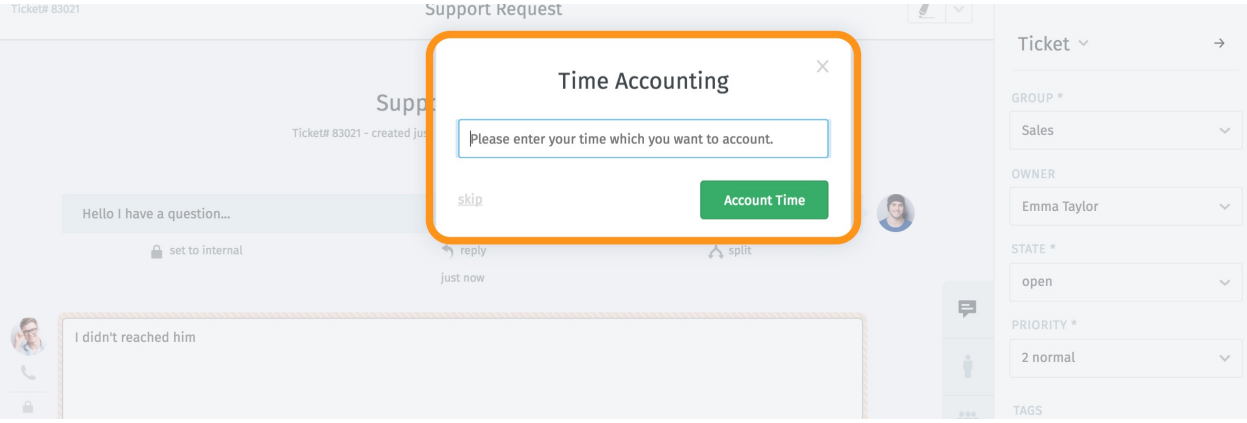
Time Accounting

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

It can be set for which tickets the time accounting is activated - that means for which tickets you can enter, how much time was needed. It's possible to set multiple conditions (these can also be arbitrarily combined and thus specified):

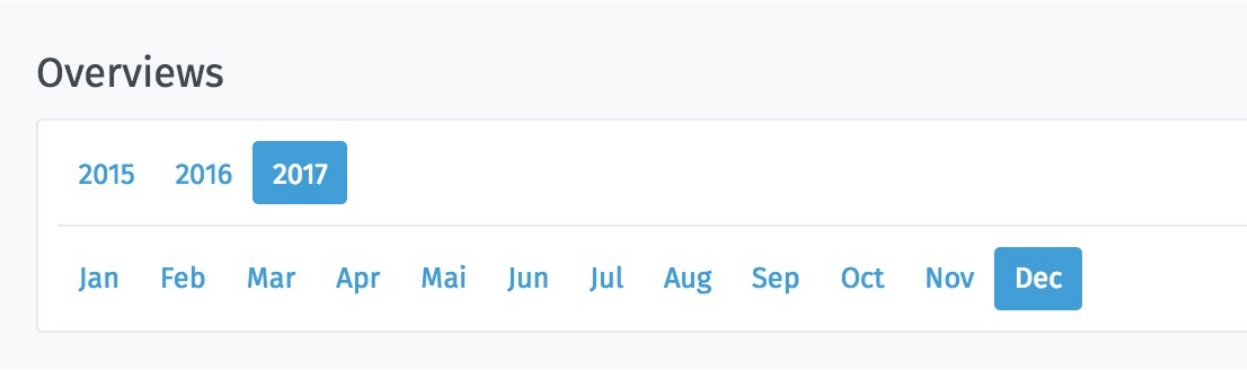


For all set cases (in this example, this is for each „Update“ of tickets that are not closed and in the group ‚Sales‘) the following field appears for entering the invested time:



In the admin area there are different possibilities to get an overview of the accounted times.

There is a monthly overview for all counted times. By clicking on the desired year and month you can see all corresponding tickets:



The tickets can be arranged according to customers and organization. All views can be downloaded as a CSV file (click on the down arrow).

Ticket

TICKET#	TITLE	CUSTOMER	ORGANIZATION	AGENT	TIME UNITS	TIME UNITS TOTAL	CREATED AT	CLOSED AT
83018	Inquiry	David ...	-	Christo...	10.0	10.0	3 days ...	3 days ...
83017	anothe...	Christo...	Chrispr...	Christo...	15.0	15.0	4 days ...	

Customer

CUSTOMER	ORGANIZATION	TIME UNITS
david@example.com		10.0
chris@chrispresso.com	Chrispresso Inc.	15.0

Organization

ORGANIZATION	TIME UNITS
Chrispresso Inc.	15.0

KAPITEL 15

Knowledge Base

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

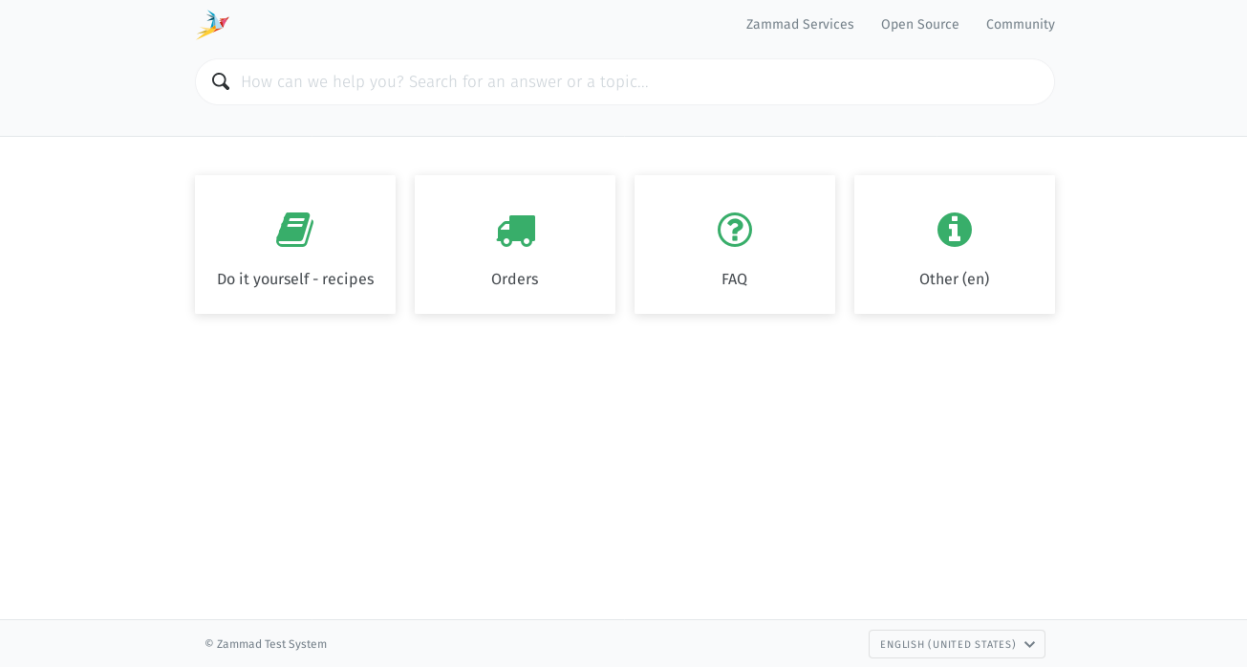


Abb. 1: See a live demo at https://support.zammad.com/help.

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

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

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.

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

15.1 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

15.2 Setup

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

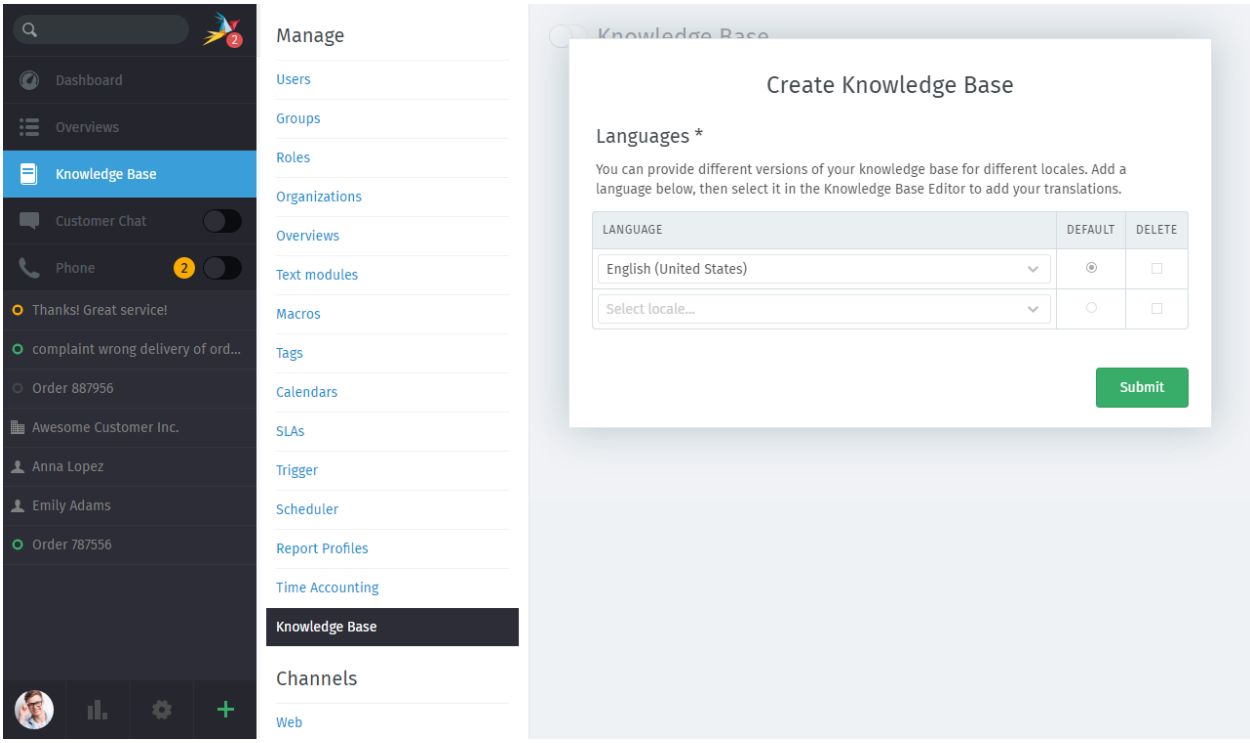


Abb. 2: You must choose at least one. (Don't worry, you can always change them later!)

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

15.2.1 Theme

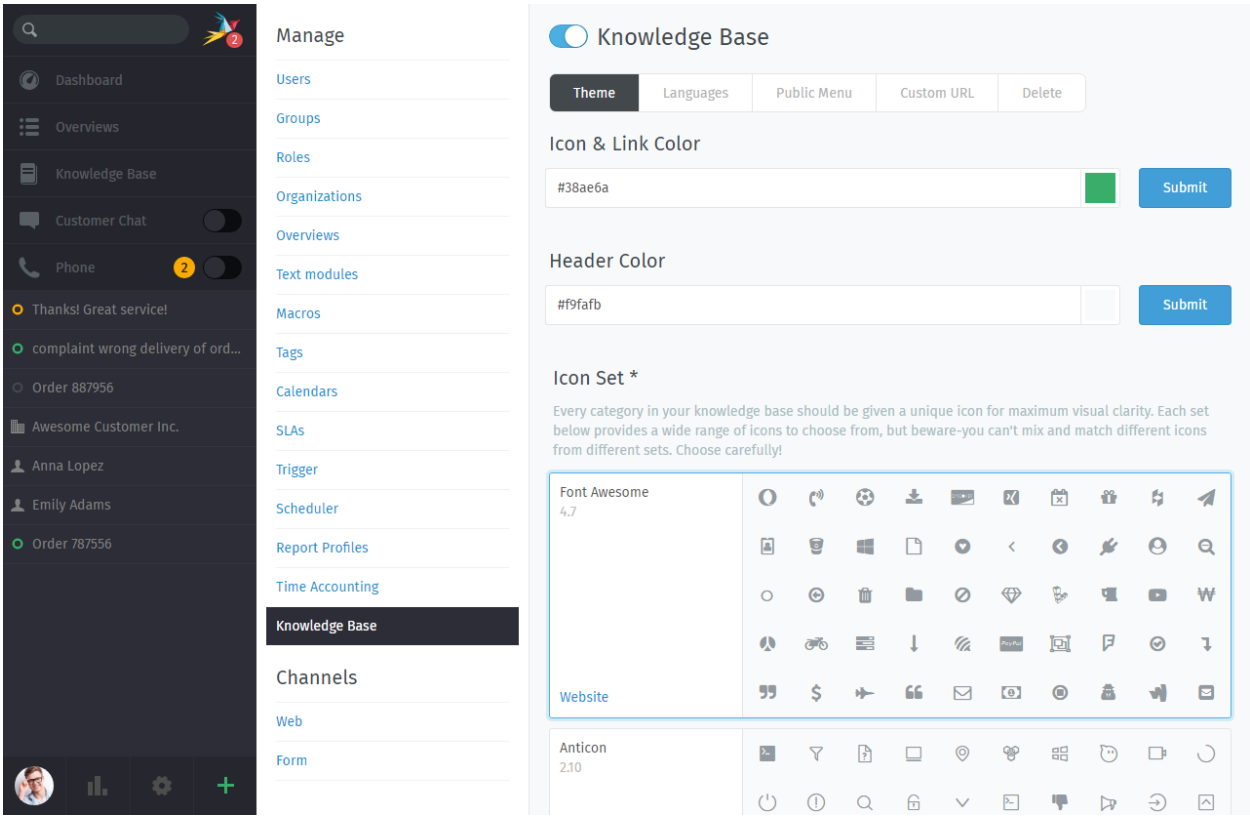


Abb. 3: Customize the appearance of the knowledge base.

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.

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:

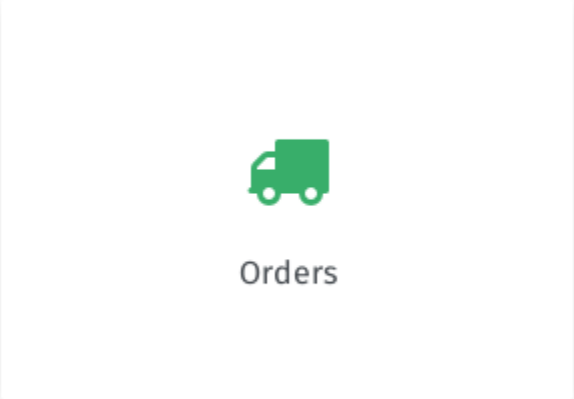
> **Warnung:** Re-assigning icons on all of your categories is tedious work. It's advisable to explore your options early to avoid having to change your mind down the road.

15.2.2 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.



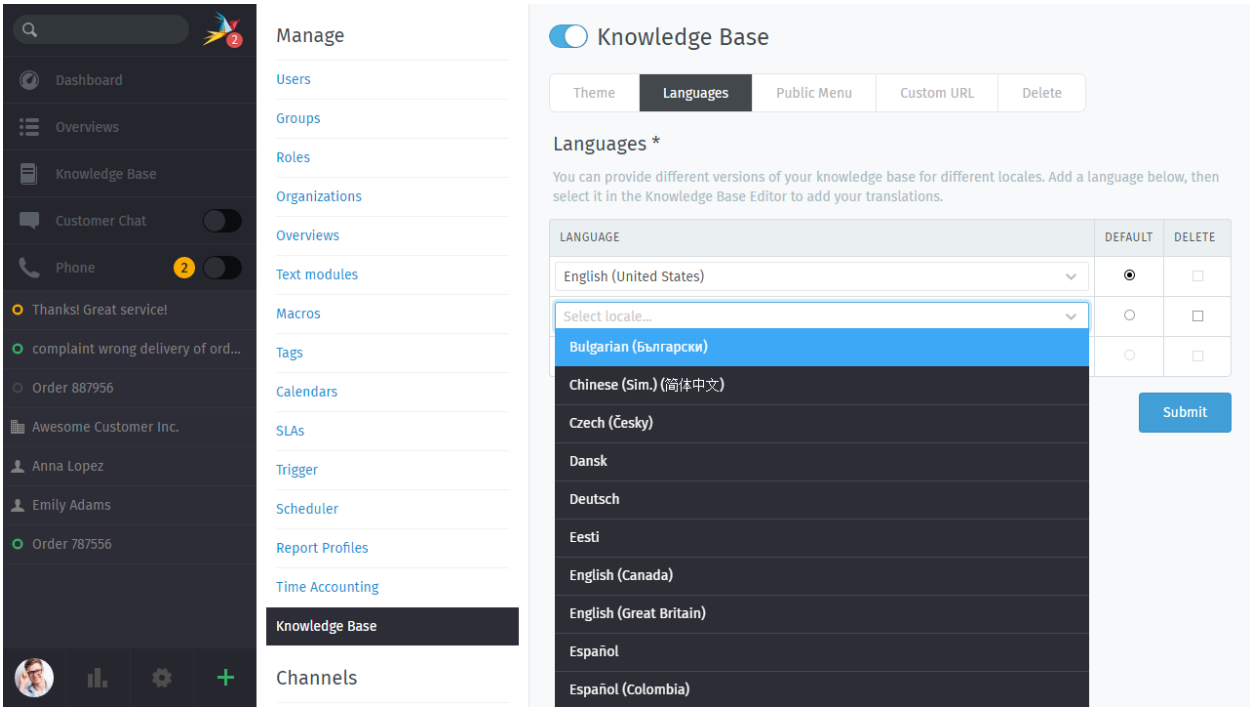


Abb. 4: Add or remove locales, or reassign the default.

15.2.3 Public Menu

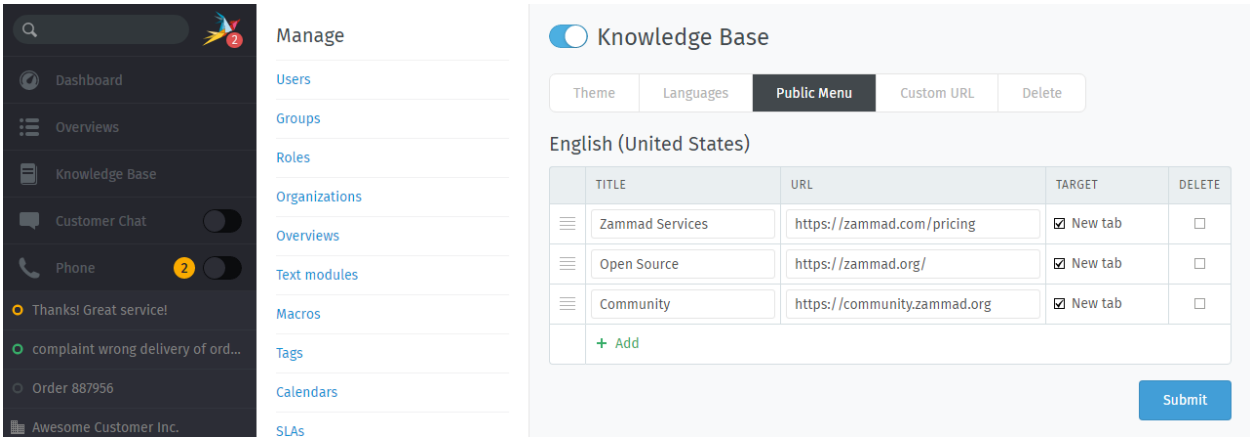
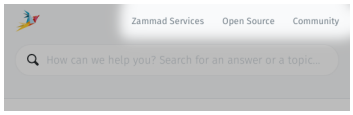


Abb. 5: Customize the nav menu that appears in the header of the knowledge base.

Hinweis: `Target` let's you decide if you want the link to be opened within a new browser tab upon click or not.

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:



15.2.4 Custom URL

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

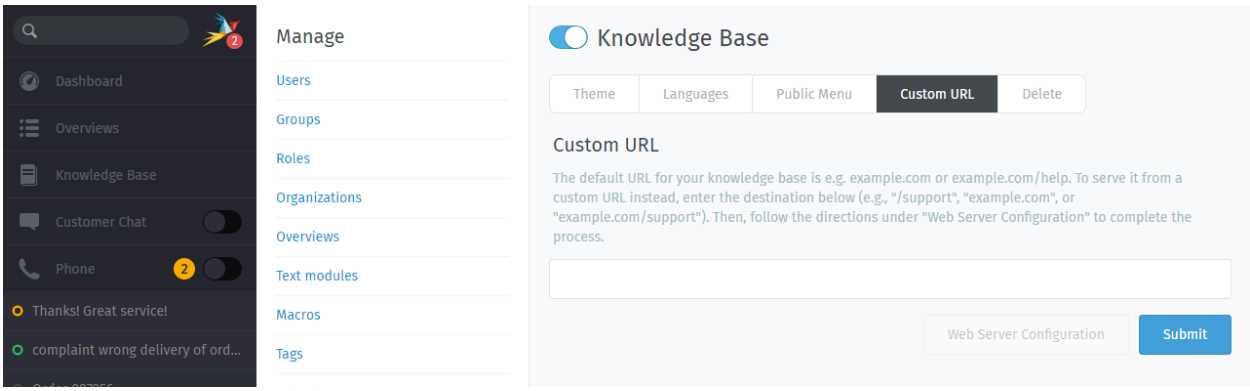


Abb. 6: Relocate the knowledge base to the URL of your choosing.

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).

15.2.5 Delete

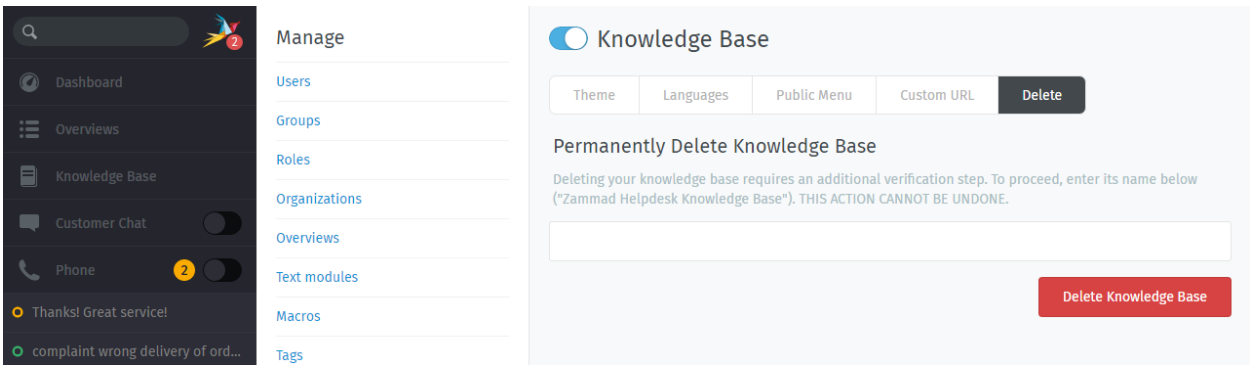


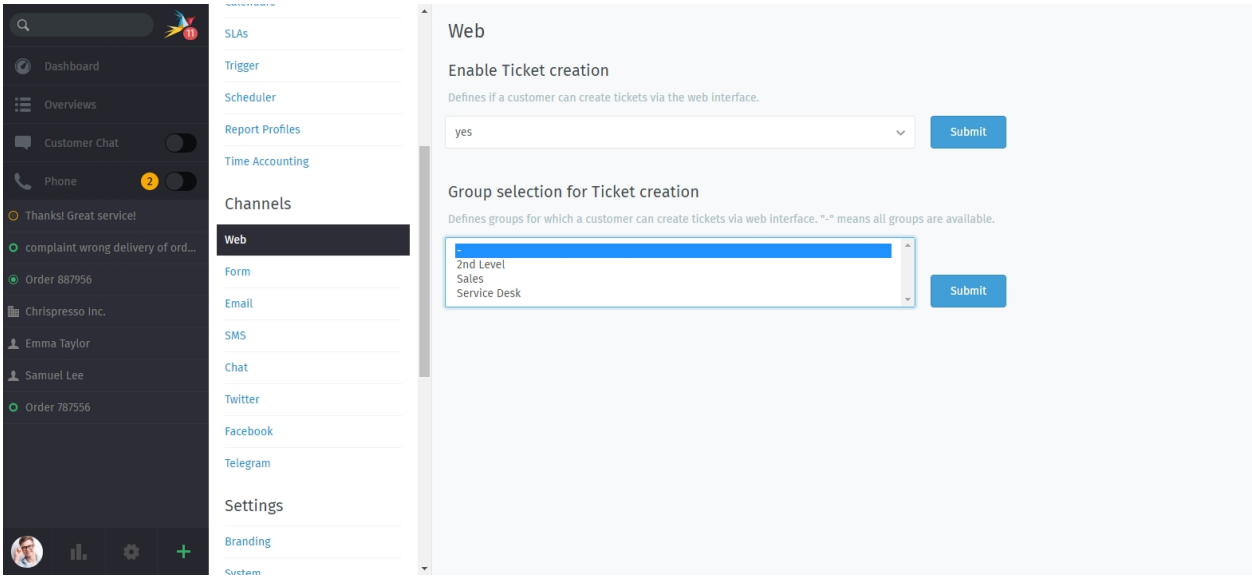
Abb. 7: Permanently delete the knowledge base and all the articles within it.

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.

KAPITEL 16

Web

You can define whether a customer can create tickets via the web interface or not. If they can you can furthermore define groups for which a customer can create tickets via web interface. „-“ means all groups are available.



That means for example: if you are working with a dispatcher, you could implement a group „Income“ and only select this group here. Thus, all incoming tickets are routed to this group and can be dispached to other groups from here.

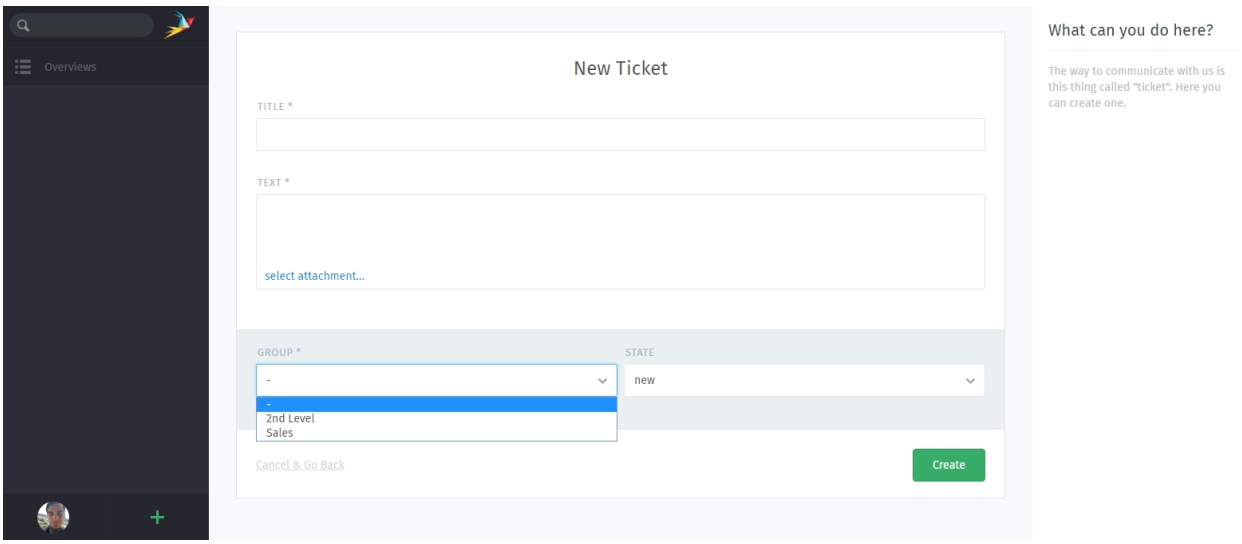


Abb. 1: In this example we selected „Sales“ and „2nd Level“ to be selectable for the customer.

KAPITEL 17

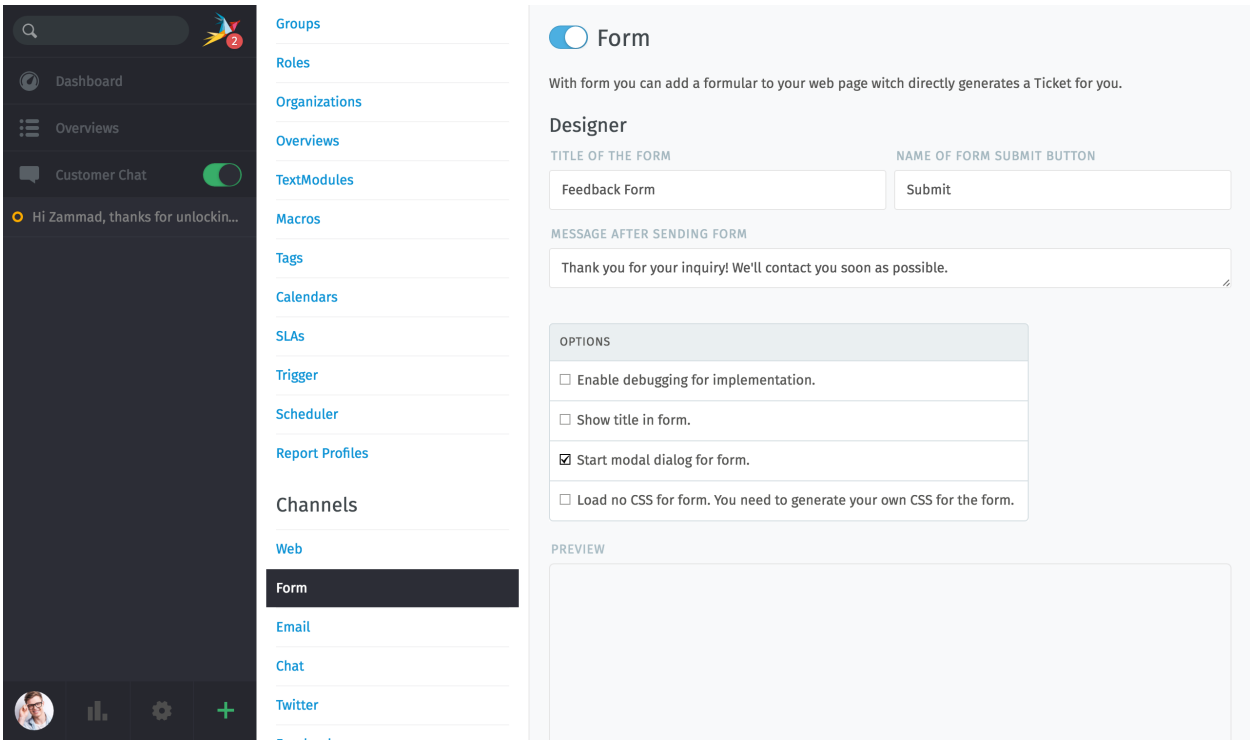
Forms

Hinweis: At the moment it is not possible to add more (or custom) fields to a Zammad Form.

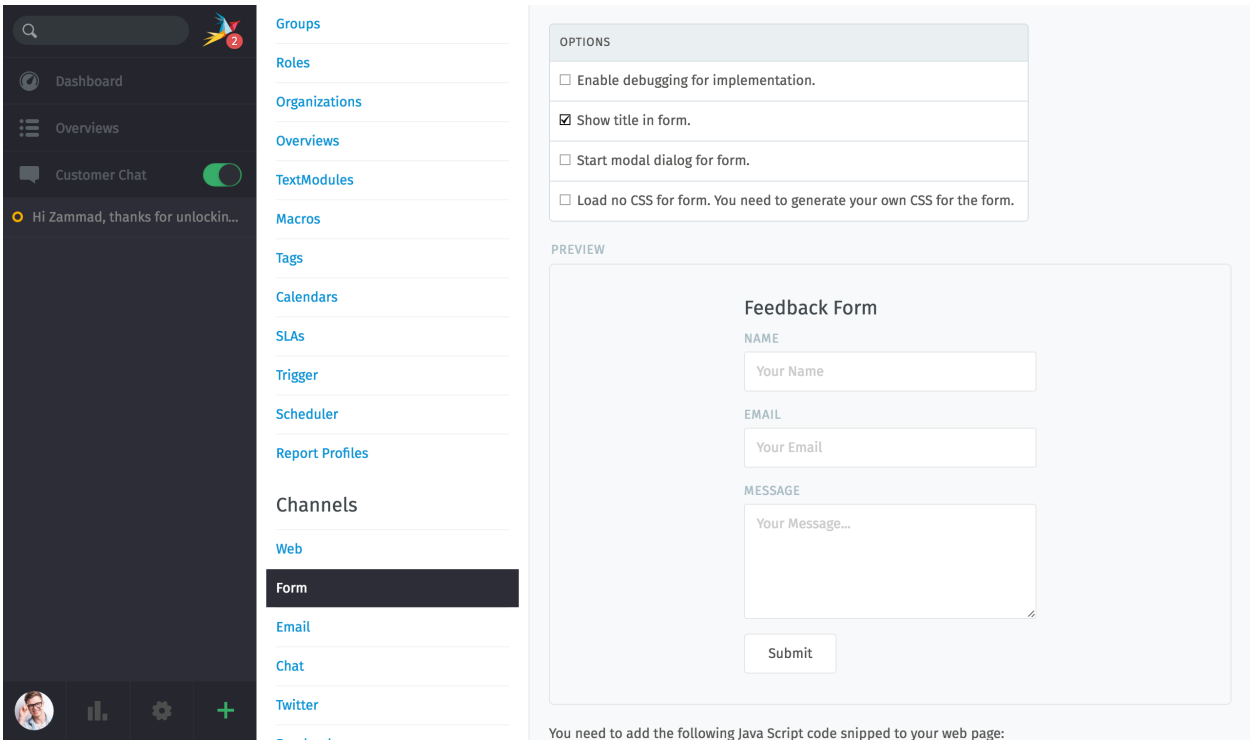
Bemerkung: The form is currently limited to one form per instance.

Feedback or contact forms are quite often used on websites. 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.

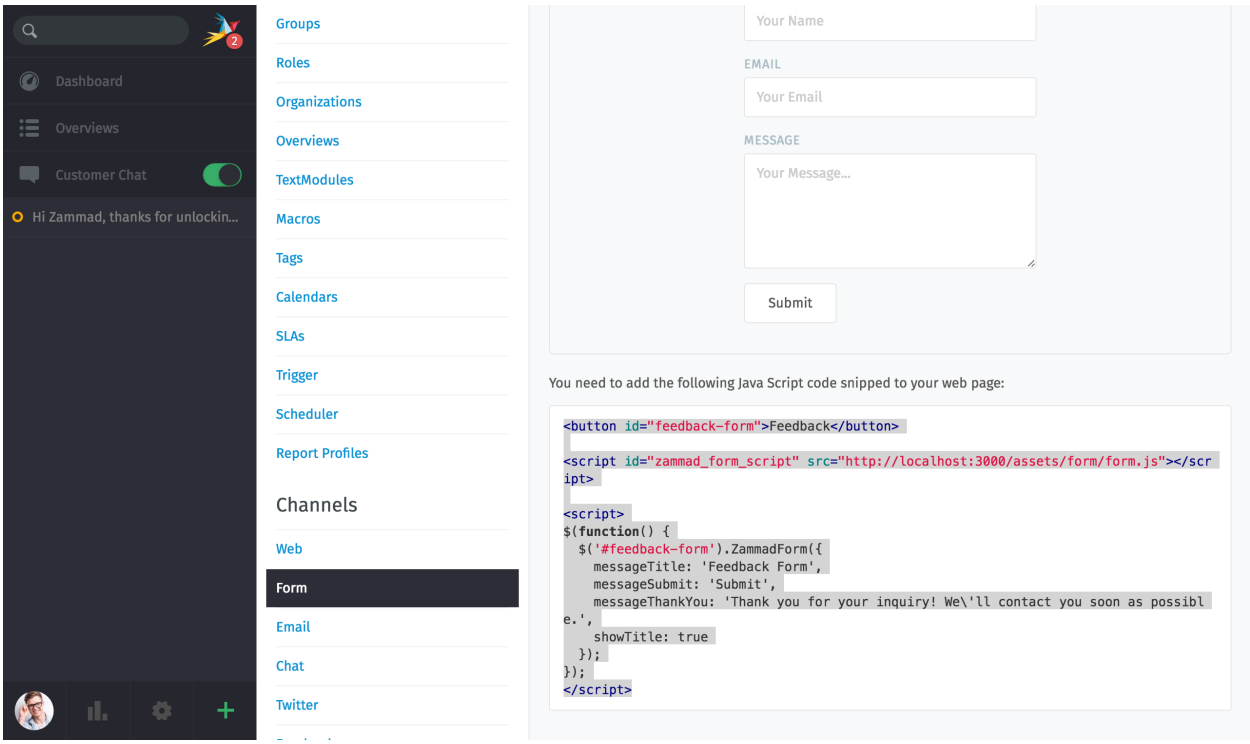
Go to „Admin -> Channels -> Form“ and enable the switch for this feature. Below you can adapt your form settings.



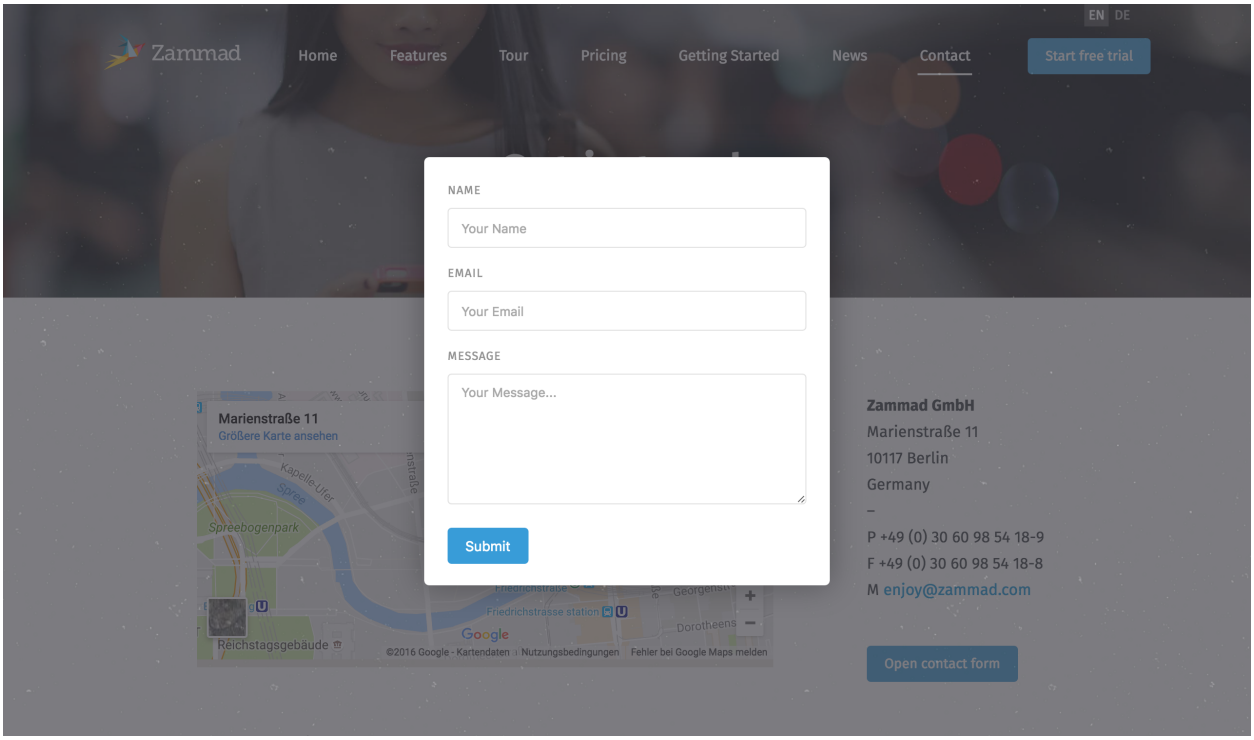
In preview mode you can test if the form fits your needs (e. g. if it opens as modal dialog or if it's shown inline on the website)



Just copy the JavaScript snippet and paste it into your website. That's all. Just 2 minutes.



Finally, the forms will look as follows:



17.1 Potential Spam-Issue

The Form function could be abused by sending a higher 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 created tickets based on different criteria. It also ensure that the email address being used is valid (with MX-Check on the email's domain).

Tab. 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

Hinweis: You can change the above 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 hosts that runs Zammad.

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')
```

17.1.1 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.

Further more it's not recommended to delete form.js from it's location, but to forbid the access to it with your webserver configuration (if wanted).

KAPITEL 18

Email

18.1 Accounts

18.1.1 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:

Gefahr: Zammad will delete all emails in your inbox during the import process.

Use the *Experts* dialog to *disable this behavior*.

Warnung: Zammad will send an auto-reply message to every email it imports. (Including the old ones!)

Use the *Experts* dialog to *change this behavior*.

Bemerkung:

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*.

Microsoft 365 / Exchange users: Shared mailboxes are not supported.

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.

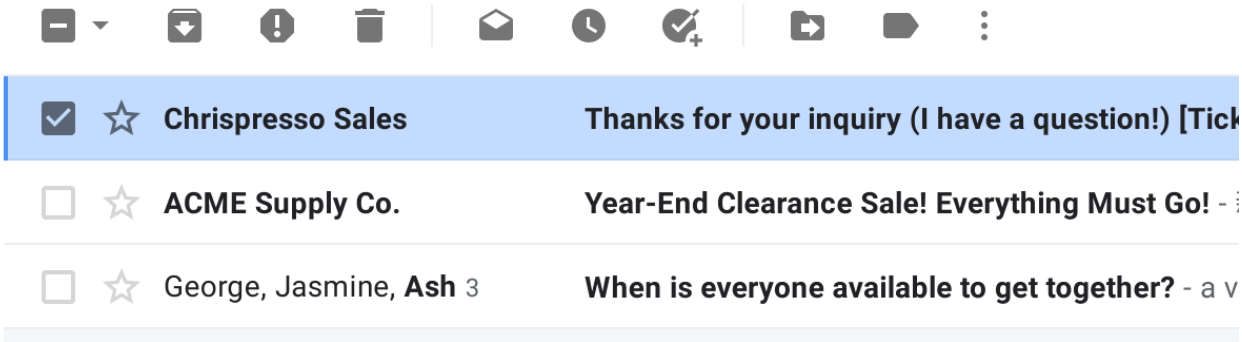


Abb. 1: A customer's inbox with an auto-reply from **Chrispresso Sales**.

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*.

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.

If your email provider requires you to enter a one-time passcode (sent via SMS) when logging in, you'll have to generate an app password to use with Zammad.

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.

Password Your account password.

If your email provider requires you to enter a one-time passcode (sent via SMS) when logging in, you'll have to generate an app password to use with Zammad.

SSL / STARTTLS Enable encryption when fetching messages.

Choose from **yes** and **no**; Zammad will detect which protocol to use.

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.)

Bemerkung: 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.)

Bemerkung: 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.

Bemerkung: This option may not be shown if:

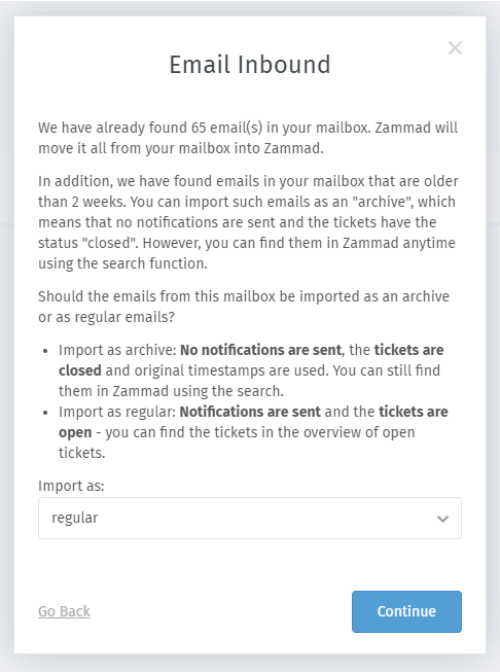


Abb. 2: How should old emails be imported?

- all messages in your inbox are less than two weeks old
- you selected **Keep messages on server: Yes**
- 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. Once this test email is received, the new account setup process is complete!

Troubleshooting

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

18.1.2 Secondary Addresses

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

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

> **Warnung: 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.

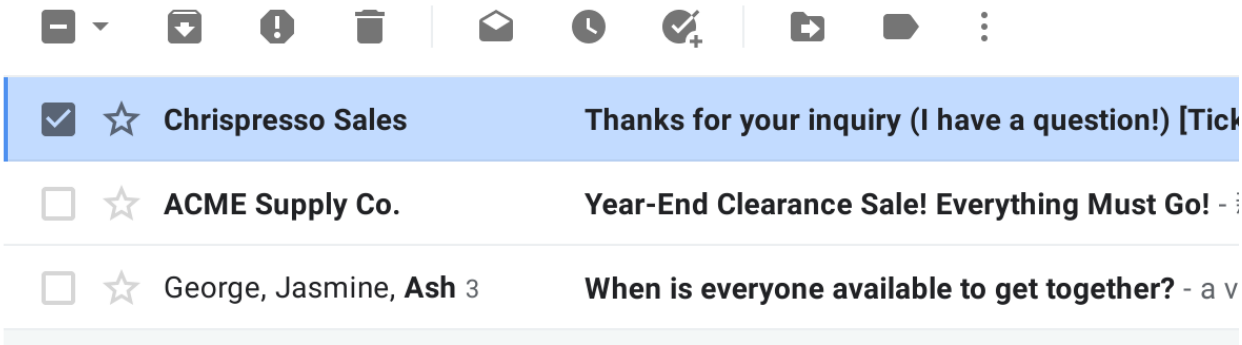


Abb. 4: A customer's inbox with an auto-reply from **Chrispresso Sales**.

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

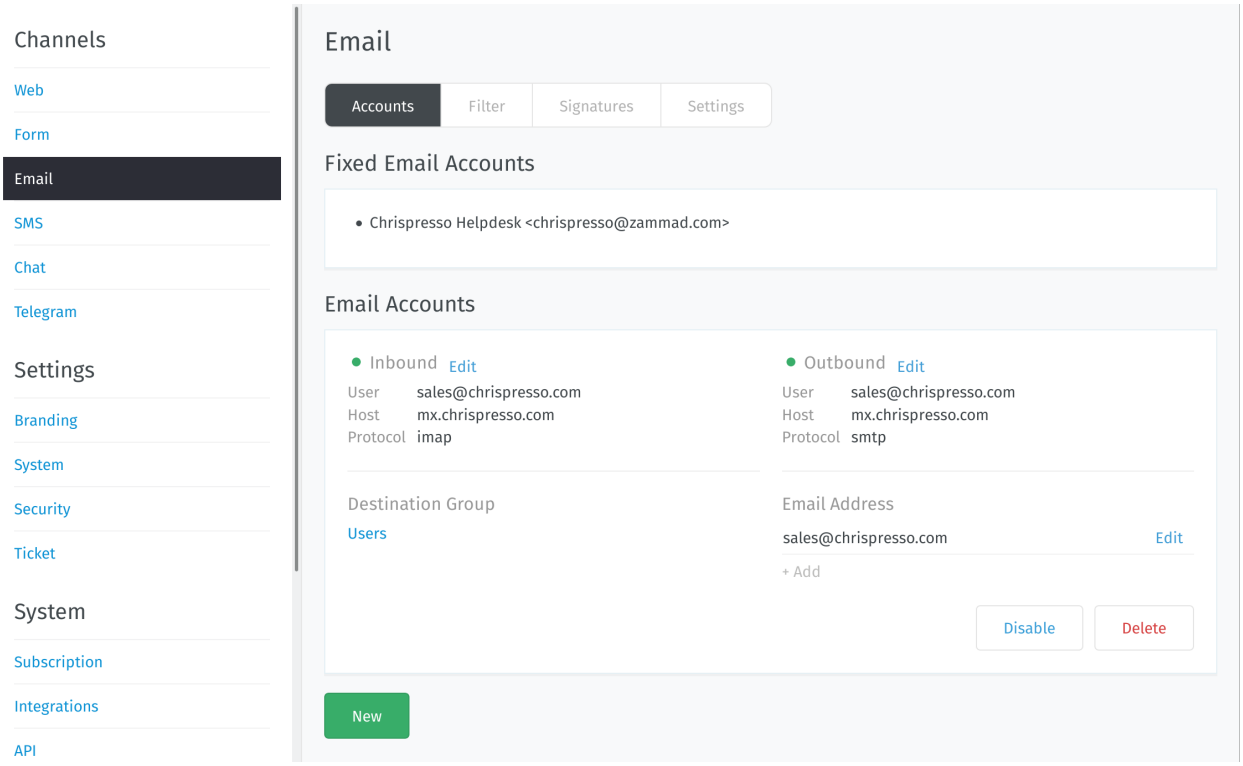
Email The alias address to send outgoing messages as.

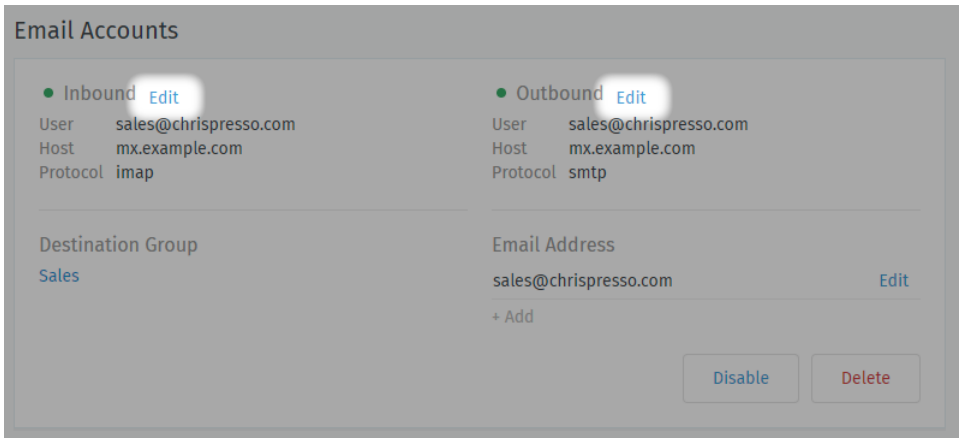
Channel The email account to be used when sending outgoing messages from this alias.

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

18.1.3 Managing Accounts

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


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

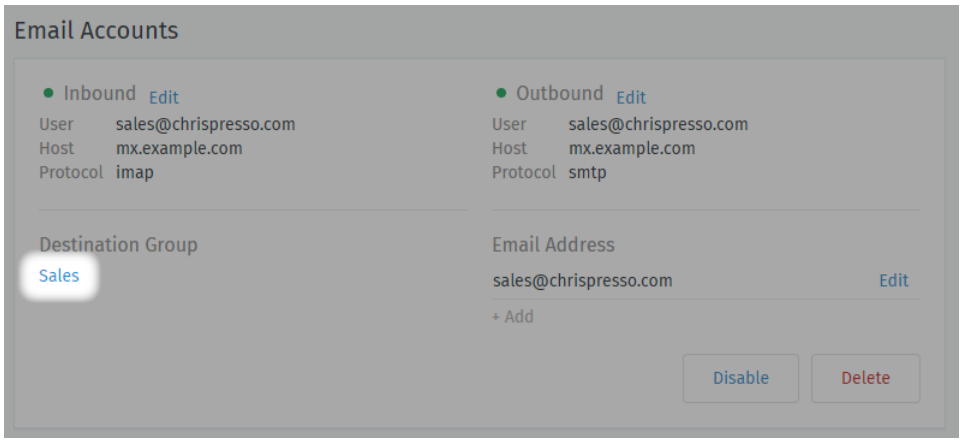

Email Accounts
Inbound Edit
User sales@chrispresso.com
Host mx.example.com
Protocol imap
Outbound Edit
User sales@chrispresso.com
Host mx.example.com
Protocol smtp
Destination Group
Sales
Email Address
sales@chrispresso.com Edit
+ Add
Disable
Delete

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.

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

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



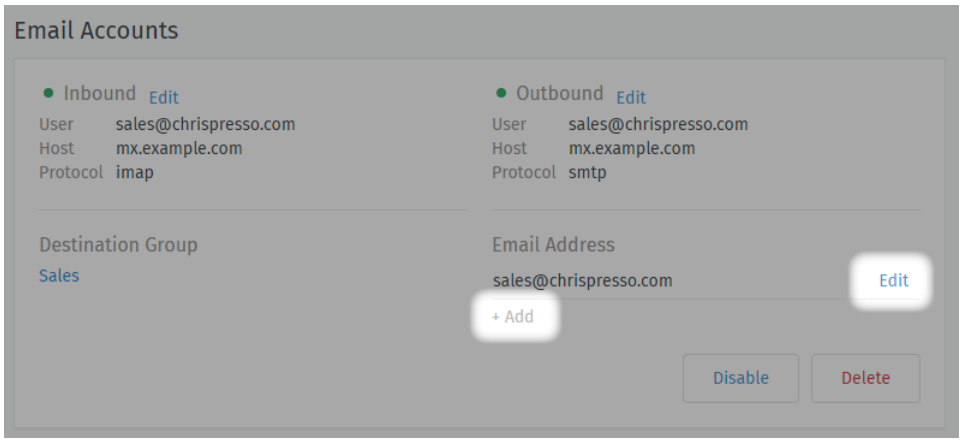
Only **active** groups will be displayed.

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

Hinweis: **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.

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

Delete Deleting an account removes its configuration from Zammad entirely.

Bemerkung: Additional Steps Required

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

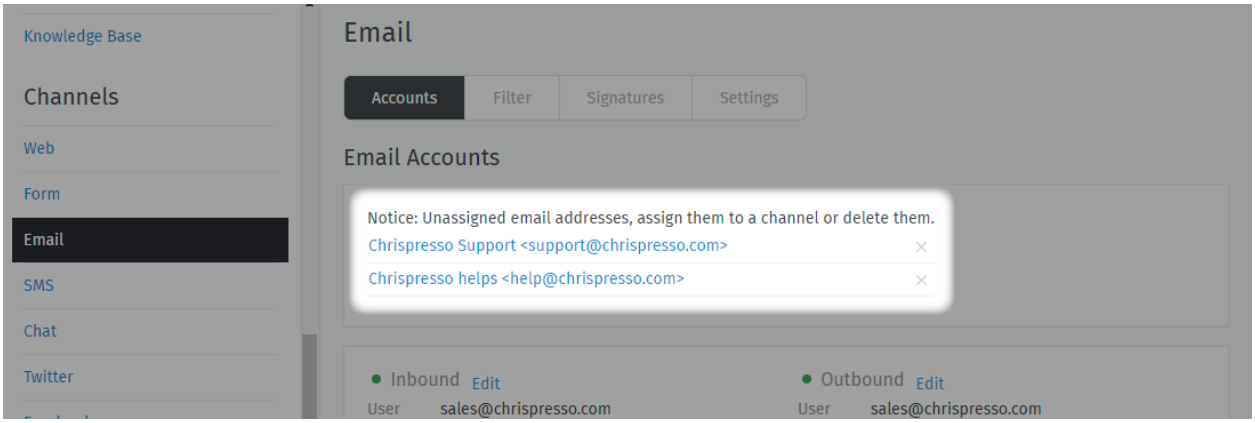


Abb. 5: Click on the address to assign it to another account, or click to delete it.

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.

18.1.4 Email Notification

Bemerkung: 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.

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

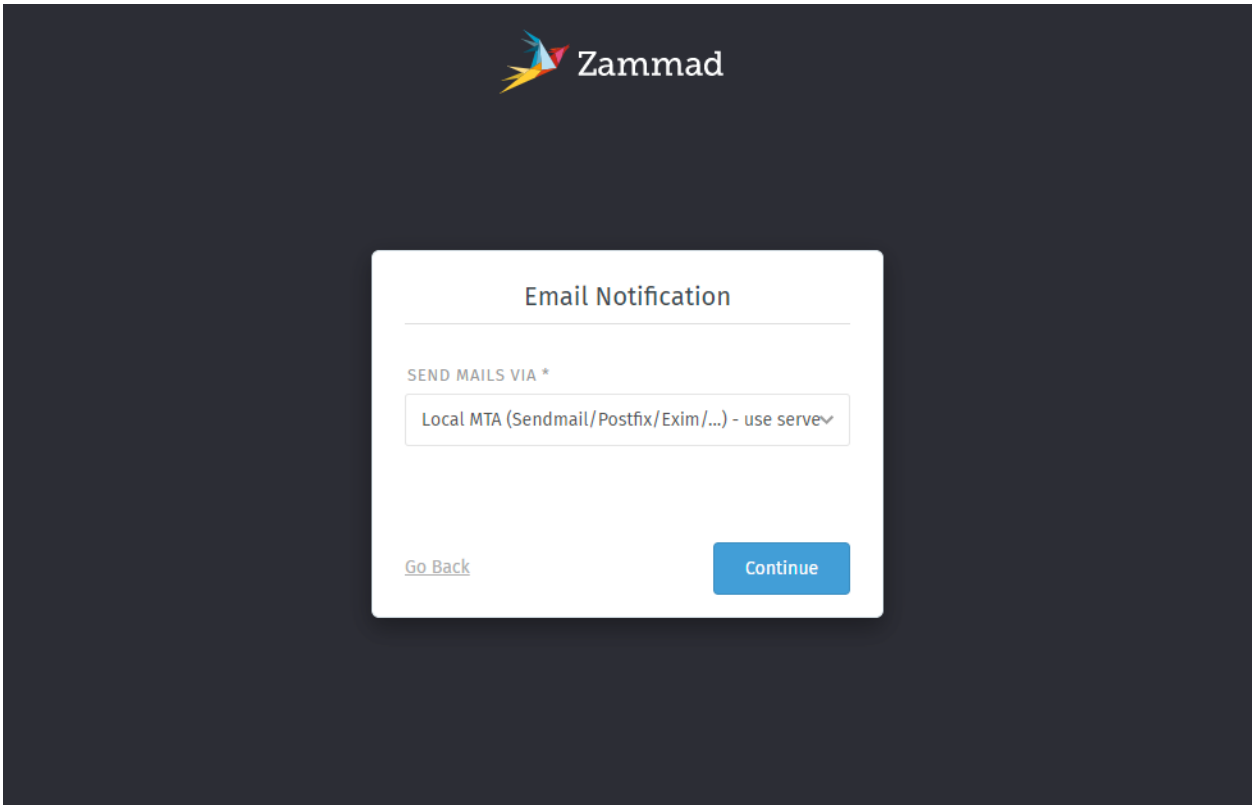
Password Your account password.

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

Zammad will detect and enable SSL/STARTTLS support automatically.

Bemerkung: 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.)

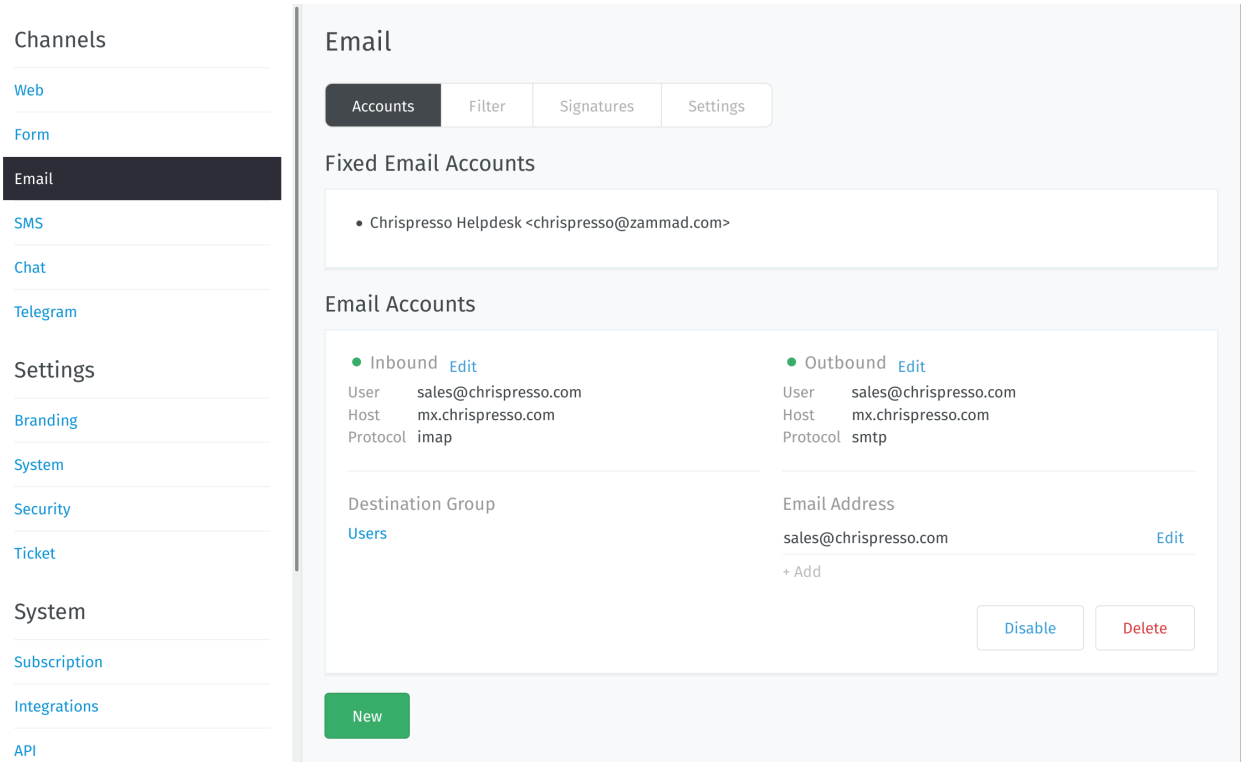
18.2 Filters

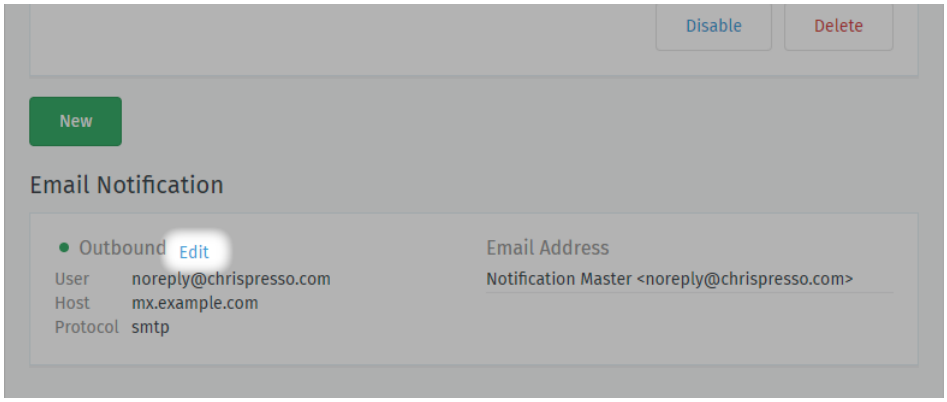
Postmaster filters allow you to match email headers (e.g. `From`, `To`, `Subject`, `X-Spam-Flag` etc.) and execute a set of actions whenever Zammad's email parser encounters a matching email. The actions will be applied to the ticket that is created or updated by this email. Here are some examples of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

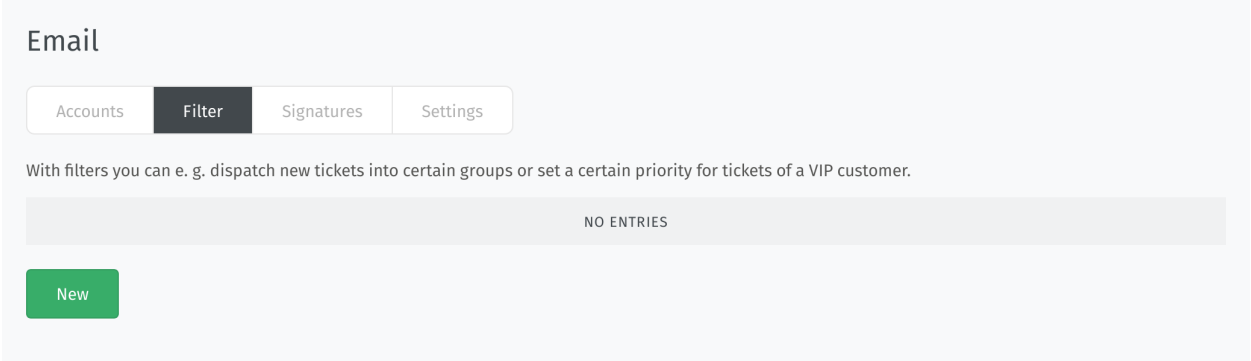
- Automatically dispatch tickets into certain groups:

  For example, tickets from `amazon.com` could automatically be dispatched to the Purchasing group.

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


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

Group: Purchasing

Bemerkung: 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

  Bemerkung: 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 an external spam filter (e.g. SpamAssassin):

  X-Spam-Flag: contains: `YES`
  Tag: add: `spam`
  State: closed

  Bemerkung: 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.

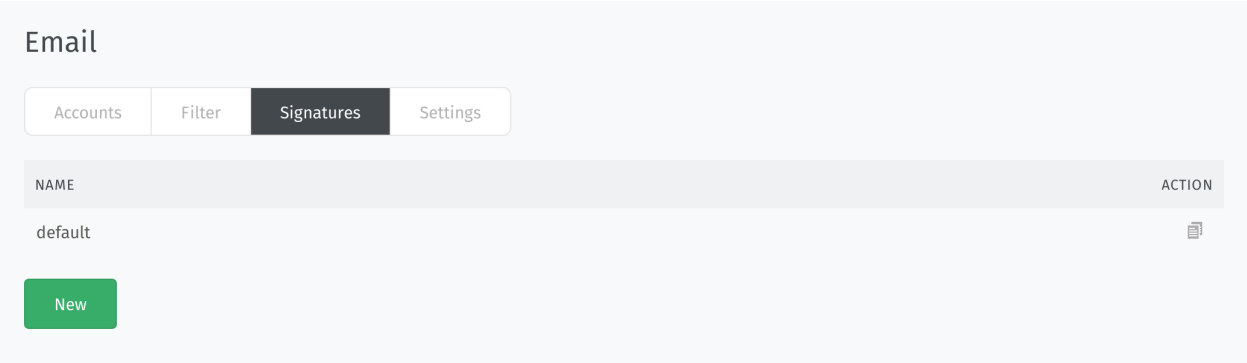
Different attributes of a filter can be combined with each other. Likewise, the following actions can be combined. The supported matches are „contains“ and „contains not“; for advanced matching, you can use regular expressions by prefixing the string with `regex:`.

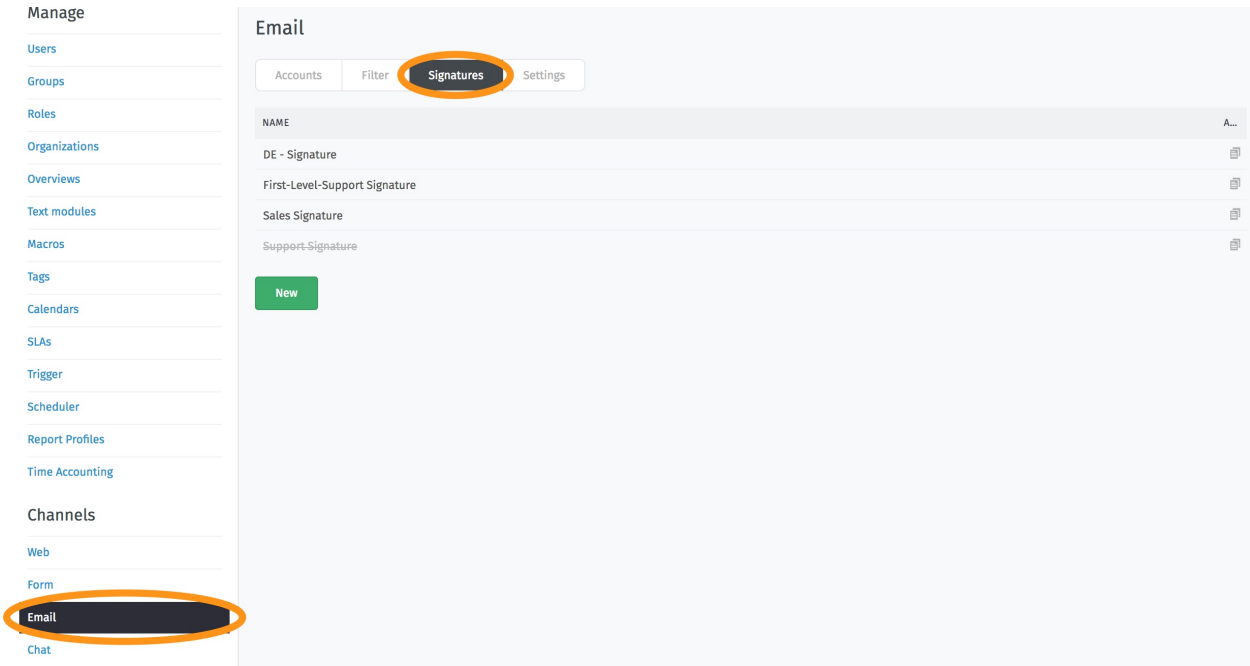
Note that Zammad matches against the full header, e.g. for a mail with „`From: Display Name <display.name@example.com>`“, the `From` condition will test against „`Display Name <display.name@example.com>`“. This is especially important when using anchored regular expressions; `regex:^display\.name@example.com$` 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.

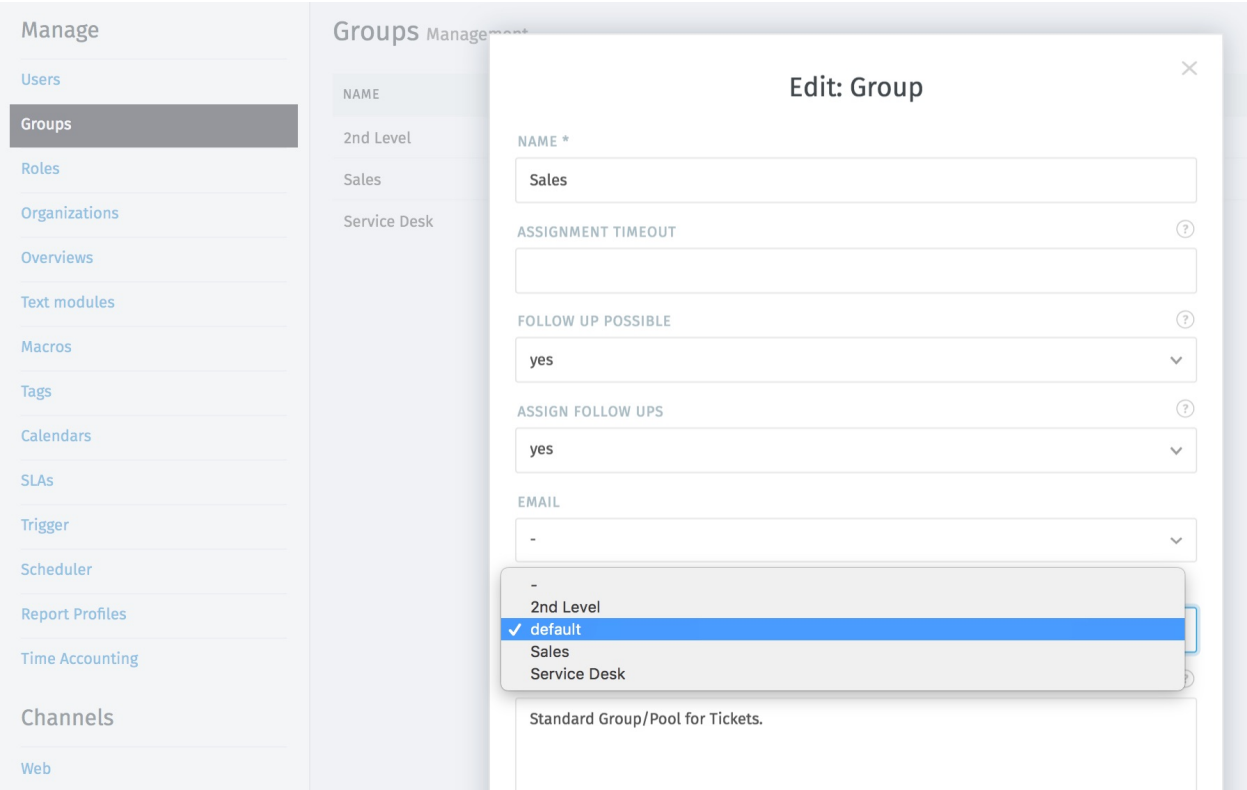
18.3 Signatures

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:

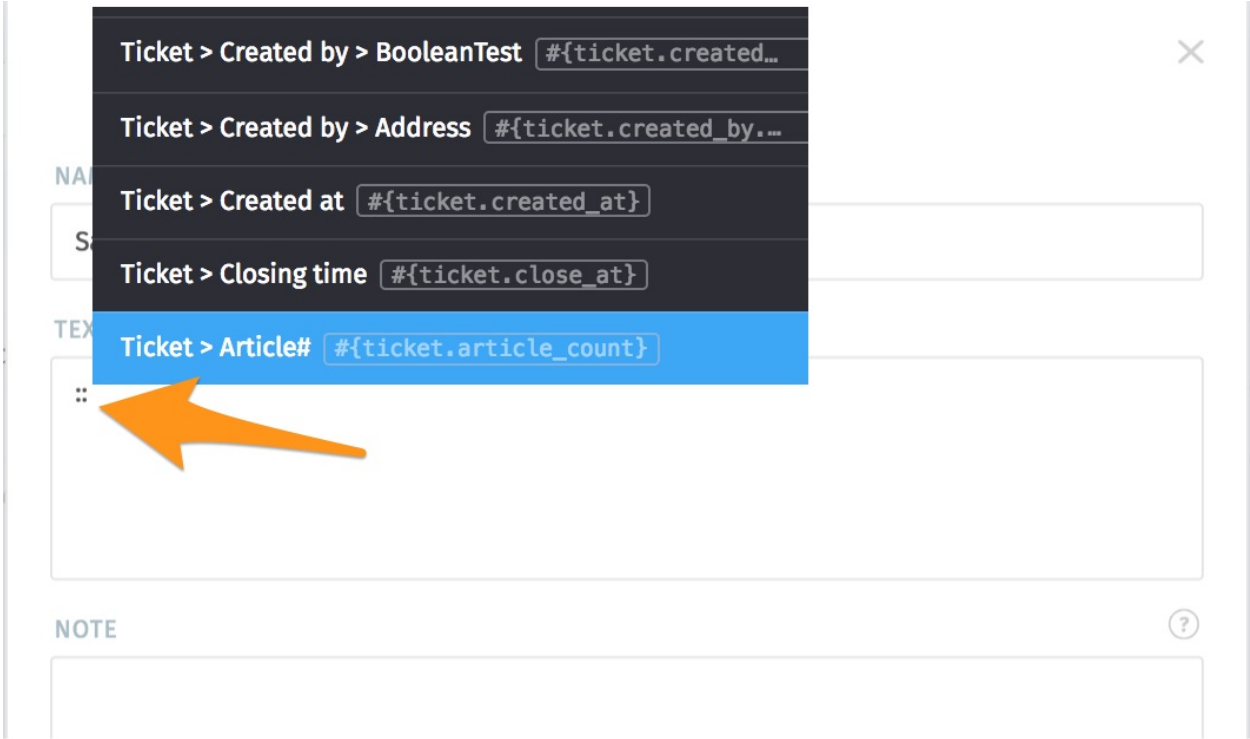


Each group can be assigned its own signature, but they can also all use the same signature.

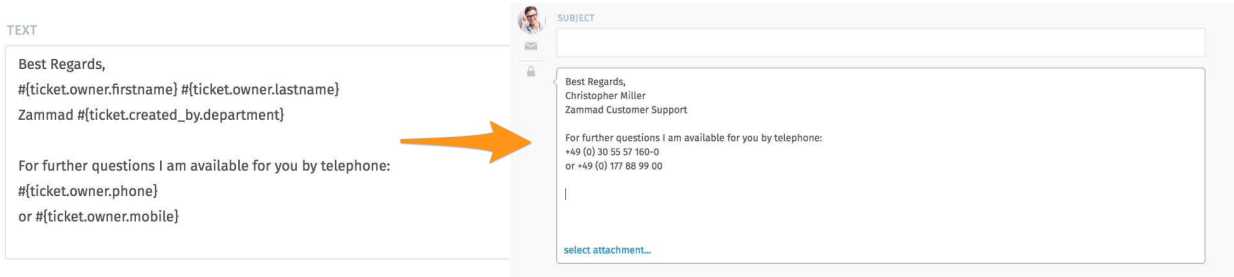
18.3.1 Dynamic Signatures

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.

Hinweis: 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:



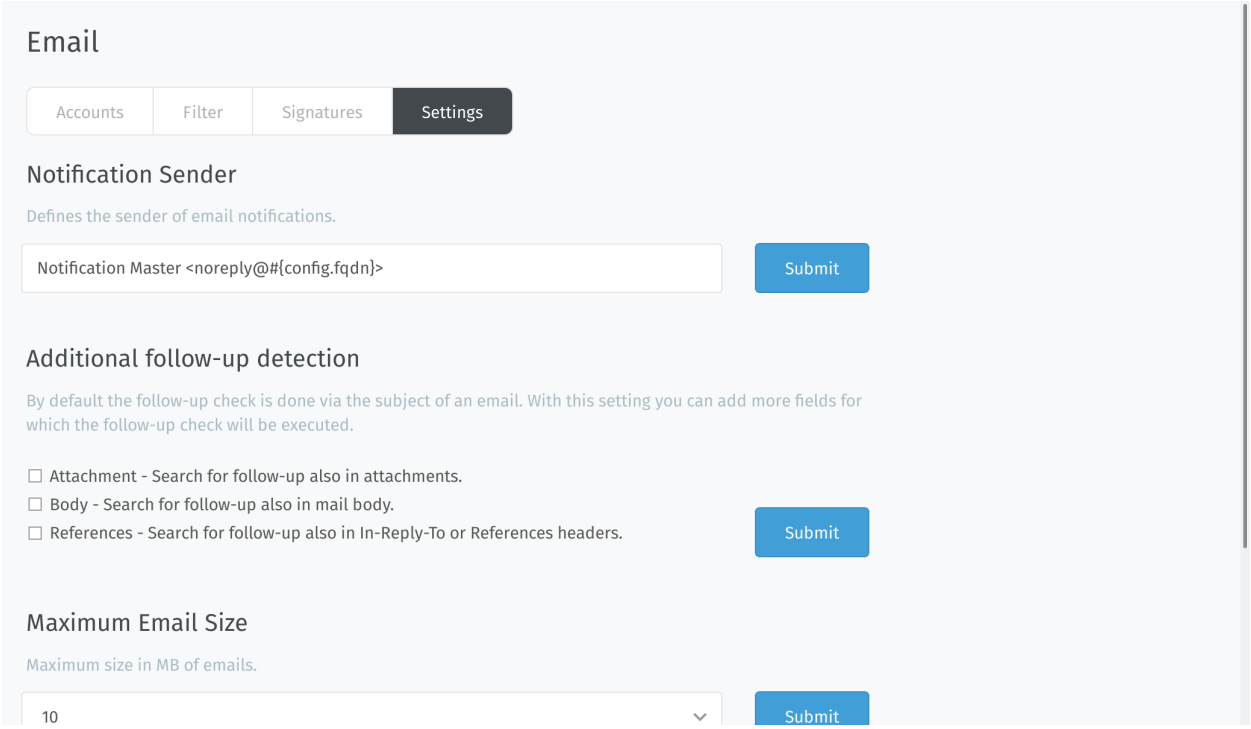
18.4 Settings

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.

Bemerkung: Some email-related settings are ticket based settings, which is why they can be found in the *Composer Settings*.

18.4.1 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.

Bemerkung: 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.

Bemerkung: Please note that searching in attachment and body might lead to false follow-up detections.

Maximum Email Size: Default value `10 MB` This one is pretty obvious: It defines the maximum allowed size of a Email Zammad will fetch. Zammad will not fetch Mails that are bigger than this option.

Bemerkung: Starting with Zammad 3.2 Zammad can provide postmaster mails (see „Send postmaster mail if mail too large“ below).

Hinweis: This technically also affects attachments for articles.

Send postmaster mail if mail too large: Default value `yes (enabled)` on fresh installations and `no (disabled)` on upgrad

Bemerkung: This option is only available with Zammad 3.2 and later. 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.

> **Bemerkung:** 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 it's 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 `-` or `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 `Take reply-to header as sender/from of email and use realname 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 receive list: Default value `yes` This option decides how Zammad should react if an agent sends a Email to it.

Option set to `yes` Whatever the first user / Email address is set within the recipient list is will be used a ticket customer.

Option set to `no` The agent will be set as ticket customer.

> **Bemerkung:** 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 any way.

The default value is: `(mailer-daemon|postmaster|abuse|root|noreply|noreply.+?|no-reply|no-reply.+?)@.+?`

Sender Format: Default value `Agent + Format Seperator + System Address Display Name` This configures the display name used in the `FROM` header of mails Zammad sends.

> **Bemerkung:** 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 + Format Seperator + System Address Display Name` This will cause Zammad to set the `FROM` header to agent name and the channels display name divided by a seperator (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.`

Sender Format Seperator: Default value `via` This is a can be a String you can freely choose. It divided the agents name and the display name of the channel when ever needed.

Ticket Subject Forward: Default value `FWD` The above string will be used on the subject if you forward a mail from Zammad.

> **Bemerkung:** `:` 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.

> **Bemerkung:** `:` 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]`

> **Bemerkung:** This does **not** limit ticket titles within the UI, just the subjects during replying to an Email.

18.4.2 Enhanced settings

Some less relevant settings can be changed via rails console if needed. As 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.

Control how Zammad **sends and receives email**.

> **Hinweis:**
>
> **Using Gmail / G Suite?** Set up a *Google channel* instead.
>
> **Using Microsoft 365?** Set up a *Microsoft 365 channel* instead.

Accounts **Connect Zammad to your email provider** so that it can watch your inbox, send auto-replies, and more.

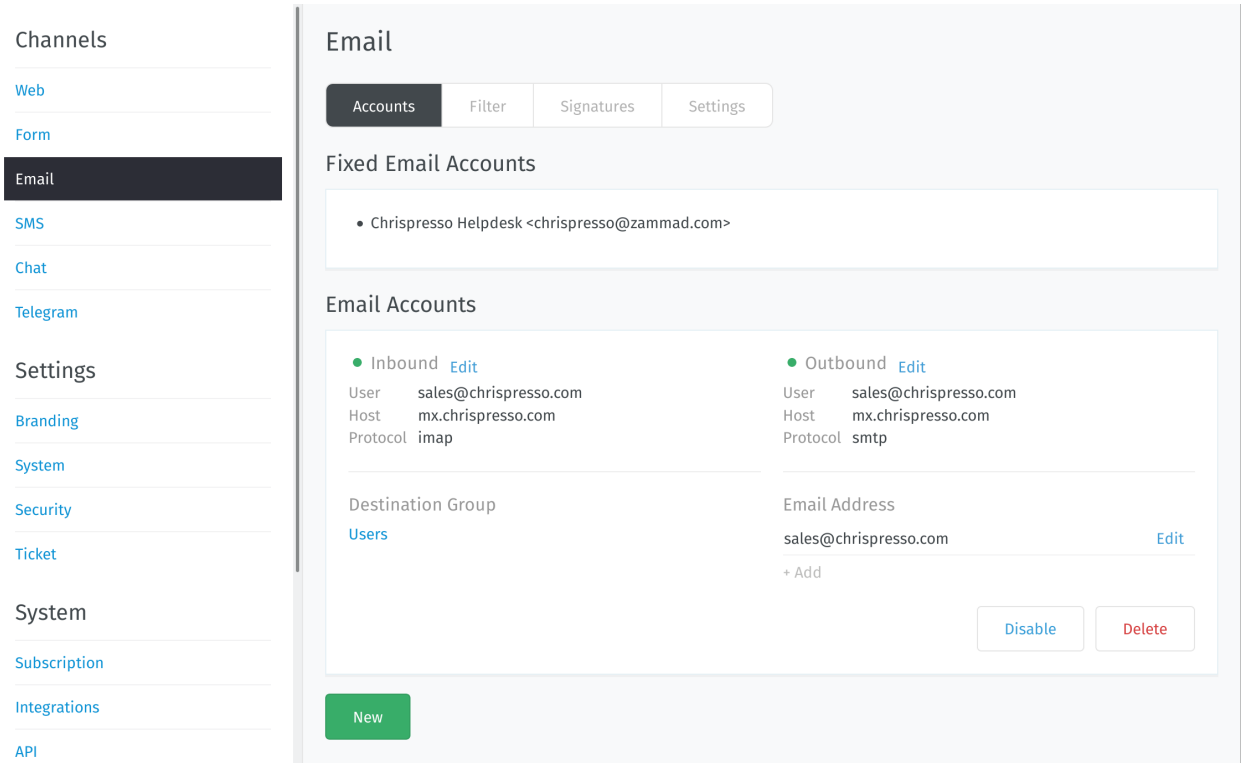
(Self-hosted users may have already completed this step during new system setup.)

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:")



Hinweis: Want to **manually edit email subjects** or **always copy parent messages into your replies**?

Check the *Composer Settings*.

18.5 Extra Options for Self-Hosted Users

If you're too cool for POP3/IMAP/SMTP...

18.5.1 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.

Bemerkung: 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)'
```

Supported email headers for presorting

Presorting based on email headers currently works for the following Ticket- and Article-Attributes.

Ticket attributes

For ticket creation use „X-Zammad-Ticket-Attribute: some value“. If you want to change ticket attributes on follow-up, use „X-Zammad-Ticket-FollowUp-Attribute: some value“.

X-Zammad-Ticket-Priority

- Sets priority of ticket (for available priorities check Zammad's database).
- Example: X-Zammad-Ticket-Priority: (1 low|2 normal|3 high)

X-Zammad-Ticket-Group

- Presorts by group (highest sort priority).
- Example: X-Zammad-Ticket-Group: [one system group]

X-Zammad-Ticket-Owner

- Assigns ticket to agent.
- Example: X-Zammad-Ticket-Owner: [login of agent]

X-Zammad-Ticket-State

- Sets state of ticket (for available states check Zammad's database)! Be careful!
- Example: X-Zammad-Ticket-State: (new|open|...)

X-Zammad-Customer-Email

- Sets customer via explicit email address.
- Example: X-Zammad-Customer-Email: [email address]

X-Zammad-Customer-Login

- Sets customer via explicit login.
- Example: X-Zammad-Customer-Login: [login]

Article attributes

Everytime an article is being created (new ticket or/and follow up) you can use „X-Zammad-Article-Attribute: some value“.

X-Zammad-Article-Sender

- Info about the sender.
- Example: X-Zammad-Article-Sender: (Agent|System|Customer)

X-Zammad-Article-Type

- Article type (for available types check Zammad's database).
- Example: X-Zammad-Article-Type: (email|phone|fax|sms|webrequest|note|twitter status|direct-message|facebook|...)

X-Zammad-Article-Visibility

- Article visibility.
- Example: X-Zammad-Article-Visibility: (internal|external)

Ignore Header

- If you want to ignore an email, just set the „X-Zammad-Ignore“ header.
- Example: X-Zammad-Ignore: [yes|true]

18.5.2 Dispatch messages with Sendmail

> **Warnung:** 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:

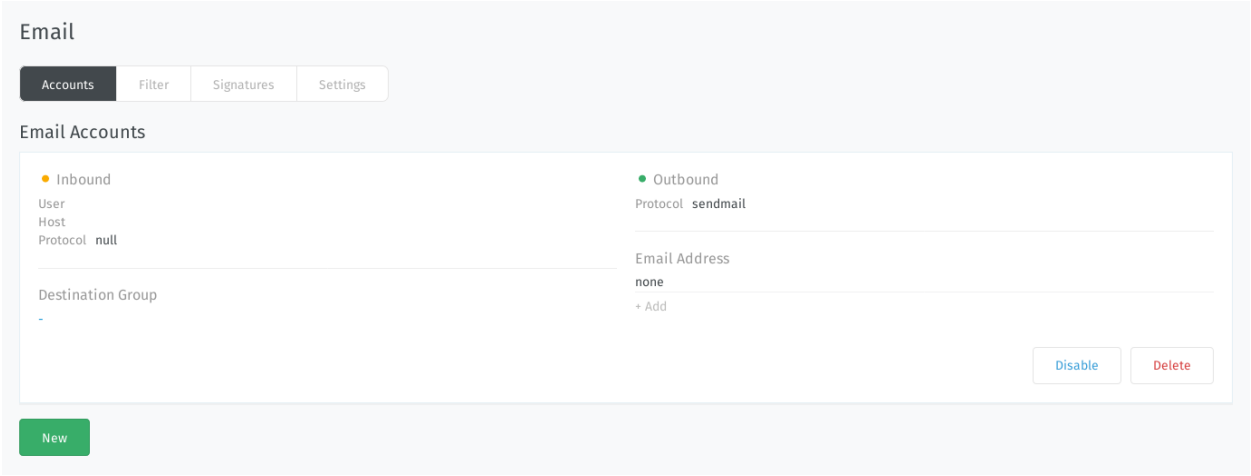


Abb. 6: Use the **Add** button under the **Email Address** heading to add new email addresses to send from.

KAPITEL 19

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.

19.1 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

19.2 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:

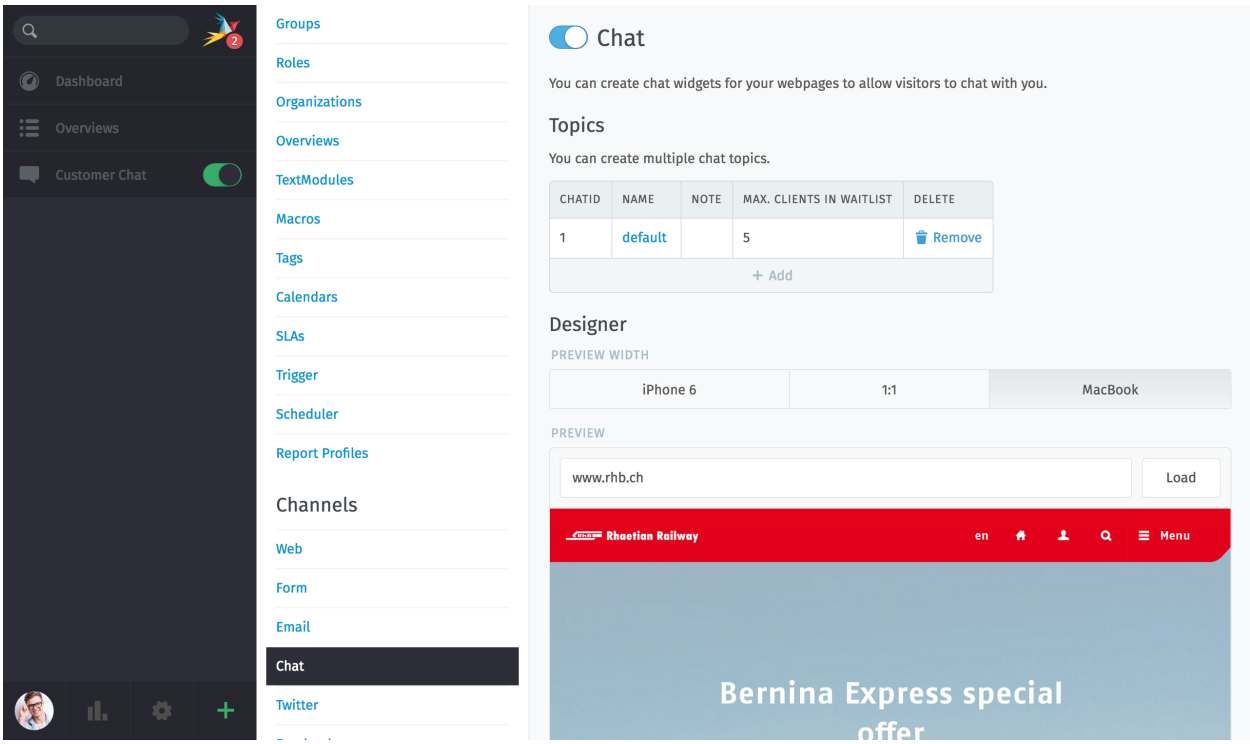
- We'll only display the chat widget, if at least one agent is available and the agent still has capacity. - If no agent is online or the agents are absend, the chat won't be available

- 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.

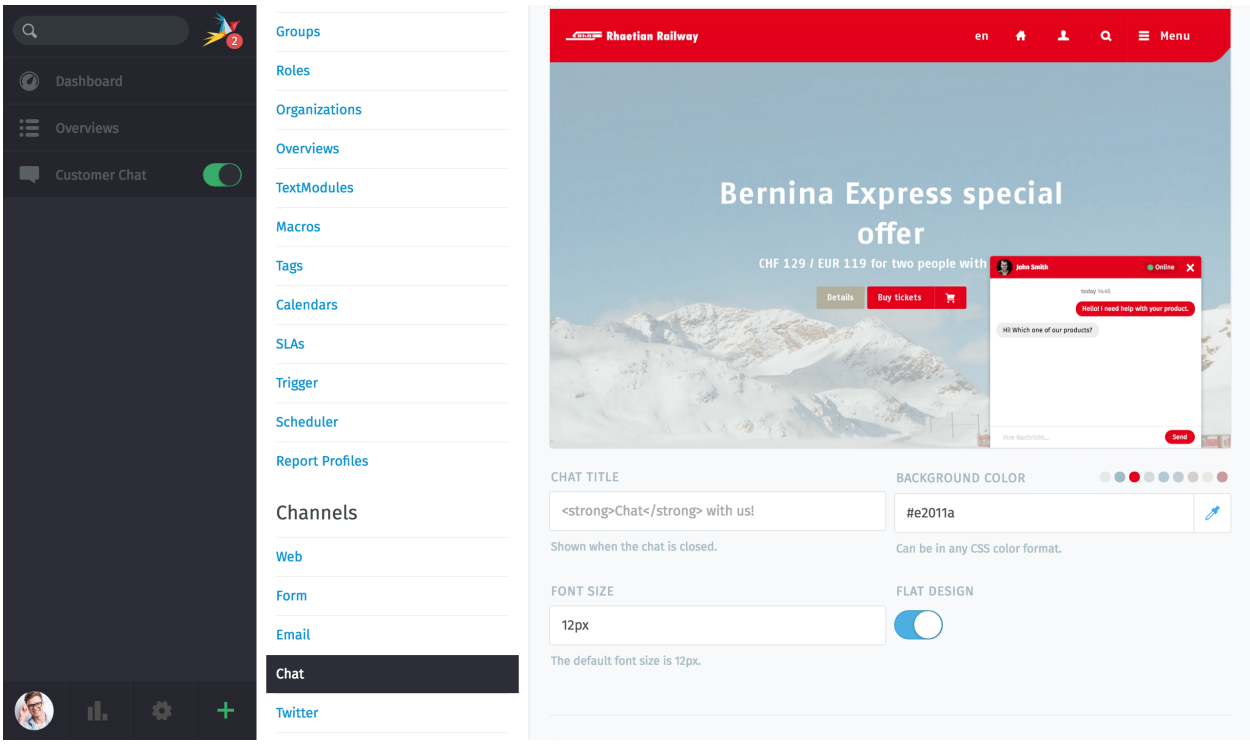
19.3 Configuration of the Chat widget

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:

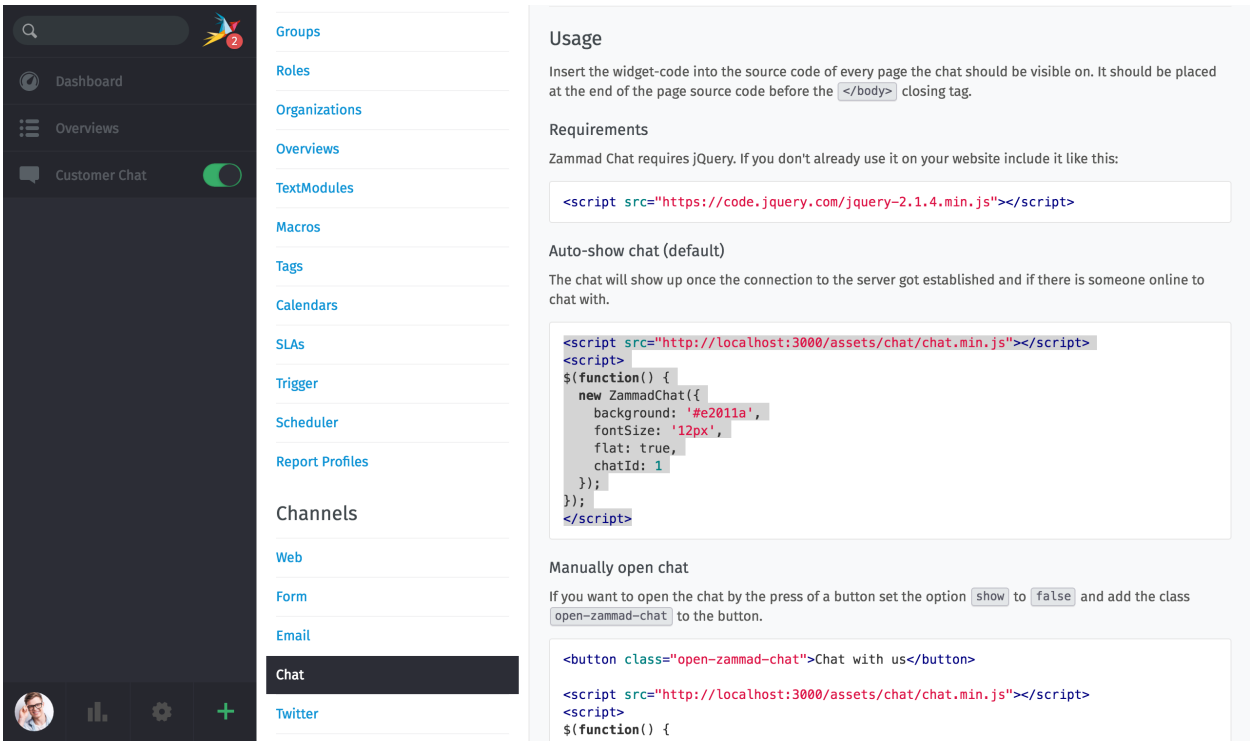


You can set up chats for different websites and edit them independently. The integrated designer helps the chat-widget to adapt to the website color. If you don't like the proposed design, you can manually adjust the design. Through the different previews you have the possibility to display directly 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:

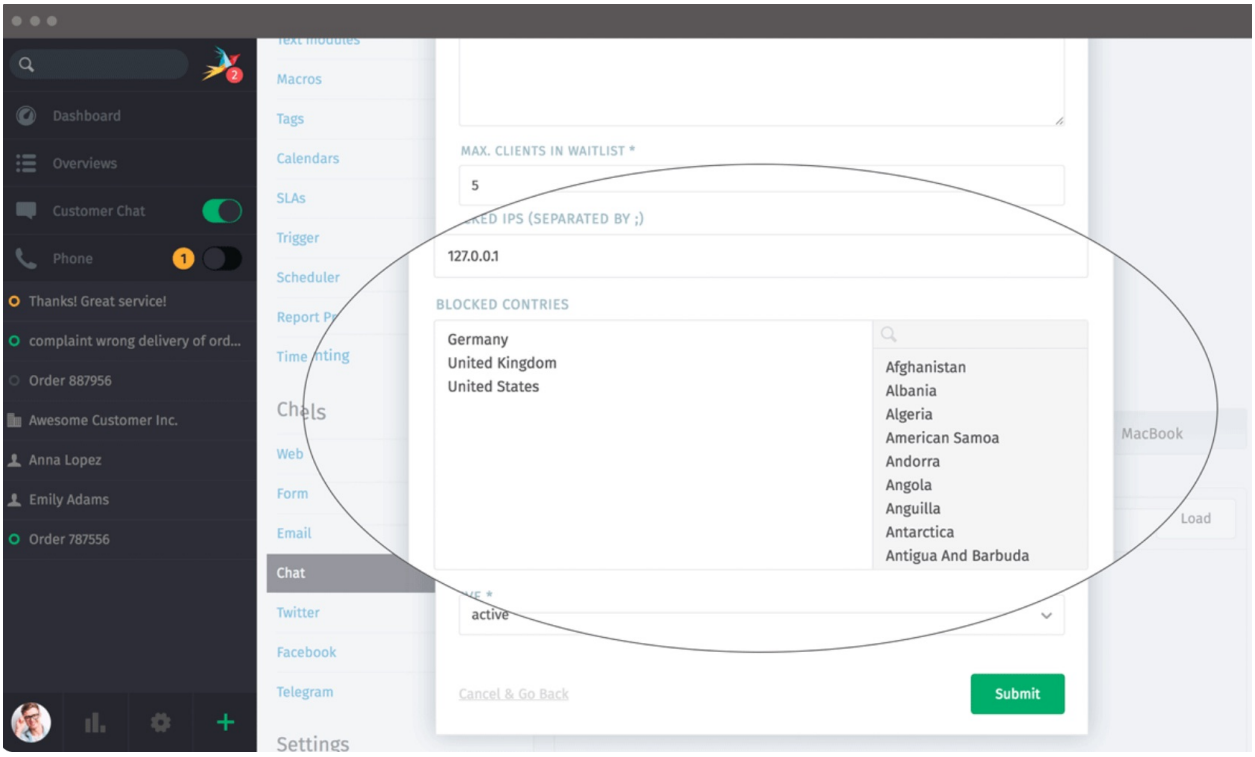
<script src="https://code.jquery.com/jquery-2.1.4.min.js"></script>

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 adresses 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:



In the admin area you can also find more information about the customization of the chat.

KAPITEL 20

Google

20.1 Accounts

20.1.1 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).

Bemerkung: 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.

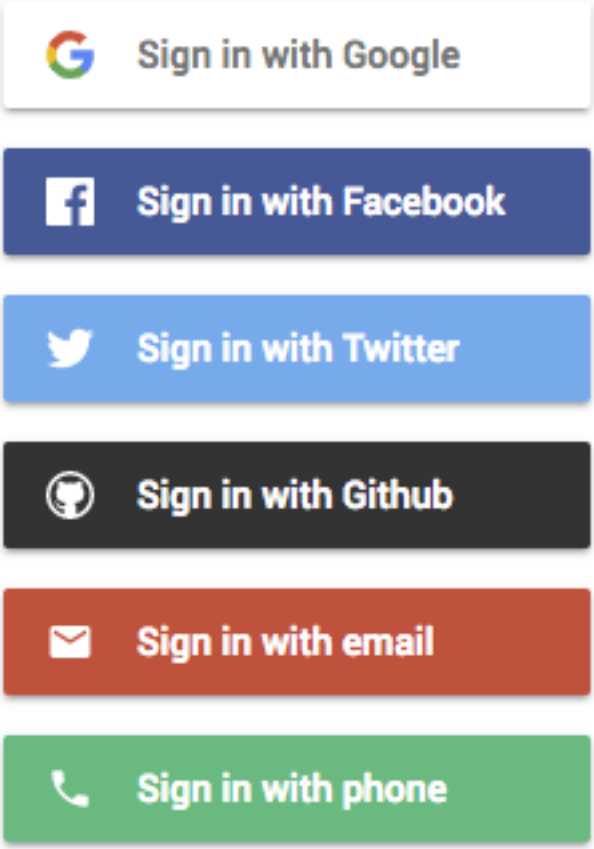
In this case, **Zammad is that third-party website**. That's why adding a Gmail account is a two-stage process: first, you have to register Zammad as a website that wishes to access Google user data; then, you have to add yourself as a Google user who agrees to let Zammad fetch your email.

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.


Sign in with Google
Sign in with Facebook
Sign in with Twitter
Sign in with Github
Sign in with email
Sign in with phone

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.)

20.1.2 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:

> **Gefahr: Zammad will delete all emails in your inbox during the import process.**
>
> Use the *Keep Messages on Server* setting to *disable this behavior*.

Warnung: 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 Gmail 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). **But there's no rush just yet**.

Upcoming versions of Zammad will feature an automated migration wizard to help you make the switch, and it'll be available well before Google officially pulls the plug on password auth.

Hinweis: If you'd prefer not to wait, you can do it manually today—just remember to **delete the email channel for your Gmail account** (and all its associated aliases) before re-adding it here.

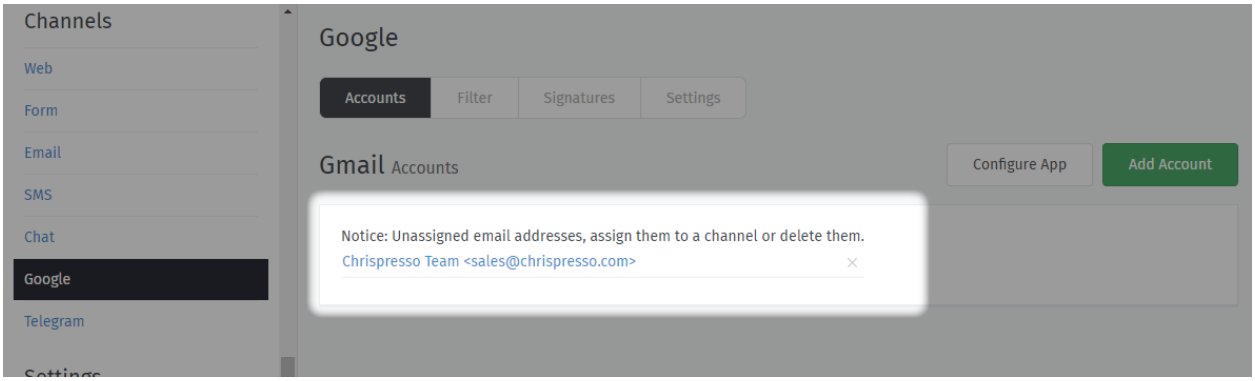


Abb. 1: Make sure you don't have any of these left over before creating your new Google channel.

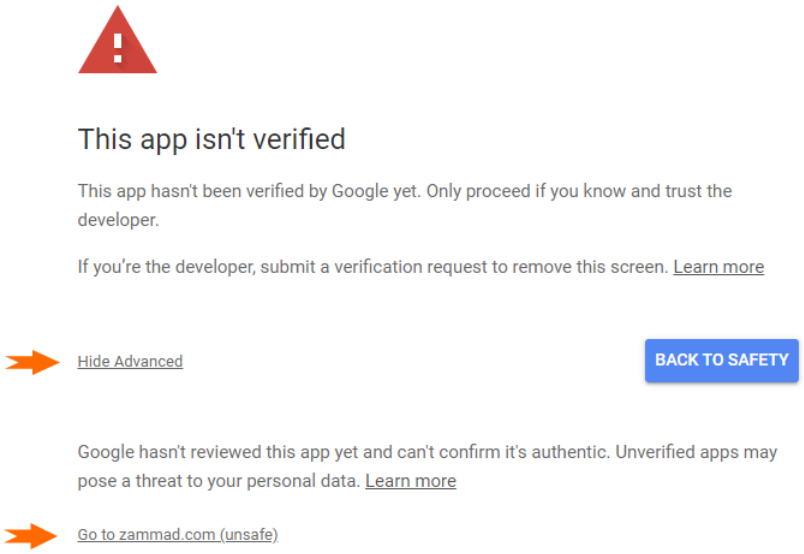
Add a New Account

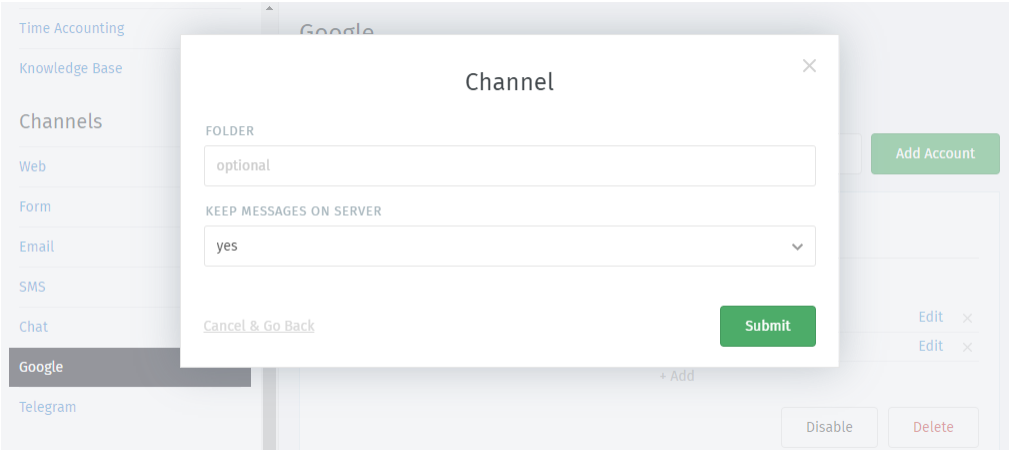
Click **Add Account** to connect your Gmail / G Suite accounts to Zammad. You will be redirected to a Google sign-in and confirmation page.

Any aliases registered in your Gmail settings will be imported automatically.

Bemerkung:

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.


This app isn't verified
This app hasn't been verified by Google yet. Only proceed if you know and trust the developer.
If you're the developer, submit a verification request to remove this screen. Learn more
Hide Advanced
BACK TO SAFETY
Google hasn't reviewed this app yet and can't confirm it's authentic. Unverified apps may pose a threat to your personal data. Learn more
Go to zammad.com (unsafe)


Time Accounting
Knowledge Base
Channels
Web
Form
Email
SMS
Chat
Google
Telegram
Channel
FOLDER
optional
KEEP MESSAGES ON SERVER
yes
Cancel & Go Back
Submit
Add Account
Edit
Edit
+ Add
Disable
Delete

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.)

Bemerkung: 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.

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?

Bemerkung: Secondary addresses in Google channels work (almost) **just like they do in email channels**, so this article is lifted (almost) verbatim from *here*.

20.1.3 Secondary Addresses

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

Abb. 2: Once you add a secondary address, you can configure a group to start sending emails with it.

Warnung: 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.

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

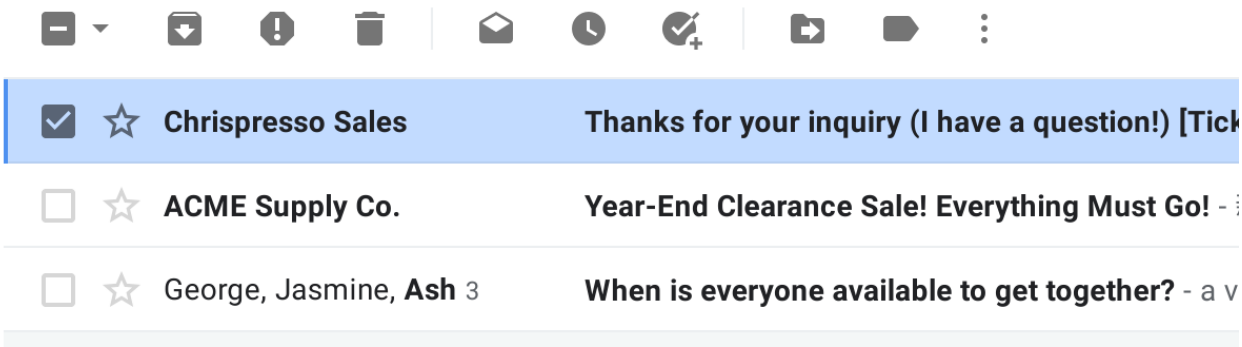


Abb. 3: A customer's inbox with an auto-reply from **Chrispresso Sales**.

Email The alias address to send outgoing messages as.

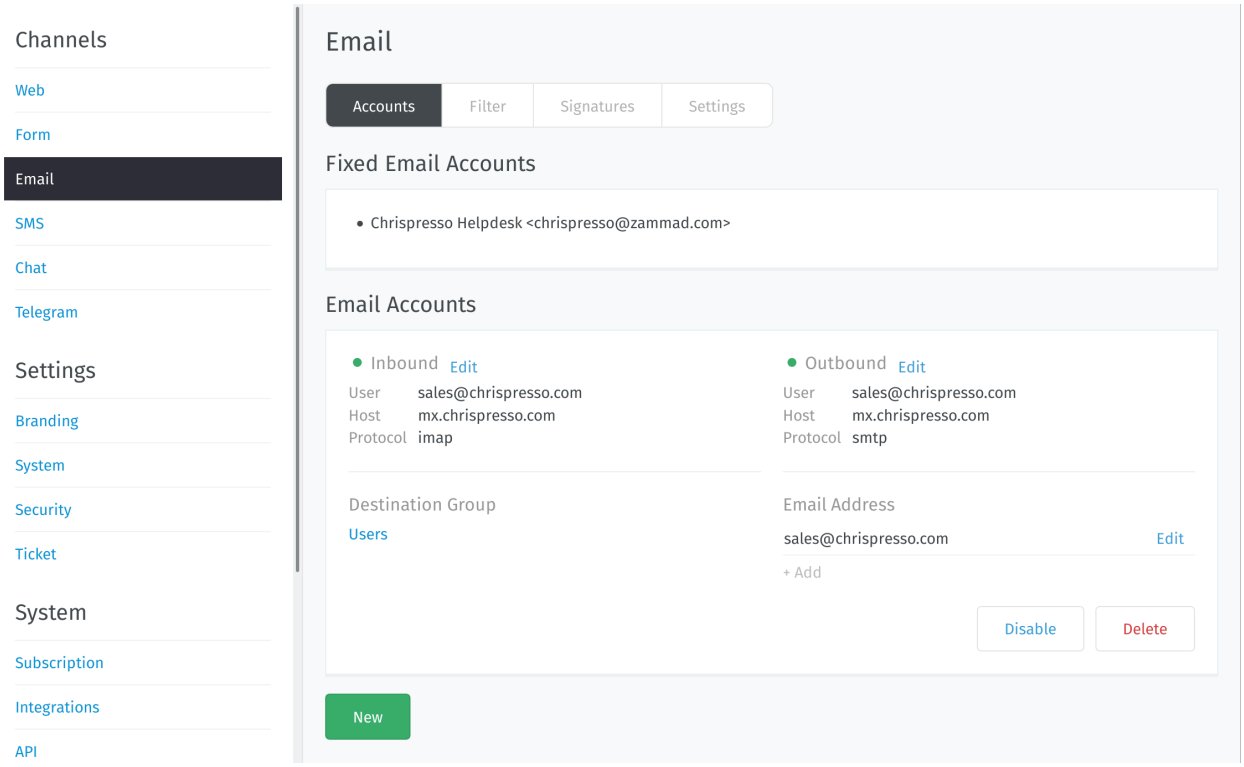
Channel The email account to be used when sending outgoing messages from this alias.

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

Bemerkung: Managing accounts in Google channels is (almost) **just like it is in email channels**, so this article is lifted (almost) verbatim from *here*.

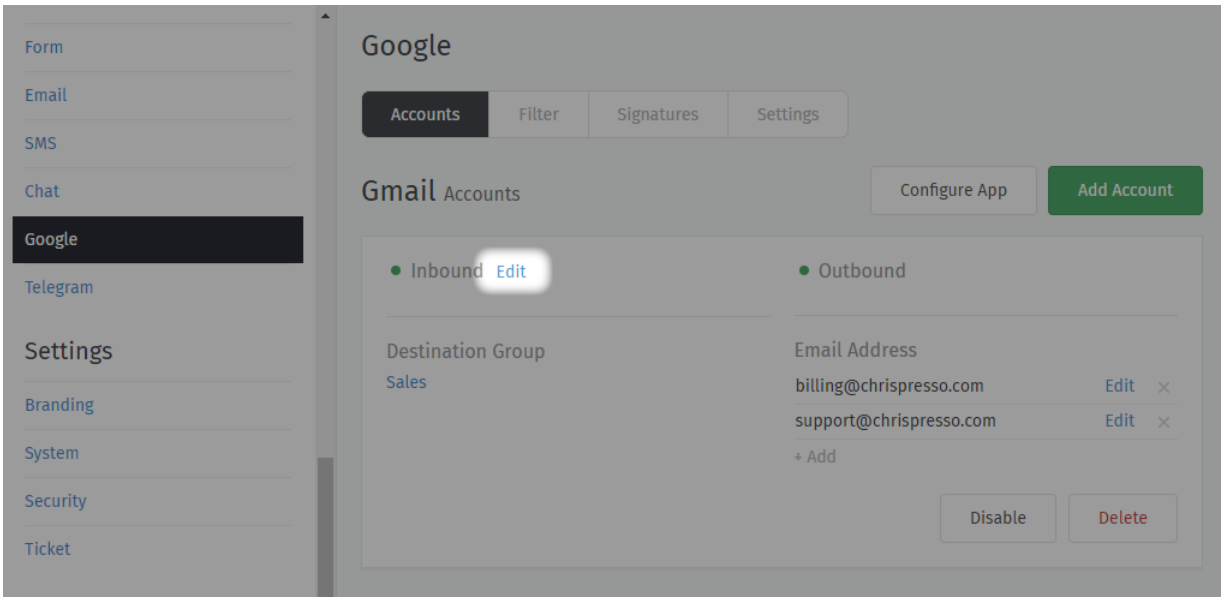
20.1.4 Managing Accounts

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

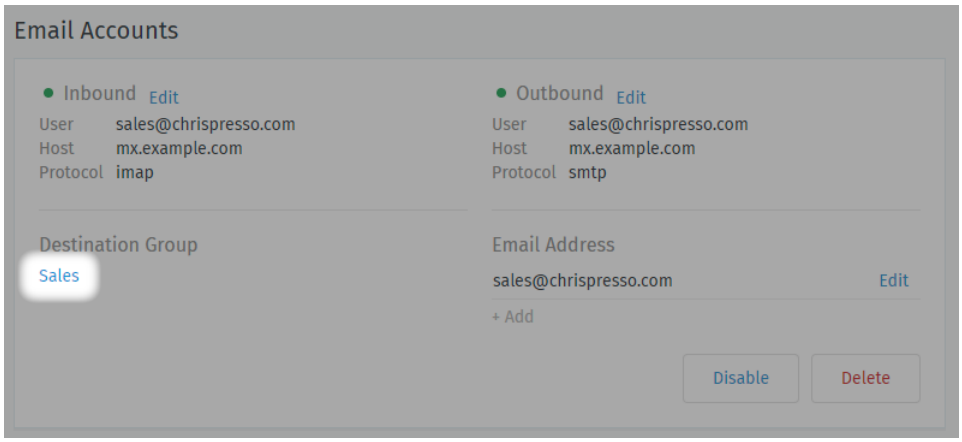


Fetch Preferences Click **Edit** on inbound account details to change how messages are retrieved from your account.

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.

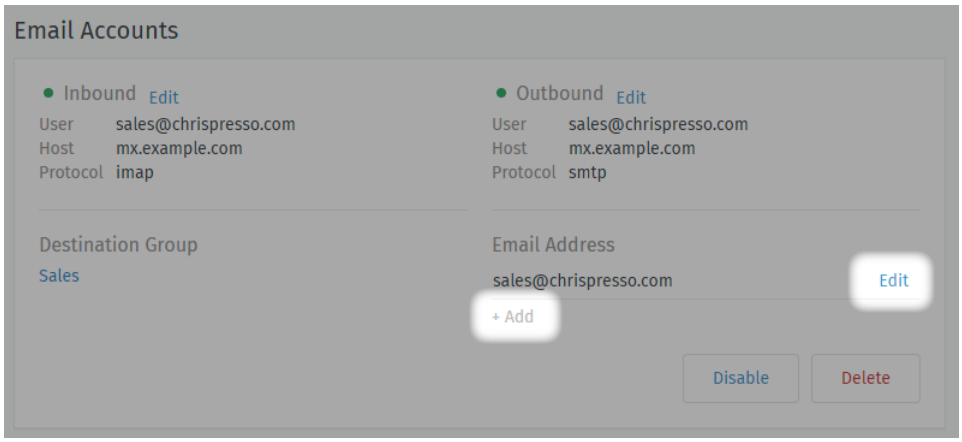
Hinweis: Still can't send outgoing email tickets? Check your *group settings*.

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.



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

Delete Deleting an account removes its configuration from Zammad entirely.

Bemerkung: 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.

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**.

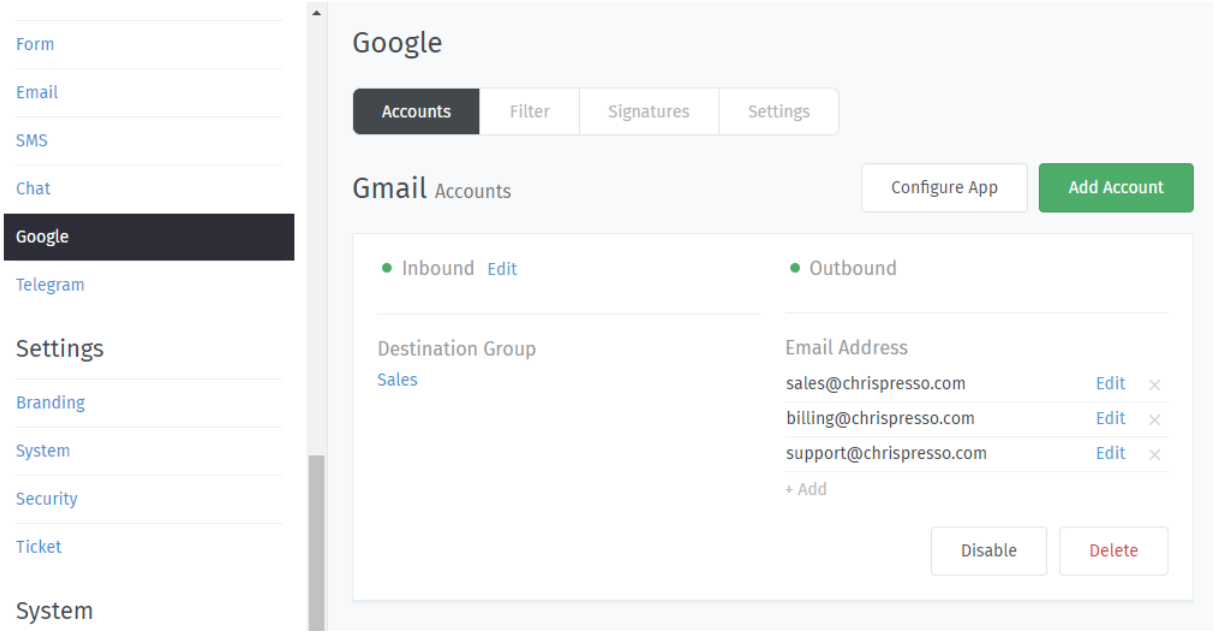
Bemerkung: How do I use my Gmail account for outgoing system notifications?

On **subscription/cloud-hosted instances**, you can't. Notifications will always come from "Notification Master <no-reply@your.zammad.domain>".

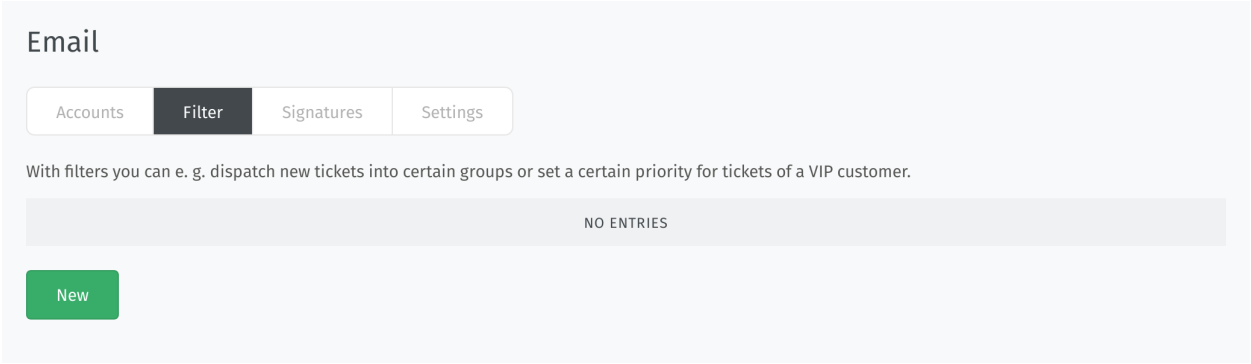
On **self-hosted instances**, we still don't recommend it. Using a Gmail account for automated, outgoing messages is risky: users who exceed Google's email sending limits can have their accounts suspended.

Set up a generic *email channel* instead, the use the *Email Notification* setting.

Bemerkung: Filters in Google channels are **just like filters in email channels**, so this article is lifted verbatim from *here*.



20.2 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

  Bemerkung: 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

  Bemerkung: 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 an external spam filter (e.g. SpamAssassin):

  X-Spam-Flag: contains: `YES`
  Tag: add: `spam`
  State: closed

  Bemerkung: 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:`.

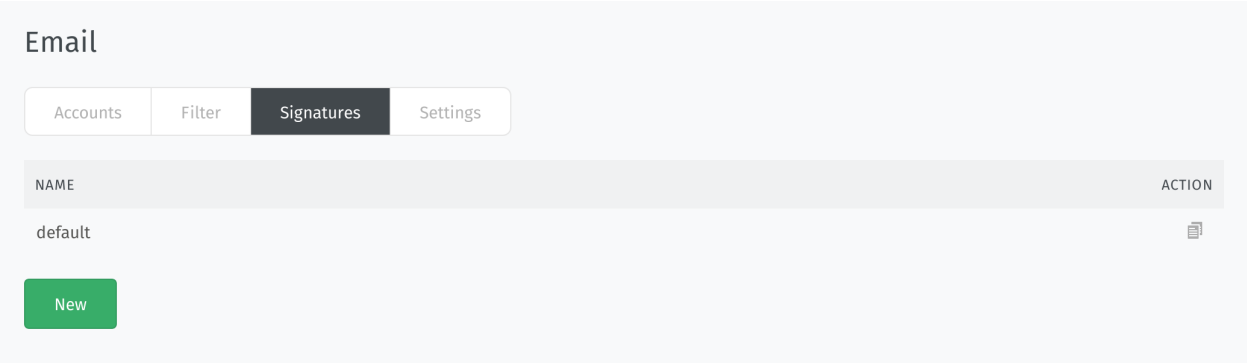
Note that Zammad matches against the full header, e.g. for a mail with „`From: Display Name <display.name@example.com>`“, the `From` condition will test against „`Display Name <display.name@example.com>`“. This is especially important when using anchored regular expressions; `regex:^display\.name@example.com$` would not match this mail!

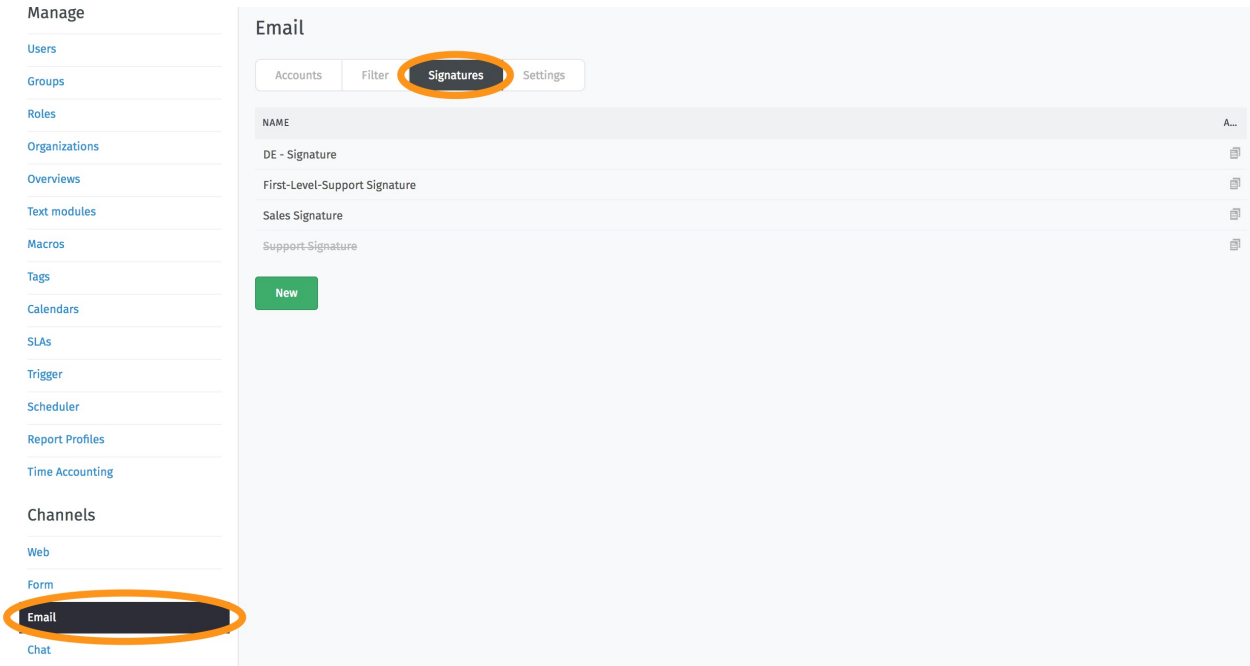
It should be borne in mind that the combined attributes build on each other. If a filter is no longer needed, it can either be temporarily set inactive or deleted directly.

Bemerkung: Signatures in Google channels are **just like signatures in email channels**, so this article is lifted verbatim from *here*.

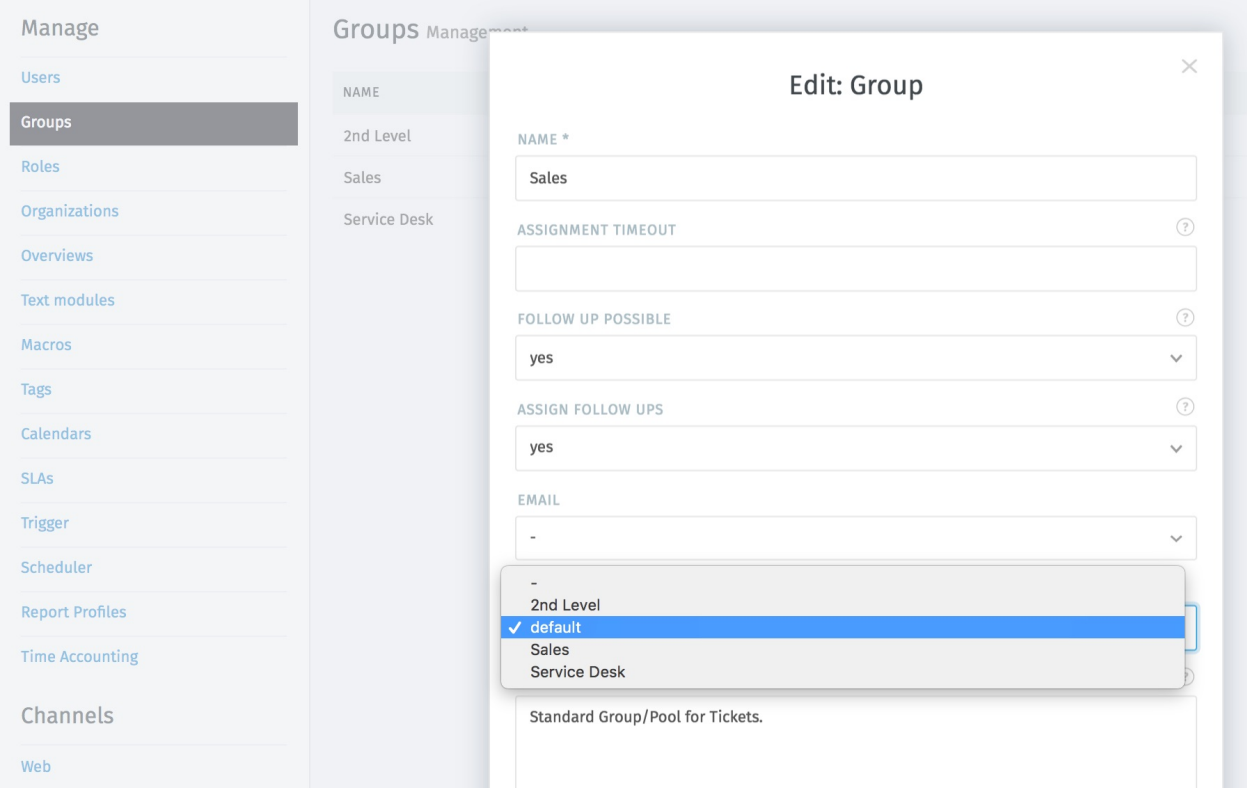
20.3 Signatures

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:

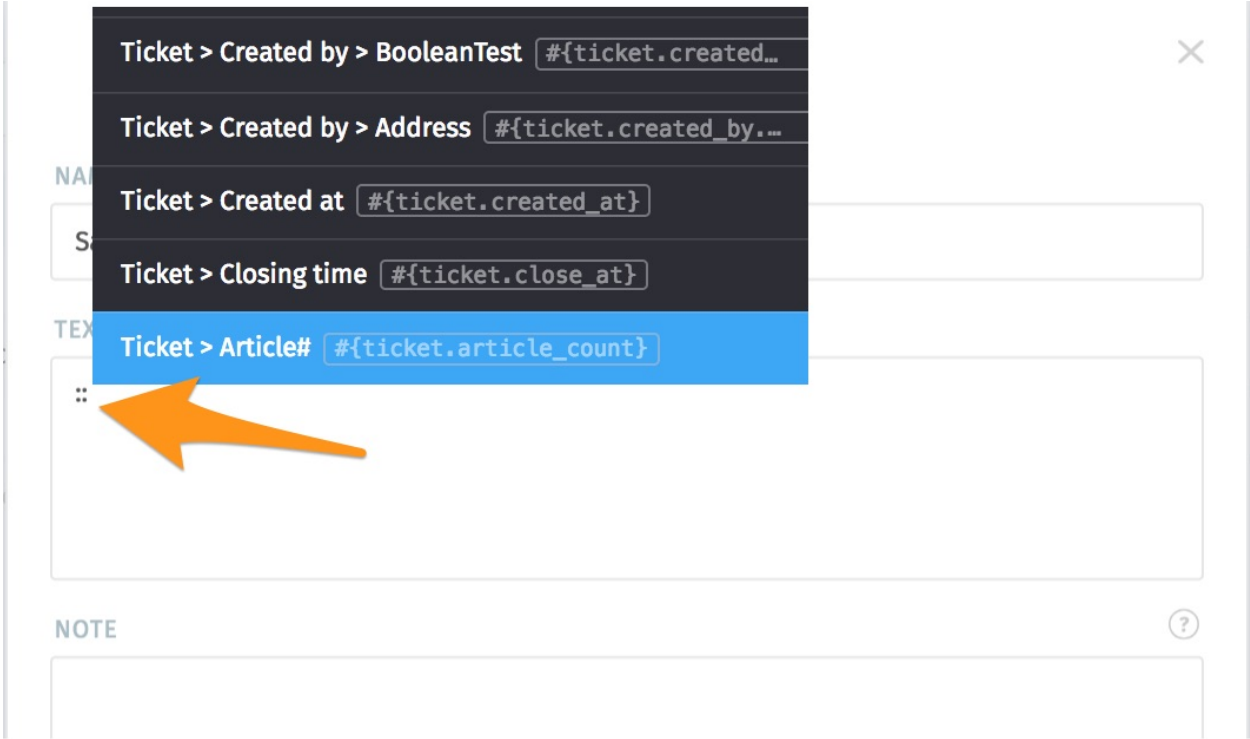


Each group can be assigned its own signature, but they can also all use the same signature.

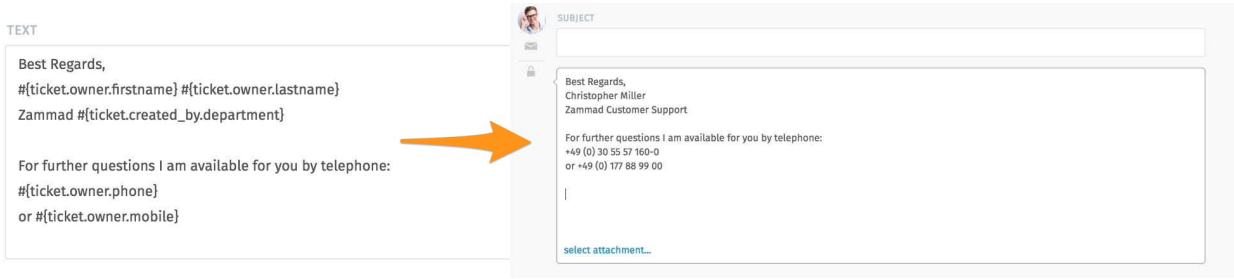
20.3.1 Dynamic Signatures

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.

Hinweis: 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:

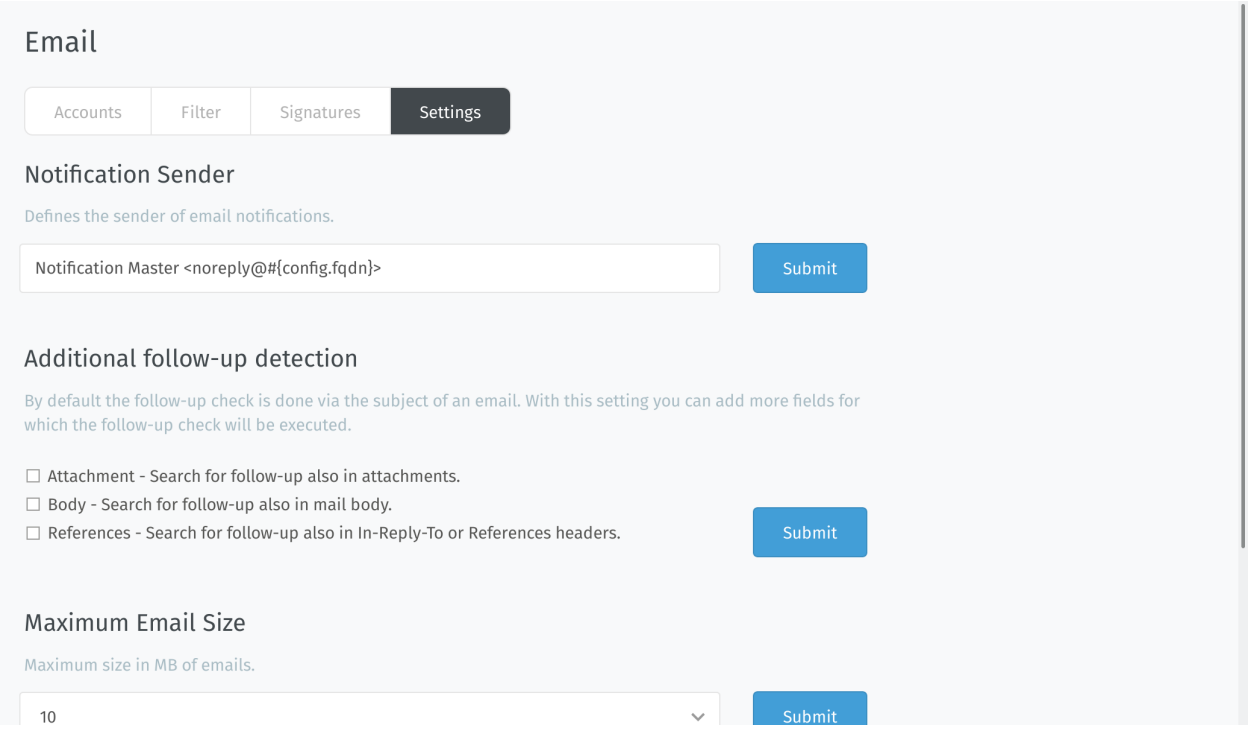


Bemerkung: Settings in Google channels are **just like settings in email channels**, so this article is lifted verbatim from *here*.

20.4 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.

Bemerkung: Some email-related settings are ticket based settings, which is why they can be found in the *Composer Settings*.



20.4.1 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.

> **Bemerkung:** 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.

> **Bemerkung:** Please note that searching in attachment and body might lead to false follow-up detections.

Maximum Email Size: Default value `10 MB` This one is pretty obvious: It defines the maximum allowed size of a Email Zammad will fetch. Zammad will not fetch Mails that are bigger than this option.

> **Bemerkung:** Starting with Zammad 3.2 Zammad can provide postmaster mails (see „Send postmaster mail if mail too large“ below).

> **Hinweis:** This technically also affects attachments for articles.

Send postmaster mail if mail too large: Default value `yes (enabled)` on fresh installations and `no (disabled)` on upgrad

Bemerkung: This option is only available with Zammad 3.2 and later. 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.

Bemerkung: 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 it's 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 `-` or `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 `Take reply-to header as sender/from of email and use realname 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 receive list: Default value `yes` This option decides how Zammad should react if an agent sends a Email to it.

Option set to `yes` Whatever the first user / Email address is set within the recipient list is will be used a ticket customer.

Option set to `no` The agent will be set as ticket customer.

Bemerkung: 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 any way.

The default value is: `(mailer-daemon|postmaster|abuse|root|noreply|noreply.+?|no-reply|no-reply.+?)@.+?`

Sender Format: Default value `Agent + Format Seperator + System Address Display Name` This configures the display name used in the `FROM` header of mails Zammad sends.

Bemerkung: 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 + Format Seperator + System Address Display Name` This will cause Zammad to set the `FROM` header to agent name and the channels display name divided by a seperator (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.`

Sender Format Seperator: Default value `via` This is a can be a String you can freely choose. It divided the agents name and the display name of the channel when ever needed.

Ticket Subject Forward: Default value `FWD` The above string will be used on the subject if you forward a mail from Zammad.

Bemerkung: `:` 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.

Bemerkung: `:` 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]`

Bemerkung: This does **not** limit ticket titles within the UI, just the subjects during replying to an Email.

20.4.2 Enhanced settings

Some less relevant settings can be changed via rails console if needed. As 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.

Connect a Gmail or G Suite account to Zammad.

Bemerkung: 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.

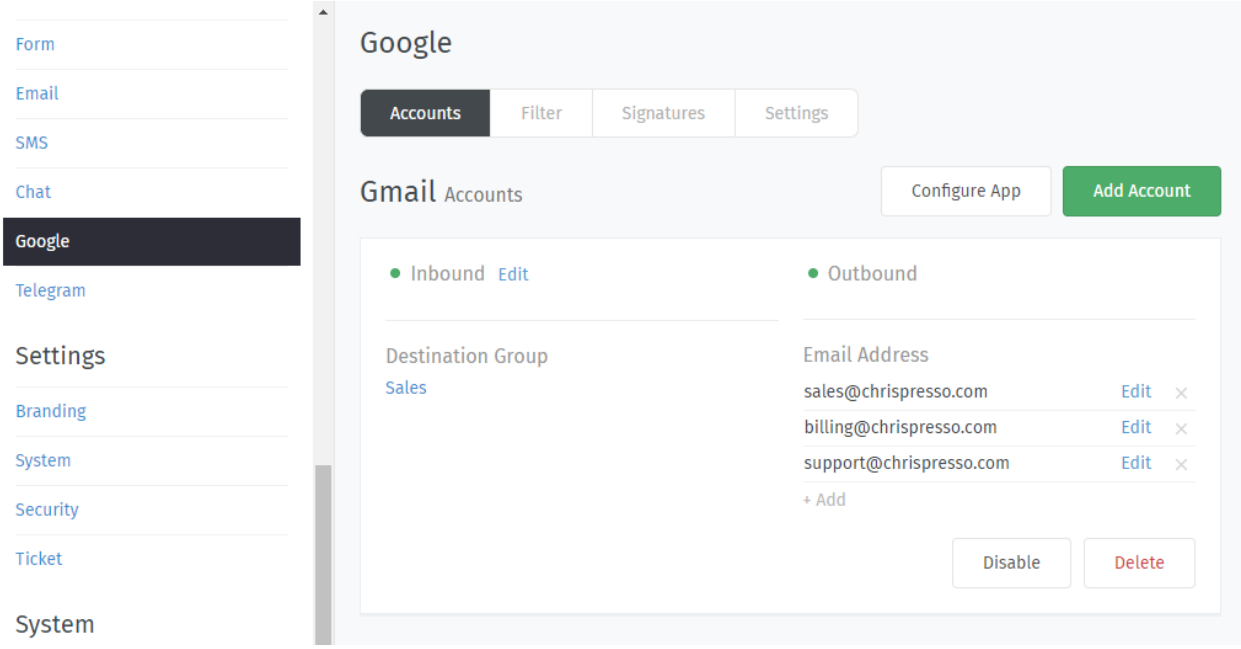
(Self-hosted users may have already completed this step during new system setup.)

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:")

Hinweis: Want to **manually edit email subjects** or **always copy parent messages into your replies**?

Check the *Composer Settings*.

KAPITEL 21

Microsoft 365

21.1 Accounts

21.1.1 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).

Bemerkung: 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*).

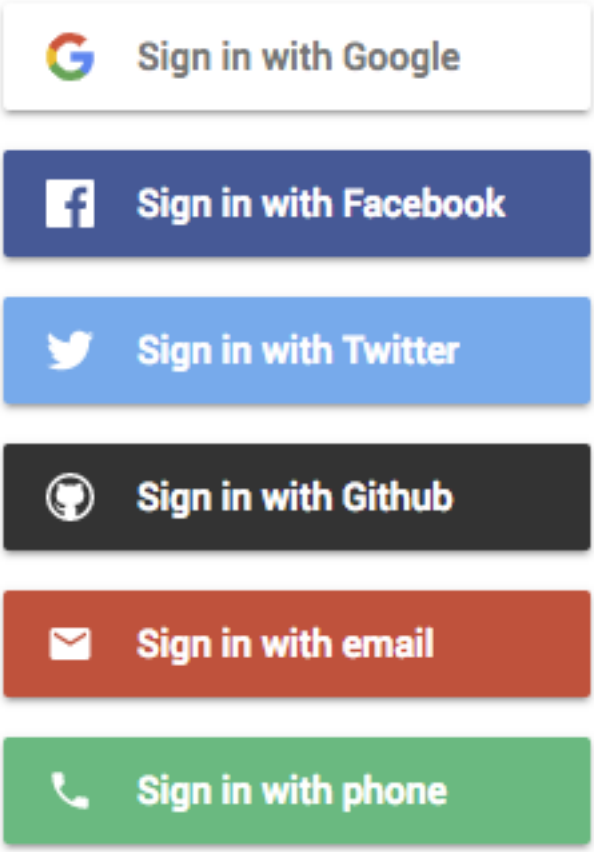
When a third-party website wants to use OAuth, it has to **register with the provider first** (*i.e.,* Microsoft). This way, the provider knows who's receiving its users' data, and can give users a way to revoke access if they change their minds.

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.

Bemerkung: Use an admin account for your organization.


Sign in with Google
Sign in with Facebook
Sign in with Twitter
Sign in with Github
Sign in with email
Sign in with phone

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 **Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)**.

   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. **Connect your Microsoft app in Zammad**

   Copy your new app registration's **Application (client) 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.

21.1.2 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:

> **Gefahr: Zammad will delete all emails in your inbox during the import process.**
>
> Use the *Keep Messages on Server* setting to *disable this behavior*.

Warnung: 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 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). **But there's no rush just yet**.

Upcoming versions of Zammad will feature an automated migration wizard to help you make the switch, and it'll be available well before Microsoft officially pulls the plug on password auth.

Hinweis: If you'd prefer not to wait, you can do it manually today—just remember to **delete the email channel for your Microsoft account** (and all its associated aliases) before re-adding it here.

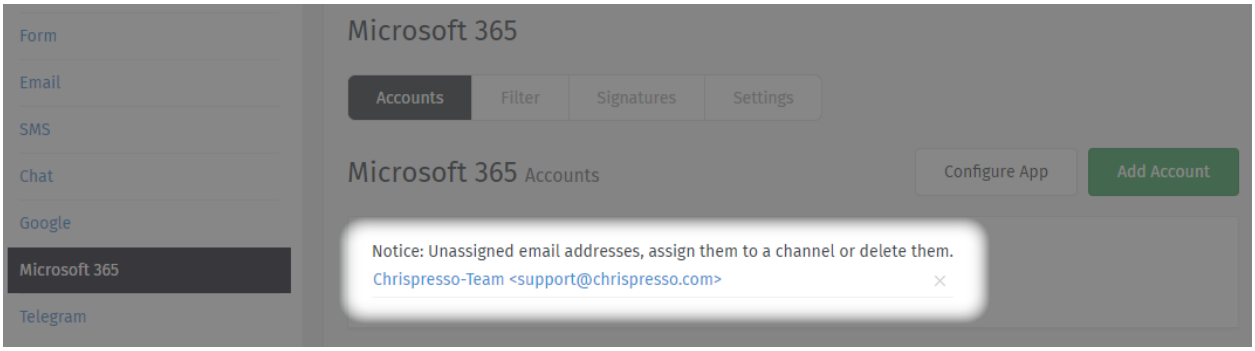


Abb. 1: Make sure you don't have any of these left over before creating your new Microsoft 365 channel.

Add a New Account

Click **Add Account** to connect your Microsoft 365 / Outlook accounts to Zammad. You will be redirected to a Microsoft sign-in and confirmation page.

Bemerkung: Aliases are not imported automatically.

See *Secondary Addresses* to add them yourself.

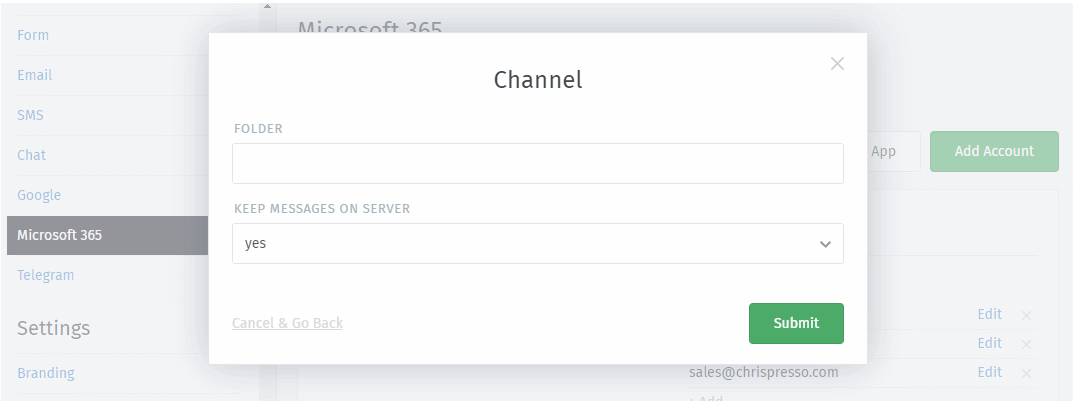
Channel

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.)

Bemerkung: 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.

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?

Bemerkung: Secondary addresses in Microsoft 365 channels work (almost) **just like they do in email channels**, so this article is lifted (almost) verbatim from *here*.

21.1.3 Secondary Addresses

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

Abb. 2: Once you add a secondary address, you can configure a group to start sending emails with it.

Warnung: 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.

Display Name The display name used for outgoing email.

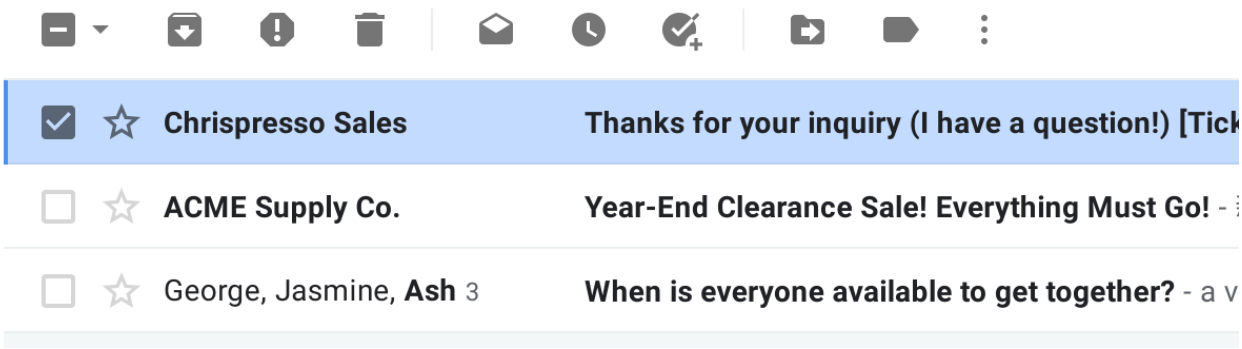


Abb. 3: A customer's inbox with an auto-reply from **Chrispresso Sales**.

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.

Bemerkung: Managing accounts in Microsoft 365 channels is (almost) **just like it is in email channels**, so this article is lifted (almost) verbatim from *here*.

21.1.4 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.

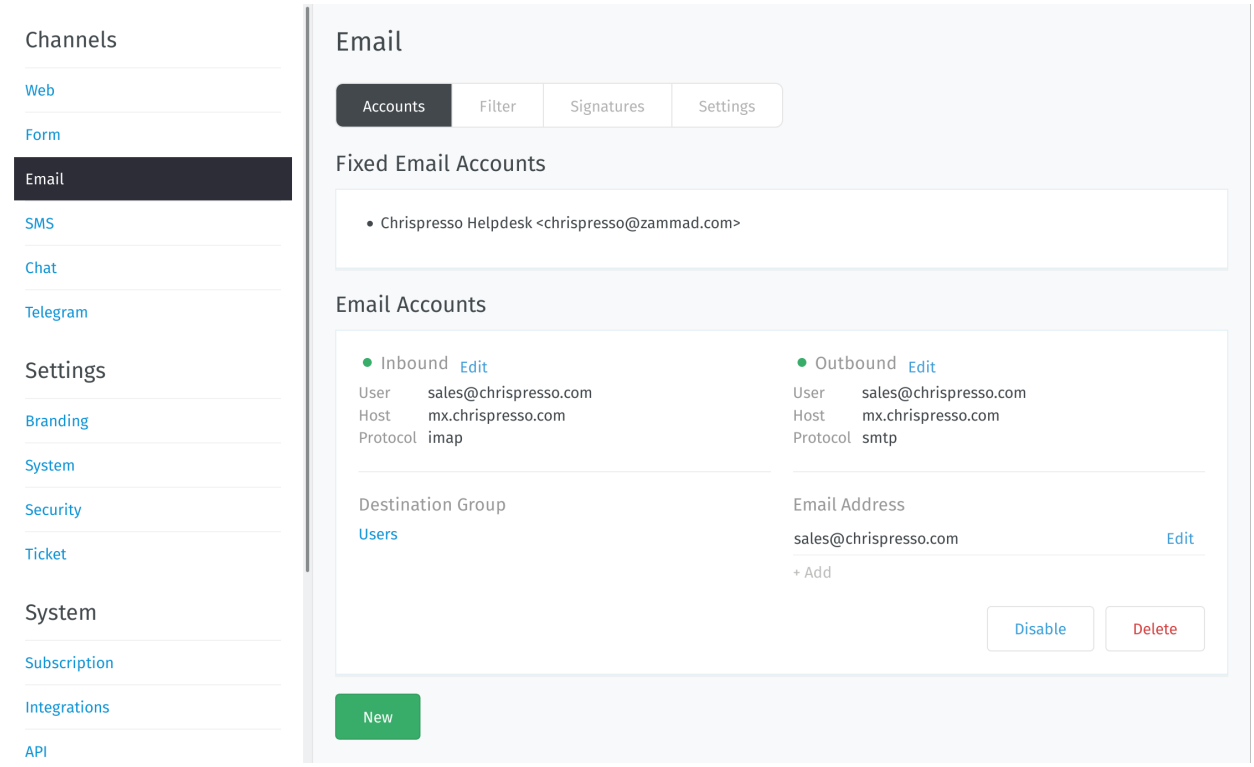
Only **active** groups will be displayed.

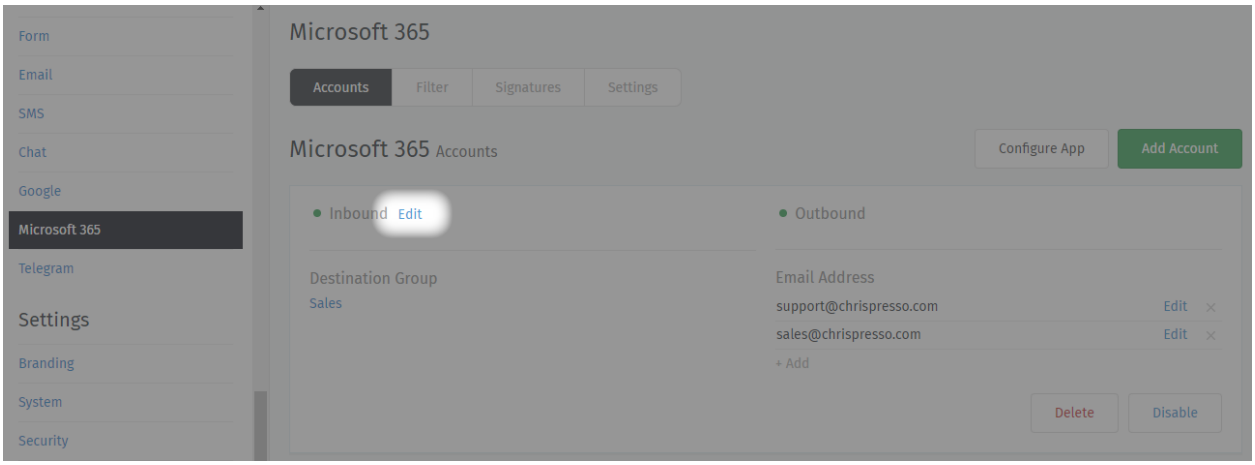
Changing this setting will not reassign existing tickets to the new group.

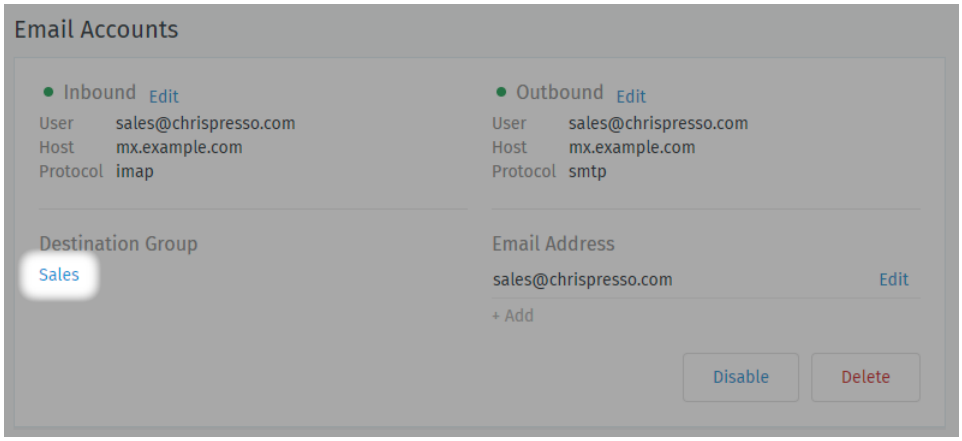
Hinweis: **Still can't send outgoing email tickets?** Check your *group settings*.

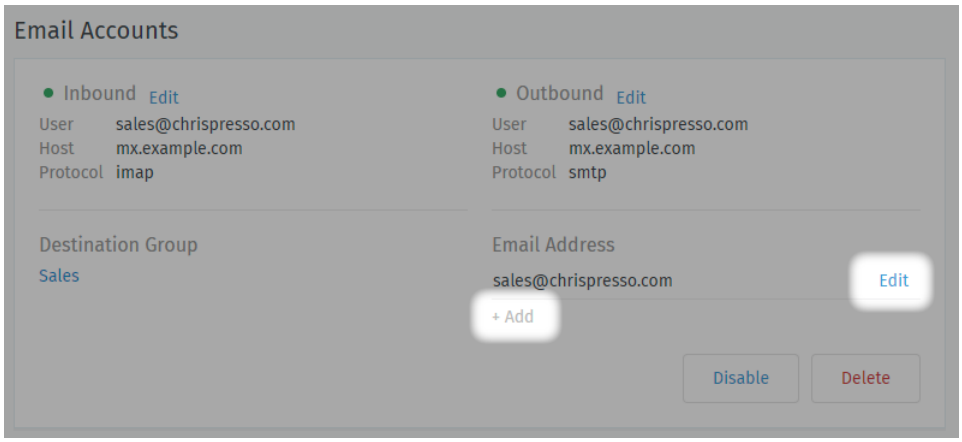
Email Address Use the **+ Add** or **Edit** buttons to set up secondary addresses on this account.

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


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
Microsoft 365
Telegram
Settings
Branding
System
Security
Microsoft 365
Accounts
Filter
Signatures
Settings
Microsoft 365 Accounts
Configure App
Add Account
Inbound Edit
Outbound
Destination Group
Sales
Email Address
support@chrispresso.com Edit
sales@chrispresso.com Edit
+ Add
Delete
Disable


Email Accounts
Inbound Edit
User sales@chrispresso.com
Host mx.example.com
Protocol imap
Outbound Edit
User sales@chrispresso.com
Host mx.example.com
Protocol smtp
Destination Group
Sales
Email Address
sales@chrispresso.com Edit
+ Add
Disable
Delete



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.

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

Delete Deleting an account removes its configuration from Zammad entirely.

Bemerkung: 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.

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**.

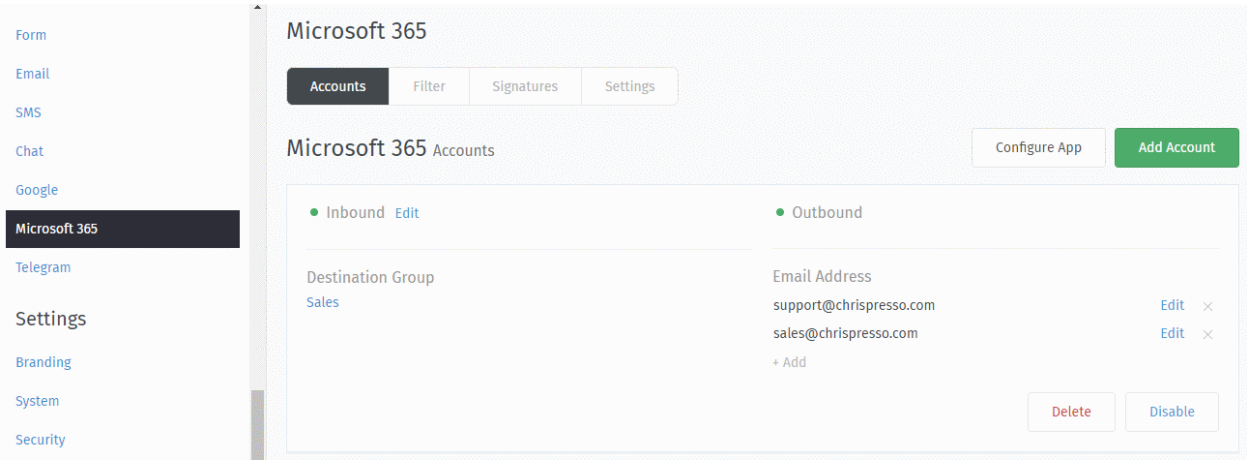
Bemerkung: How do I use my Microsoft 365 account for outgoing system notifications?

On **subscription/cloud-hosted instances**, you can't. Notifications will always come from "Notification Master <noreply@your.zammad.domain>".

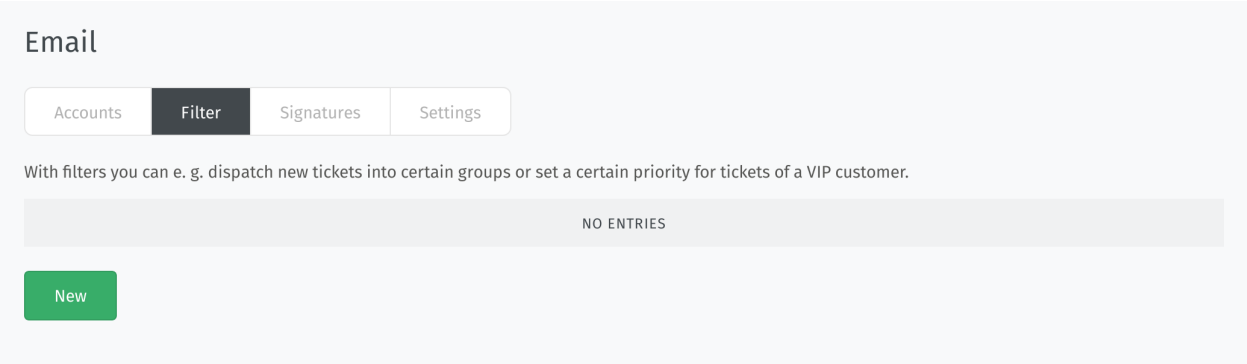
On **self-hosted instances**, we still don't recommend it. Using a Microsoft account for automated, outgoing messages is risky: users who exceed Microsoft's email sending limits can have their accounts suspended.

Set up a generic *email channel* instead, then use the *Email Notification* setting.

Bemerkung: Filters in Microsoft 365 channels are **just like filters in email channels**, so this article is lifted verbatim from *here*.



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

  Bemerkung: 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

Bemerkung: 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 an external spam filter (e.g. SpamAssassin):

  X-Spam-Flag: contains: `YES`
  Tag: add: `spam`
  State: closed

  Bemerkung: 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.

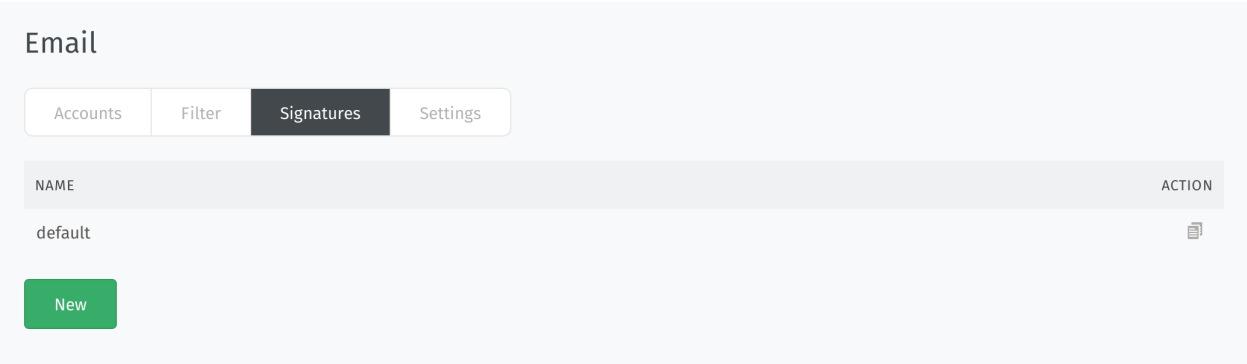
Different attributes of a filter can be combined with each other. Likewise, the following actions can be combined. The supported matches are „contains“ and „contains not“; for advanced matching, you can use regular expressions by prefixing the string with `regex:`.

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!

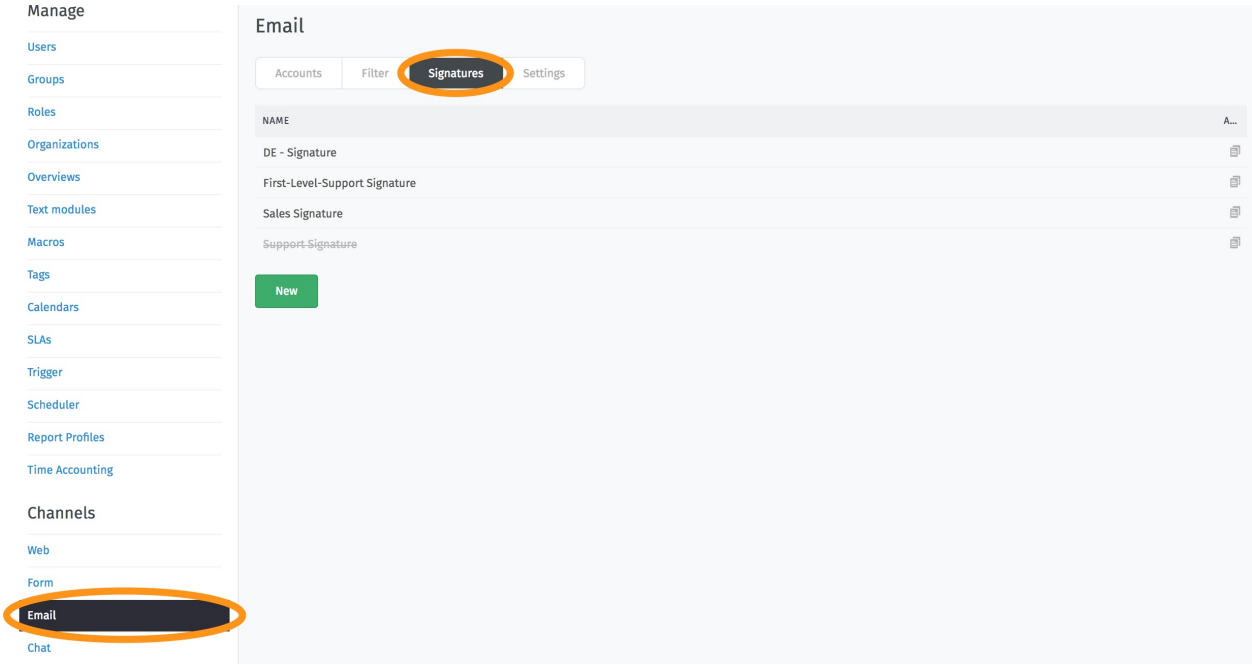
It should be borne in mind that the combined attributes build on each other. If a filter is no longer needed, it can either be temporarily set inactive or deleted directly.

Bemerkung: Signatures in Microsoft 365 channels are **just like signatures in email channels**, so this article is lifted verbatim from *here*.

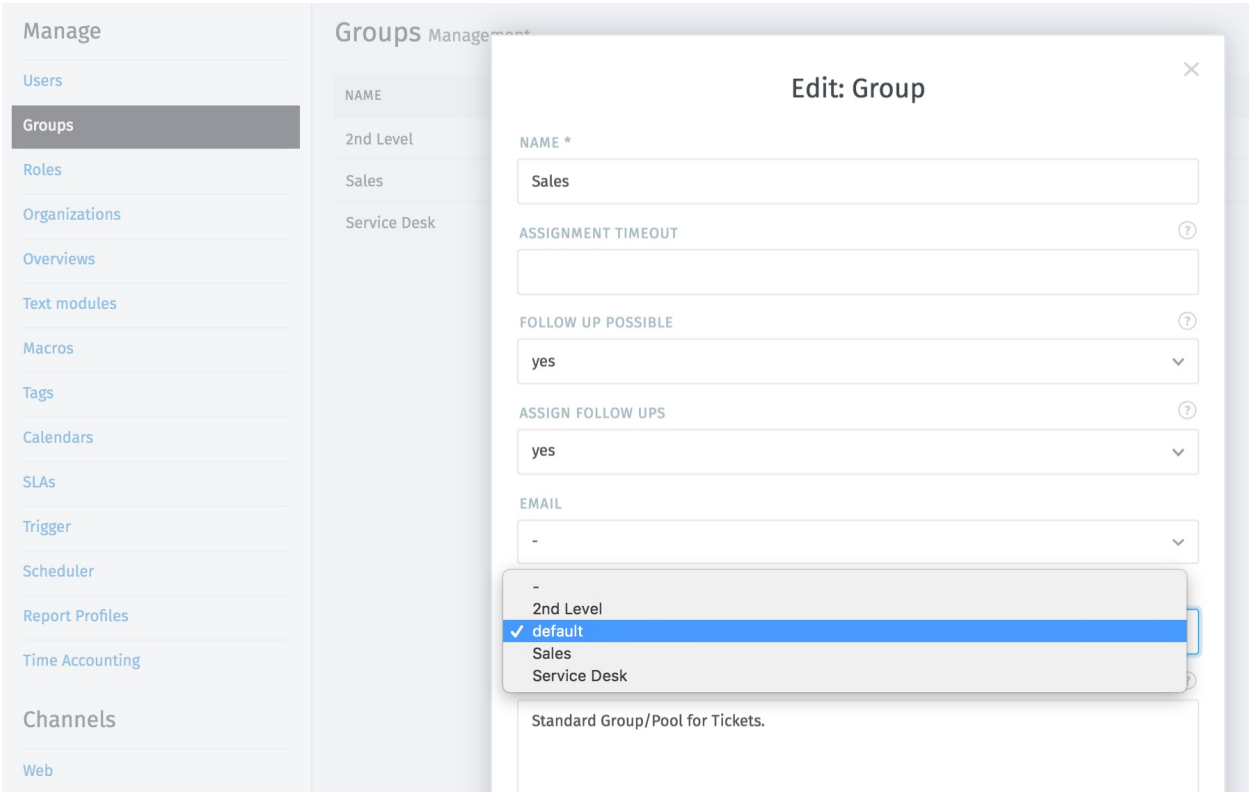
21.3 Signatures



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:

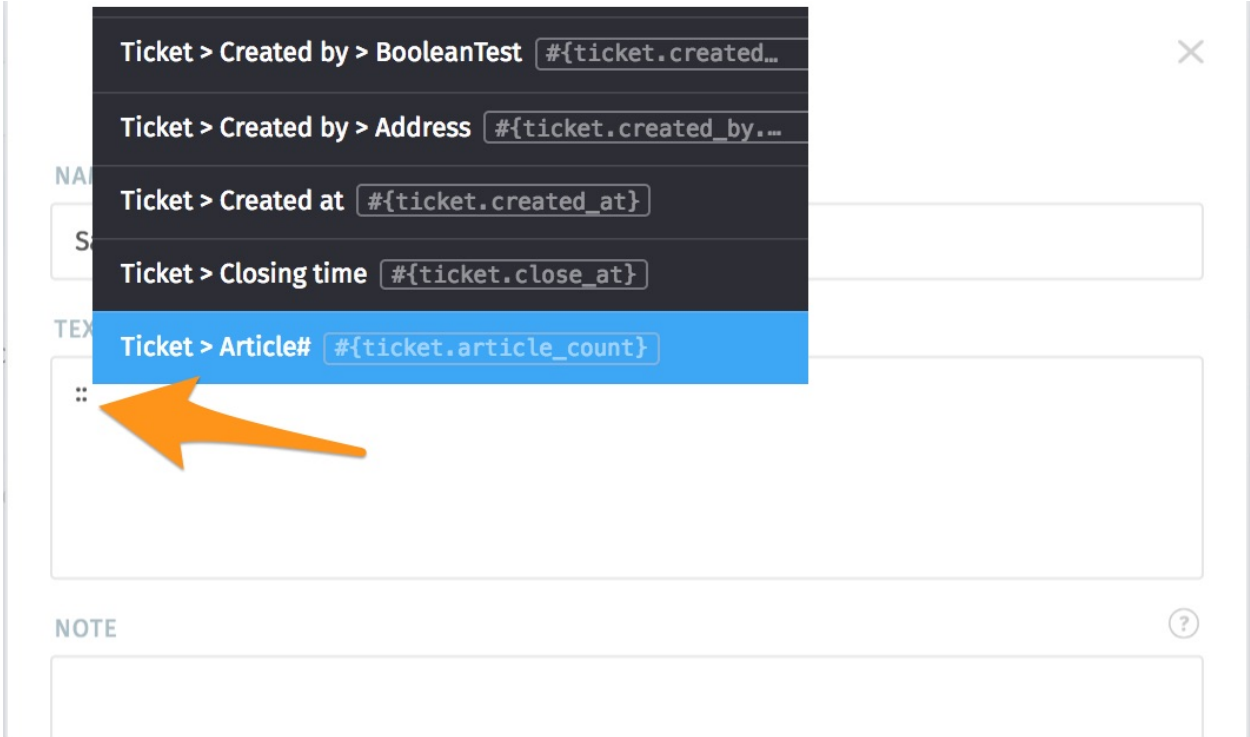


Each group can be assigned its own signature, but they can also all use the same signature.

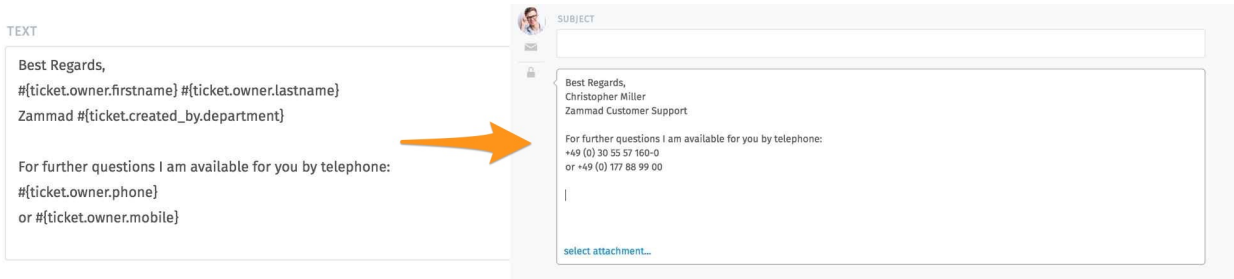
21.3.1 Dynamic Signatures

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.

Hinweis: 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.

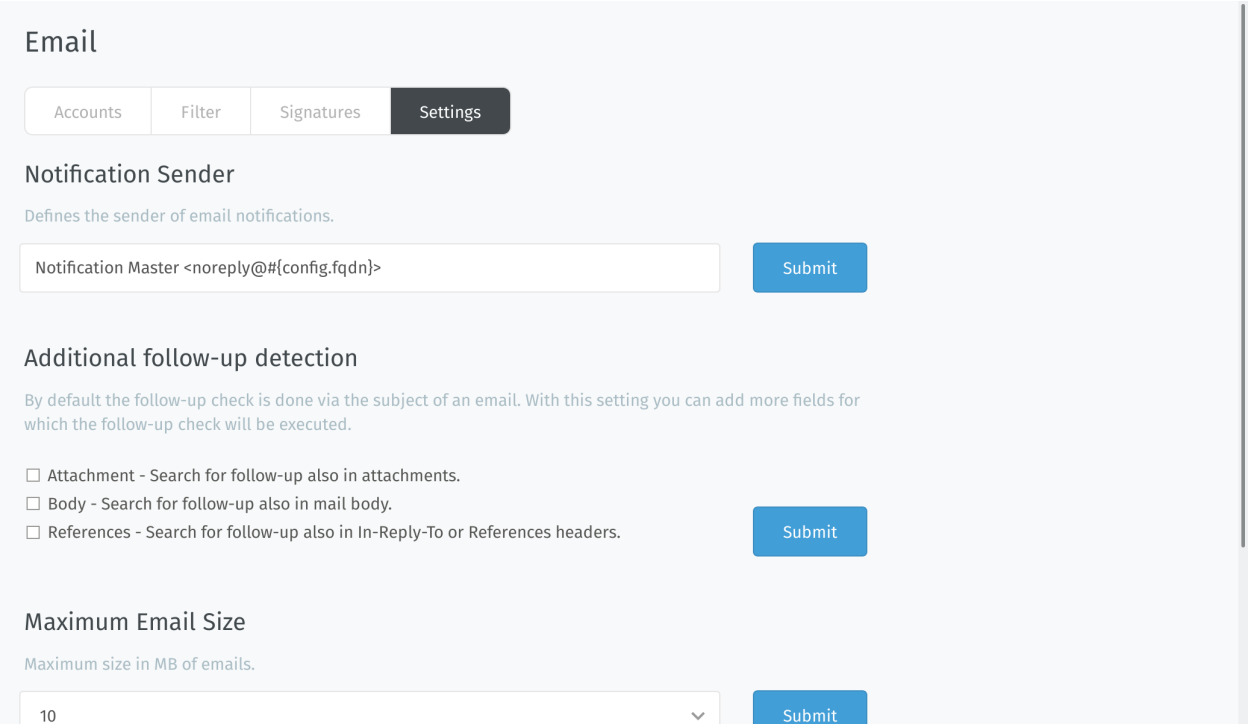


Here is an example of a signature with variables and the result when you write a mail:



Bemerkung: Settings in Microsoft 365 channels are **just like settings in email channels**, so this article is lifted verbatim from *here*.

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.

Bemerkung: 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.

Bemerkung: 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.

Bemerkung: Please note that searching in attachment and body might lead to false follow-up detections.

Maximum Email Size: Default value `10 MB` This one is pretty obvious: It defines the maximum allowed size of a Email Zammad will fetch. Zammad will not fetch Mails that are bigger than this option.

Bemerkung: Starting with Zammad 3.2 Zammad can provide postmaster mails (see „Send postmaster mail if mail too large“ below).

Hinweis: This technically also affects attachments for articles.

Send postmaster mail if mail too large: Default value `yes (enabled)` on fresh installations and `no (disabled)` on upgrad

Bemerkung: This option is only available with Zammad 3.2 and later. 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.

Bemerkung: 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 it's 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 `-` or `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 `Take reply-to header as sender/from of email and use realname 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 receive list: Default value `yes` This option decides how Zammad should react if an agent sends a Email to it.

Option set to `yes` Whatever the first user / Email address is set within the recipient list is will be used a ticket customer.

Option set to `no` The agent will be set as ticket customer.

Bemerkung: 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 any way.

The default value is: `(mailer-daemon|postmaster|abuse|root|noreply|noreply.+?|no-reply|no-reply.+?)@.+?`

Sender Format: Default value `Agent + Format Seperator + System Address Display Name` This configures the display name used in the `FROM` header of mails Zammad sends.

> **Bemerkung:** 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 + Format Seperator + System Address Display Name` This will cause Zammad to set the `FROM` header to agent name and the channels display name divided by a seperator (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.`

Sender Format Seperator: Default value `via` This is a can be a String you can freely choose. It divided the agents name and the display name of the channel when ever needed.

Ticket Subject Forward: Default value `FWD` The above string will be used on the subject if you forward a mail from Zammad.

> **Bemerkung:** `:` 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.

> **Bemerkung:** `:` 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]`

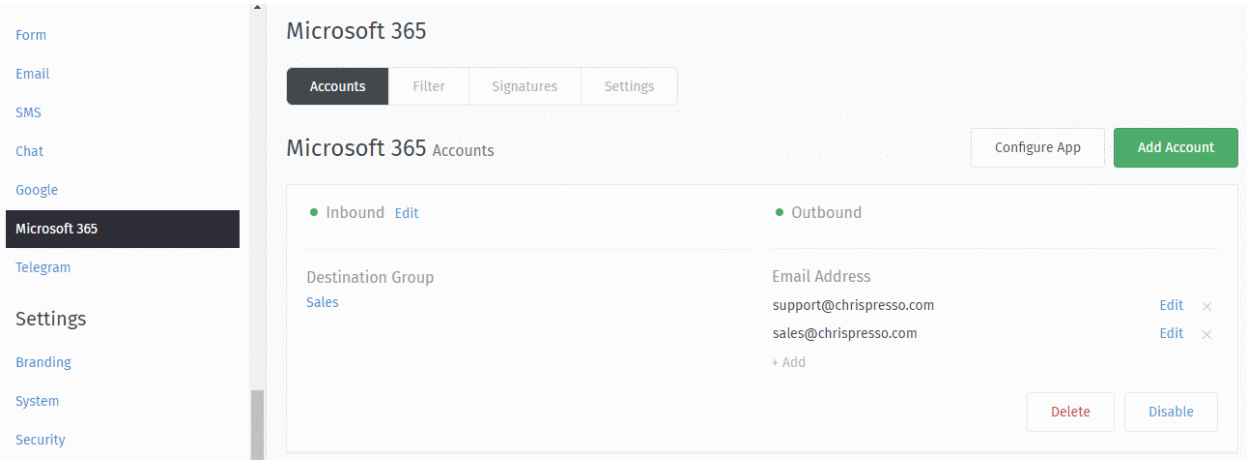
> **Bemerkung:** This does **not** limit ticket titles within the UI, just the subjects during replying to an Email.

21.4.2 Enhanced settings

Some less relevant settings can be changed via rails console if needed. As 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.

Connect a Microsoft 365 account (formerly "Office 365") to Zammad.

> **Bemerkung:** Microsoft 365 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 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:")

Hinweis: Want to **manually edit email subjects** or **always copy parent messages into your replies**?

Check the *Composer Settings*.

KAPITEL 22

Twitter

Zammad supports Twitter integration, meaning you can send and receive tweets and DMs just like emails!

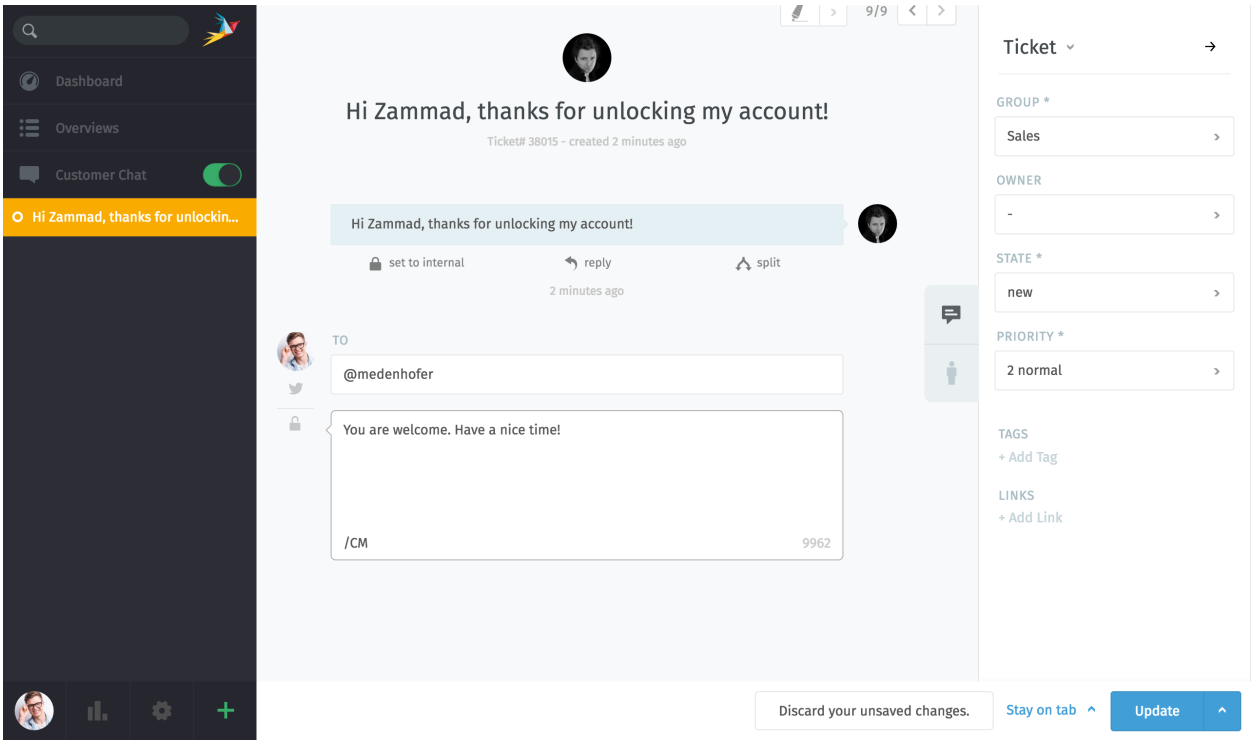


Abb. 1: Twitter tickets will show a Twitter bird in the reply area. Just click on the reply button (as you would for an email) to tweet back.

Bemerkung: 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.

22.1 1. Apply for a Twitter Developer account

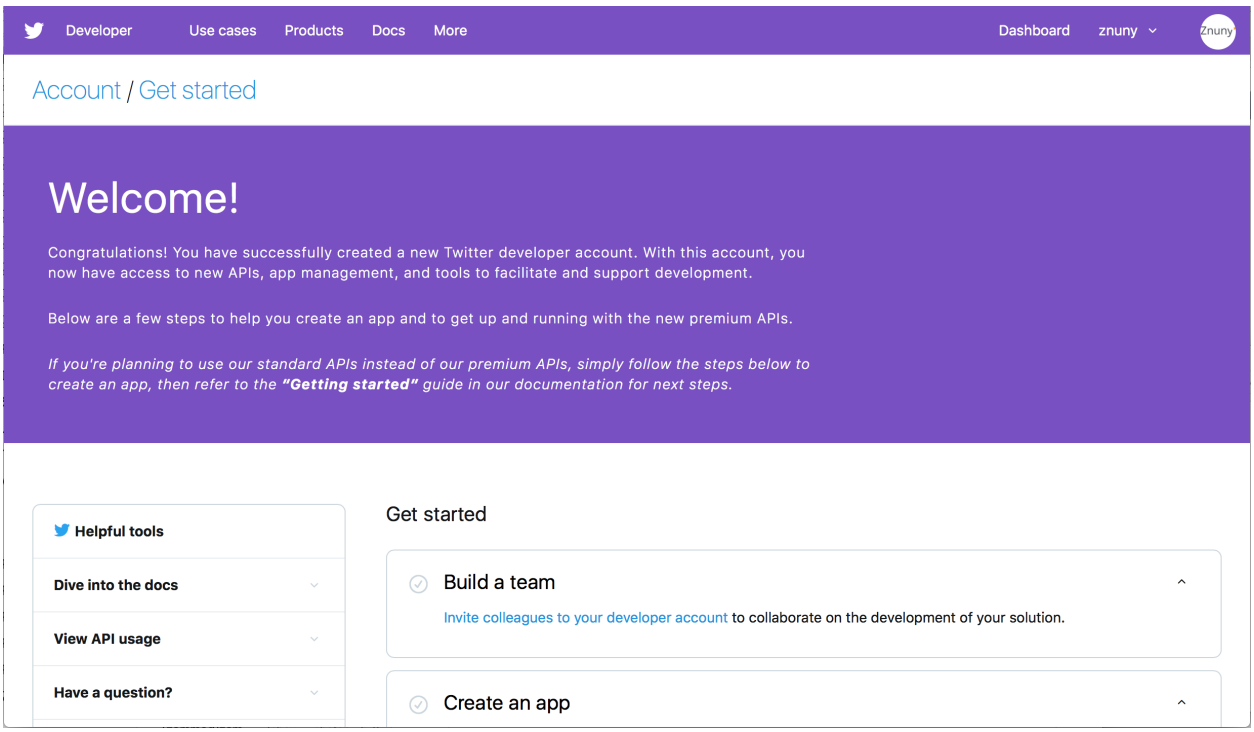


Abb. 2: This welcome page is displayed after completing the application for a Developer account.

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.
```

22.2 2. Create a new Twitter app for Zammad

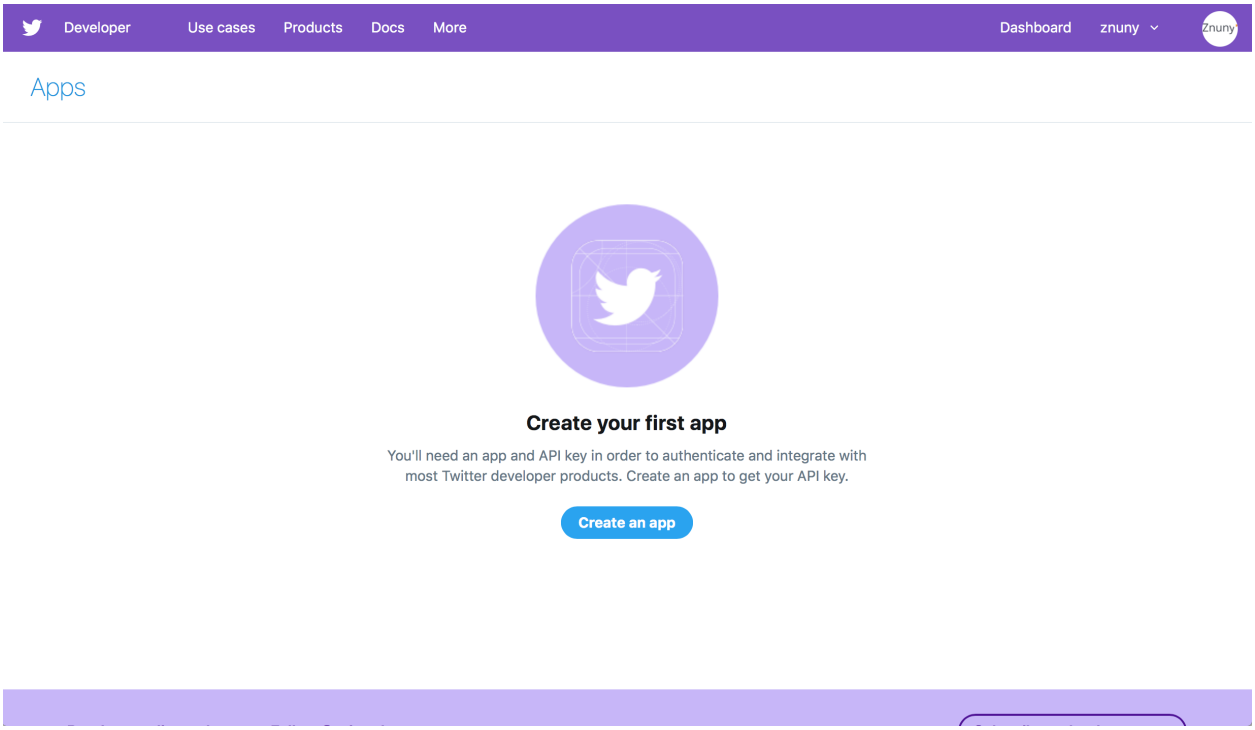


Abb. 3: To create a new app, select **Apps** under your user menu, then click **Create an app**.

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.
```

22.3 3. Set your new app's permissions

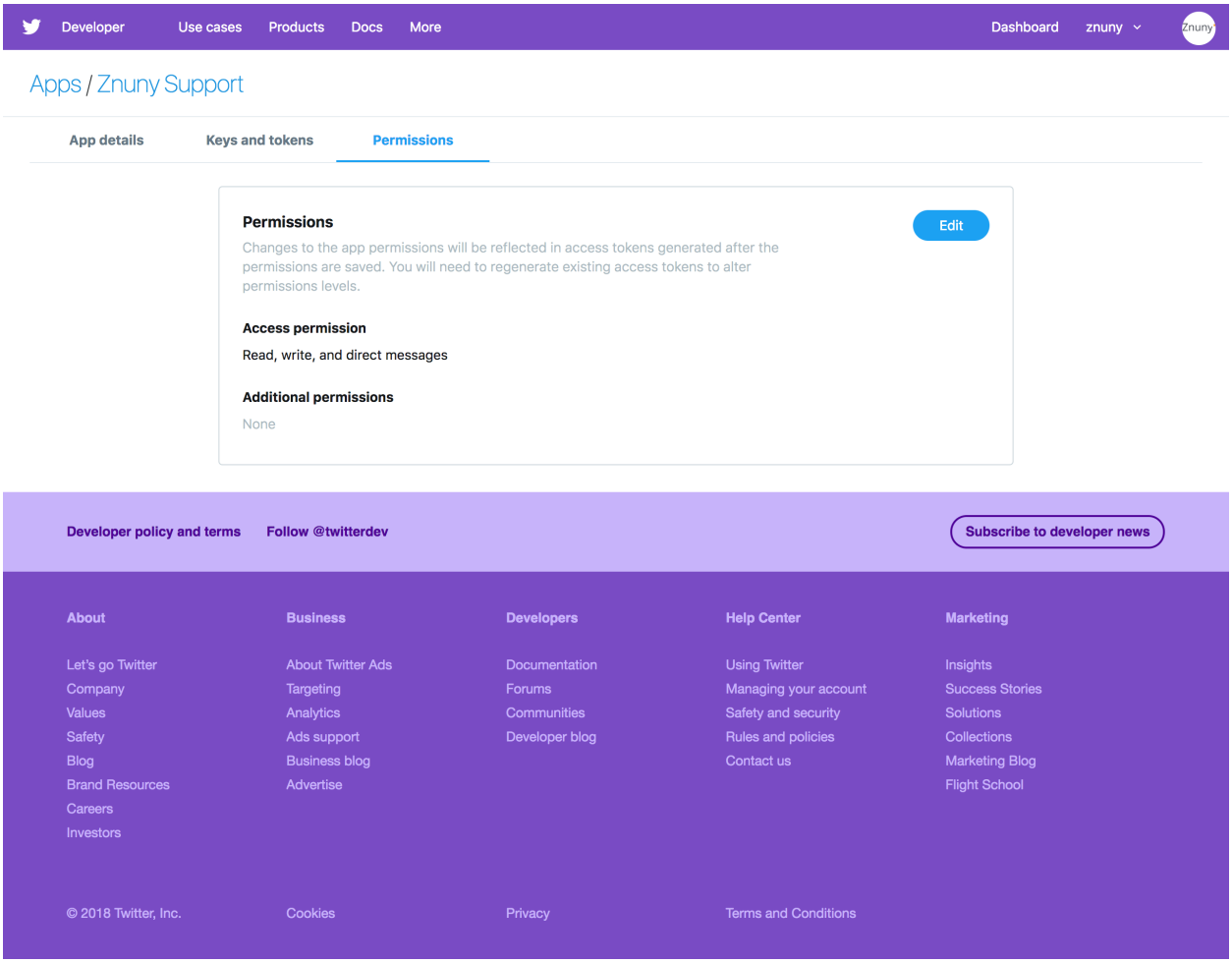


Abb. 4: Select *Read, write, and access direct messages.*

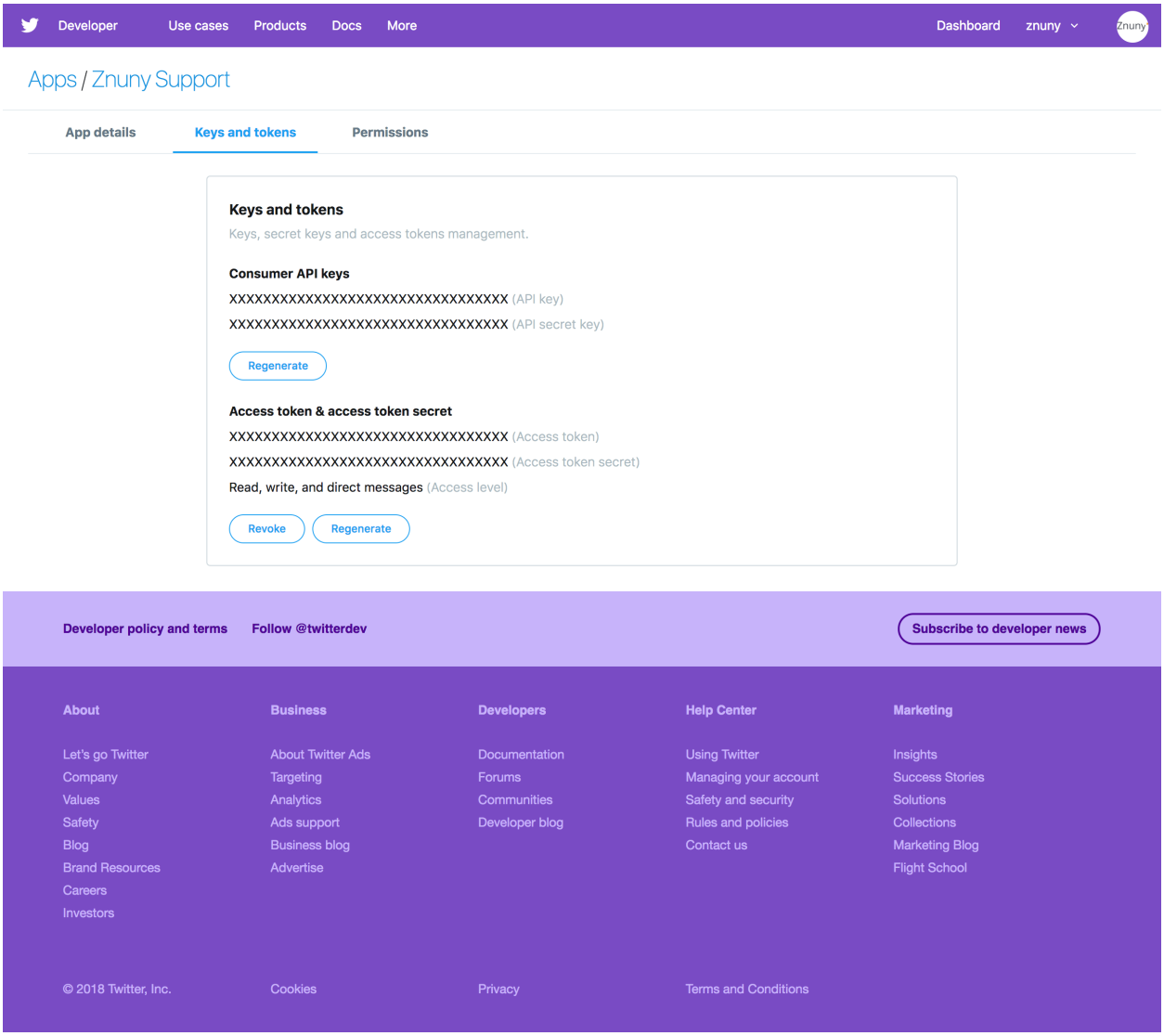


Abb. 5: You will need all four keys/tokens later, so don't close this tab.

22.4 4. Generate a new access token & secret

22.5 5. Set up a dev environment

After creating your app, set up a dev environment for the Account Activity API.

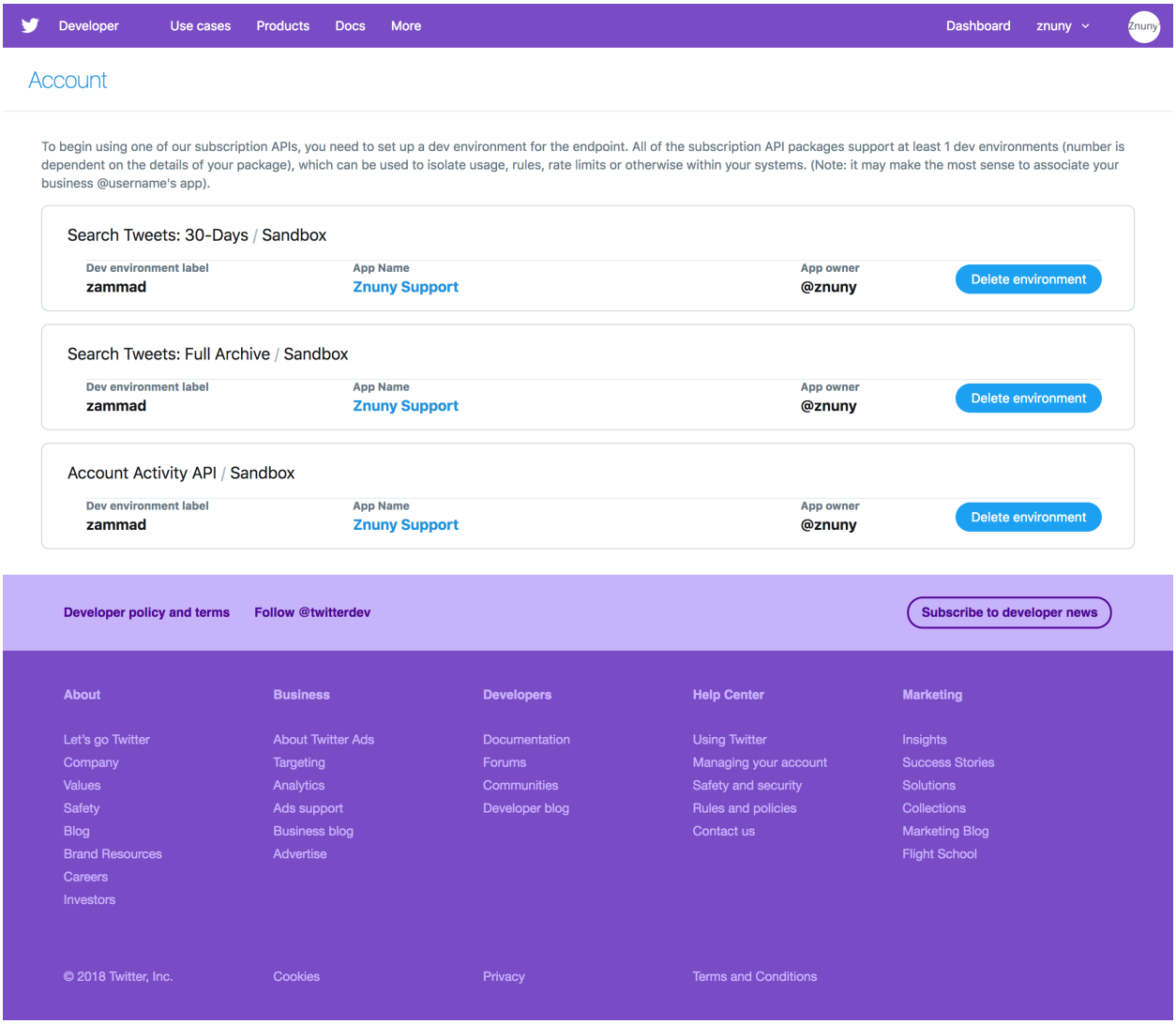


Abb. 6: Name it whatever you like (*e.g.*, `zammad`). You will need the label later, so don't forget it.

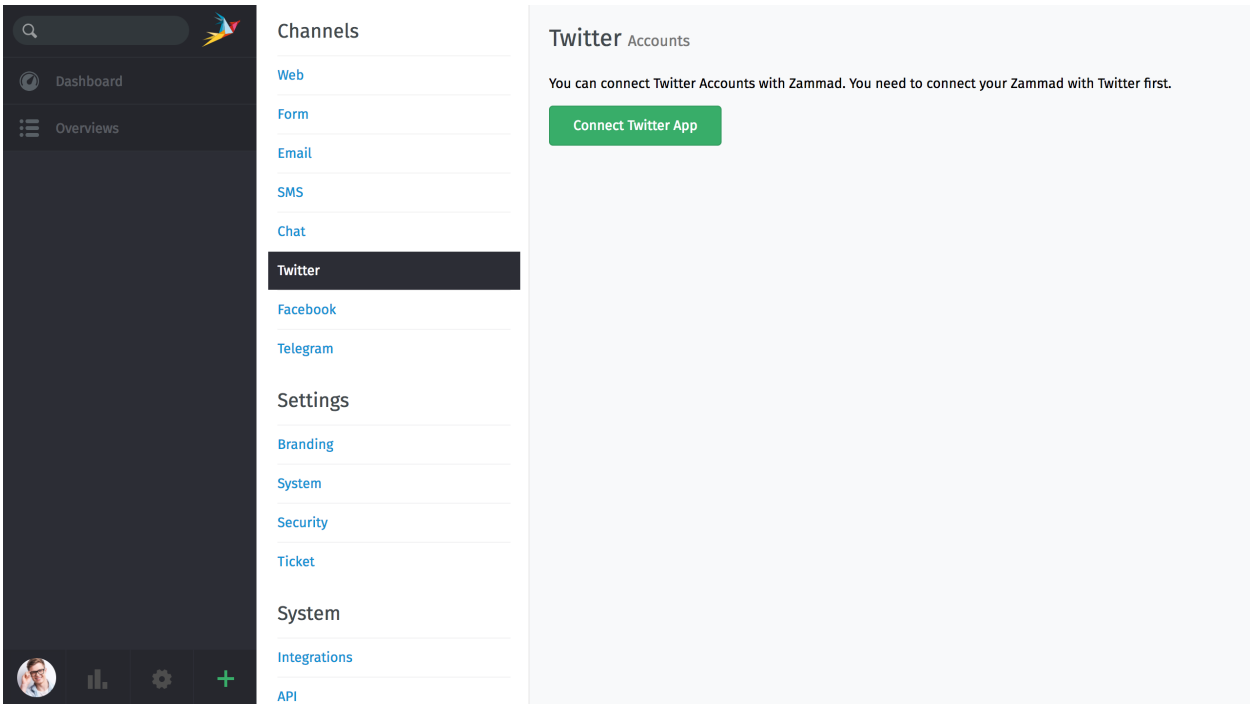


Abb. 7: Add your new Twitter app under **Channels > Twitter** in the admin settings panel. You will need the keys, tokens, and dev environment label from Steps 4 and 5.

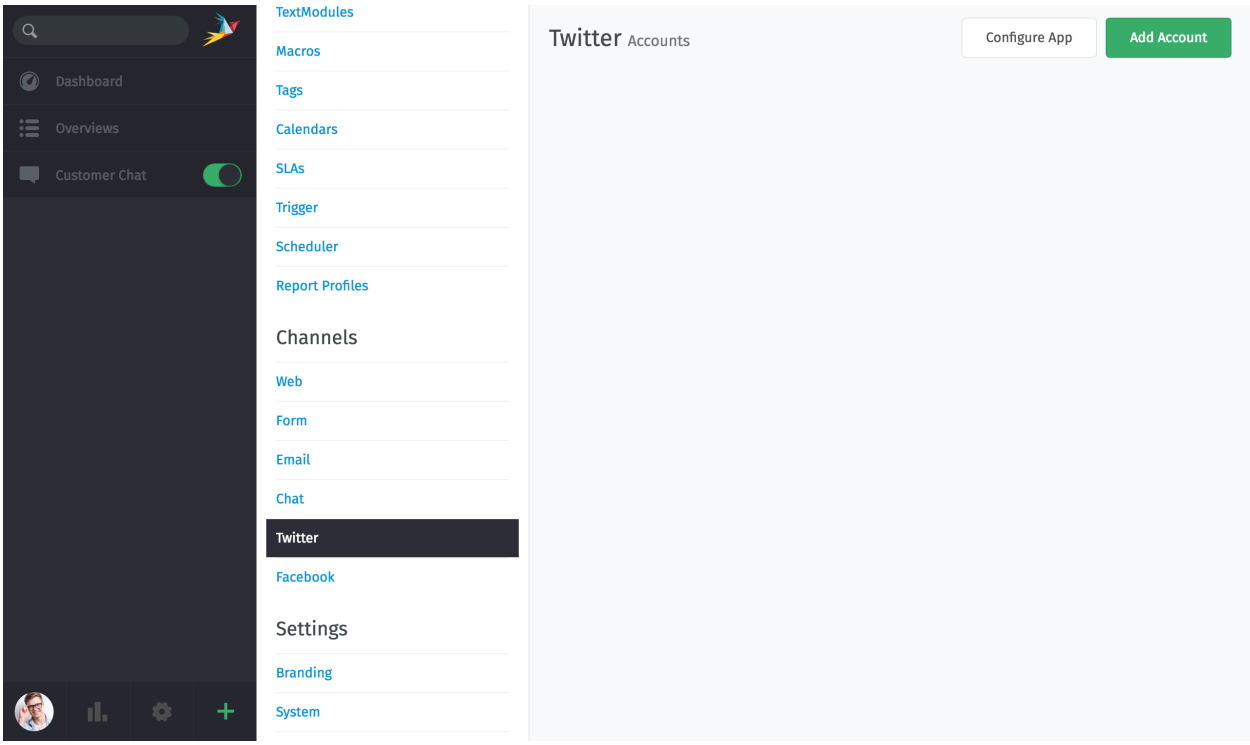


Abb. 8: Click **Add Account** under **Channels > Twitter** in the admin settings panel.

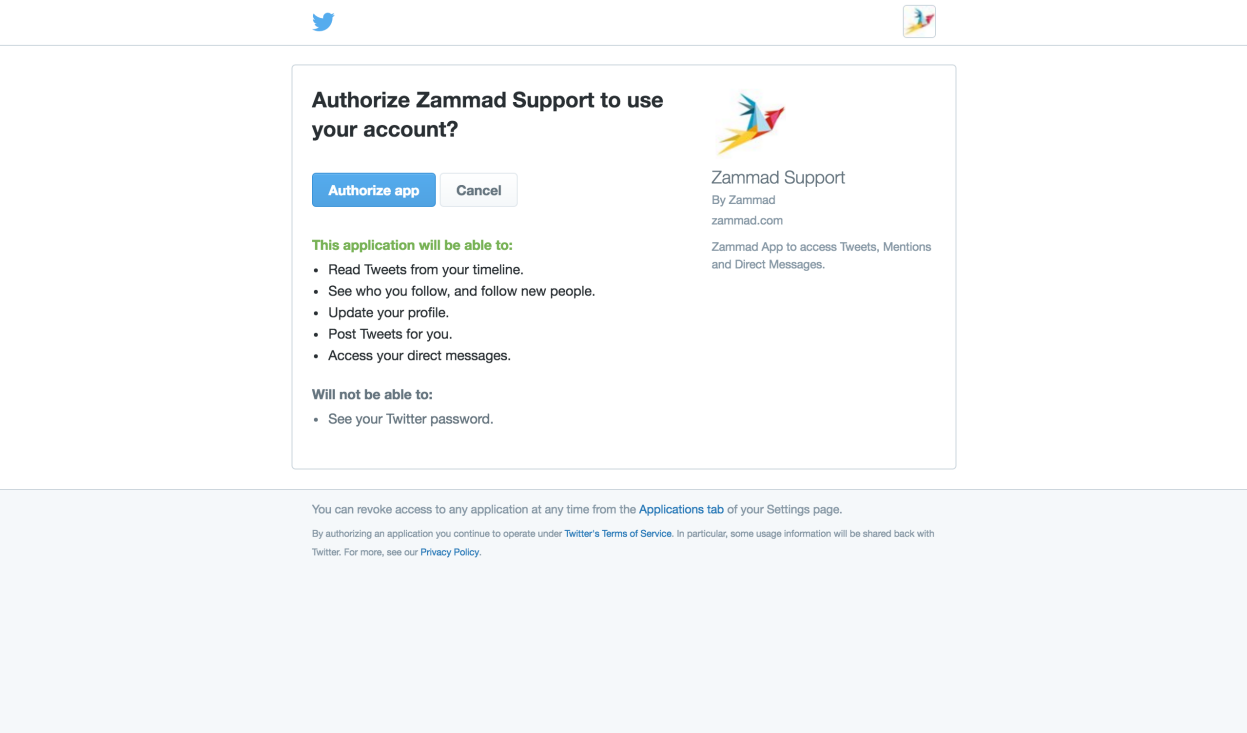


Abb. 9: You will be redirected to Twitter and asked to authorize Zammad to access your account.

22.6 6. Add your new Twitter app in Zammad

22.7 7. Add your Twitter account in Zammad

22.8 8. Configure filters

That's it! Now, incoming tweets and DMs will be automatically turned into Zammad tickets.

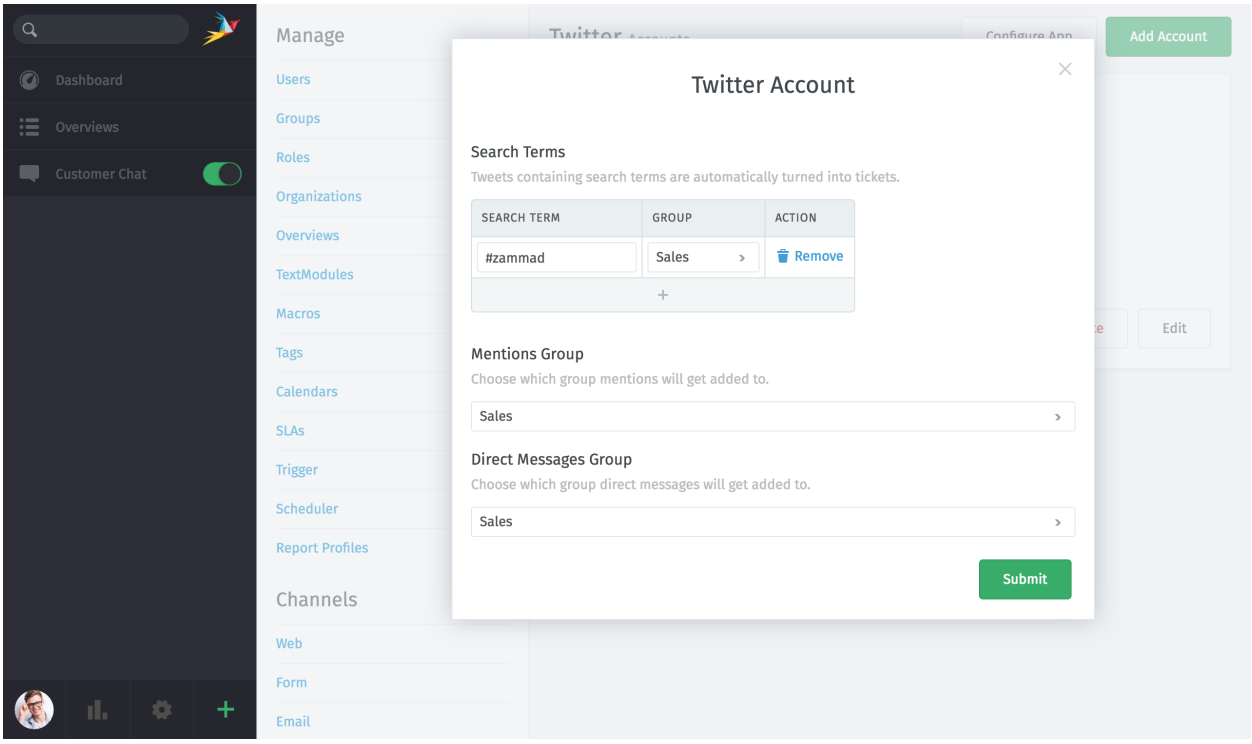


Abb. 10: Set up filters to automatically create new tickets based on **#tags** or **@mentions**.

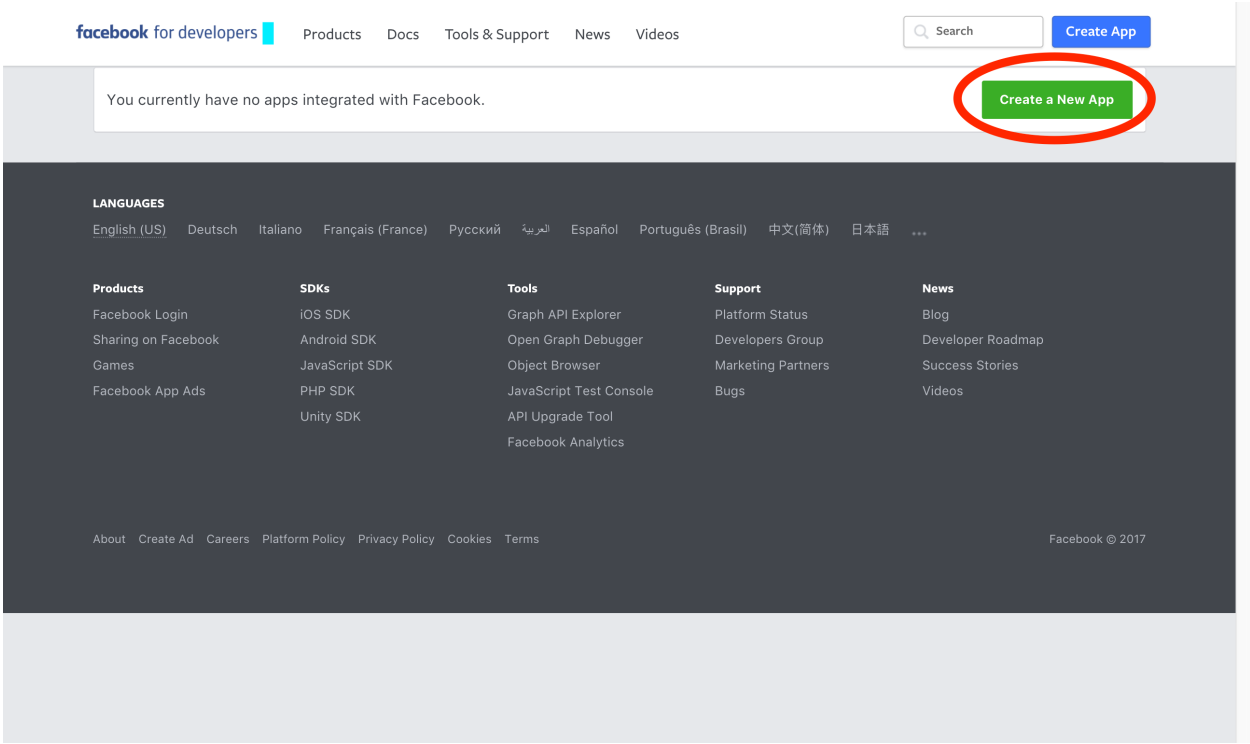
KAPITEL 23

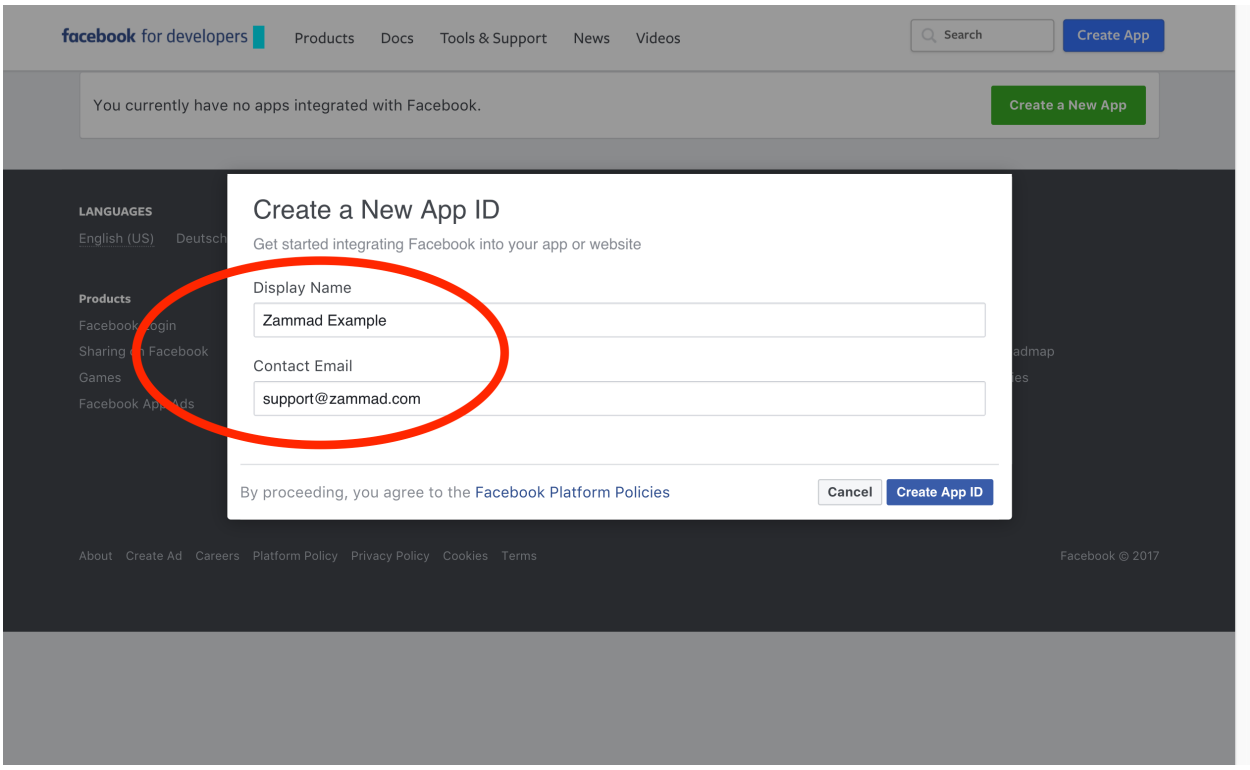
Facebook

Hinweis: 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

1
0
06 08 10 12 14 16 18 20 22 24 26 28 30
OCT

API Stats

Calls | Errors | Average Request Time

Calls

1
0
06 08 10 12 14 16 18 20 22 24 26 28 30
OCT

Application Level Rate Limiting

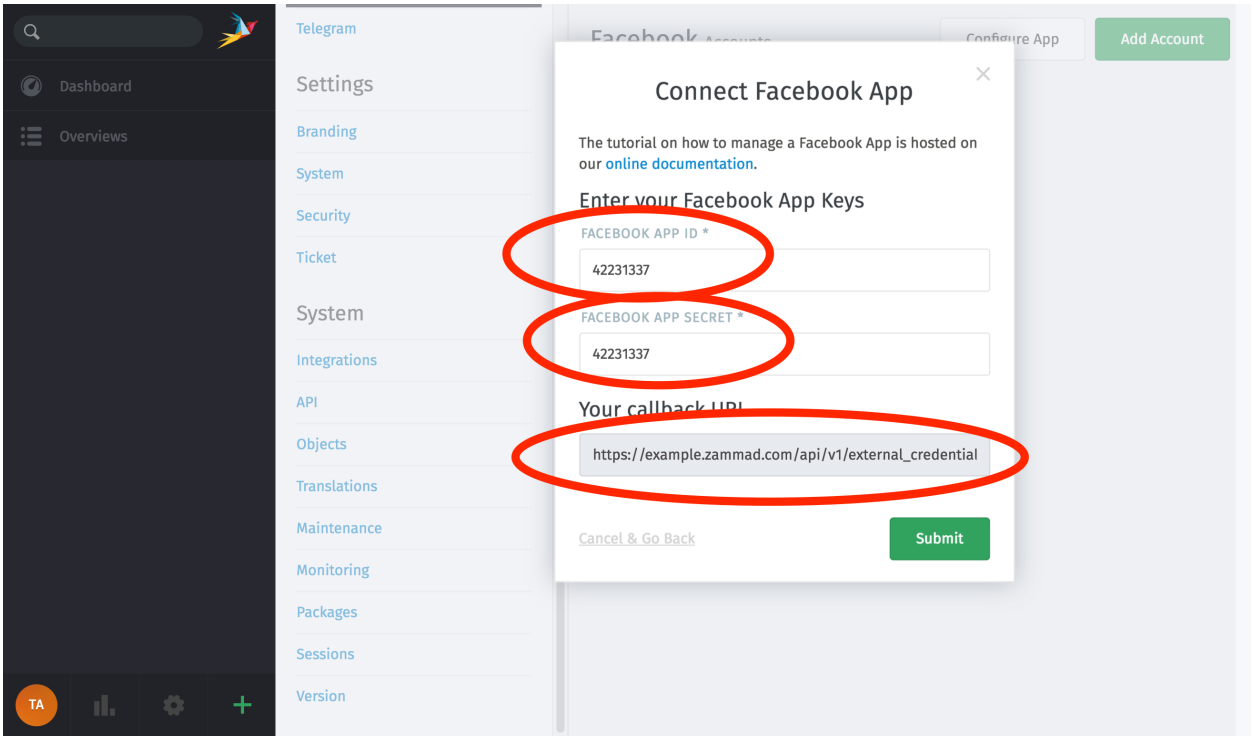
Last 24 Hours

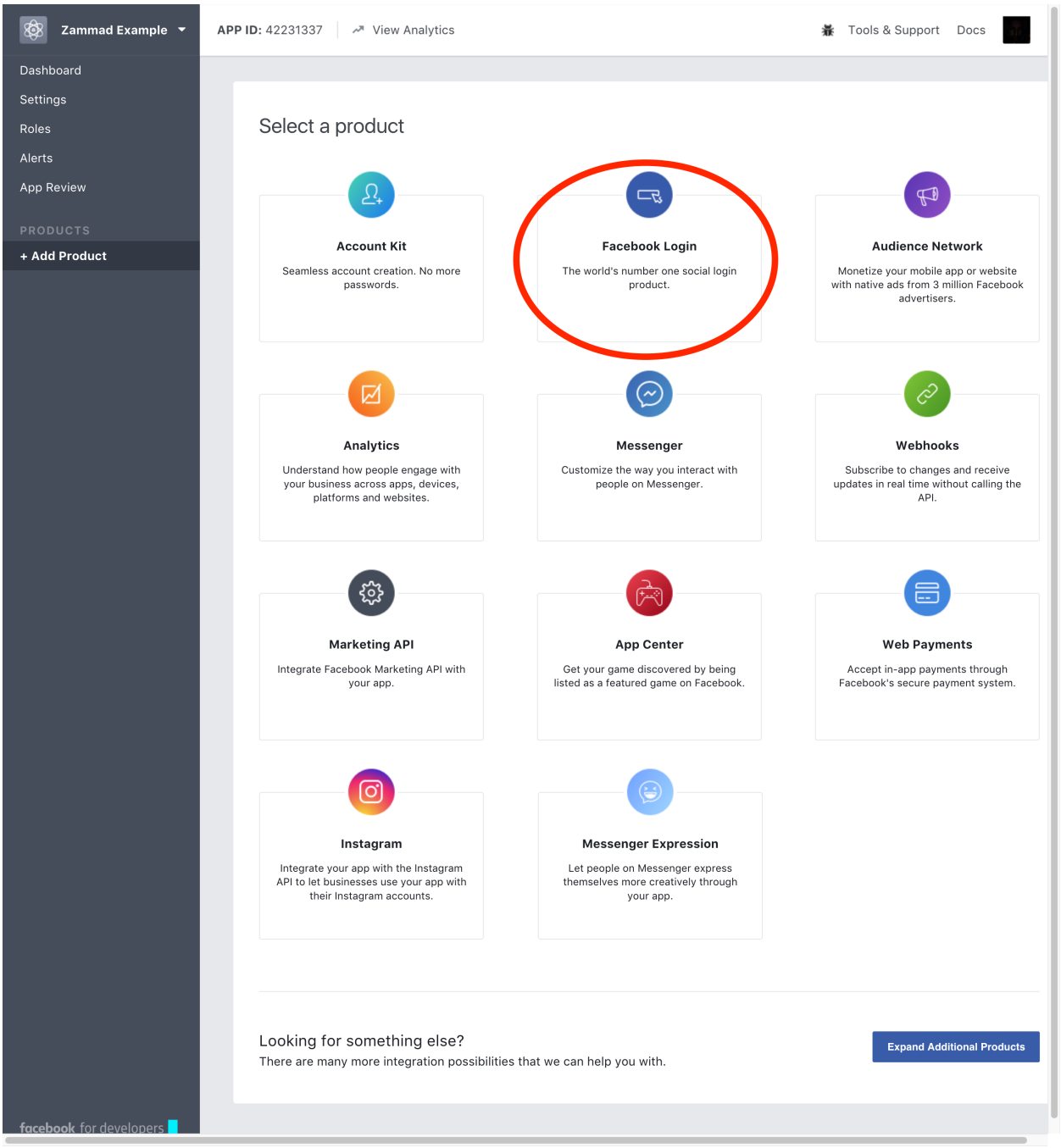
API Utilization
Rate Limit
Approaching Limit

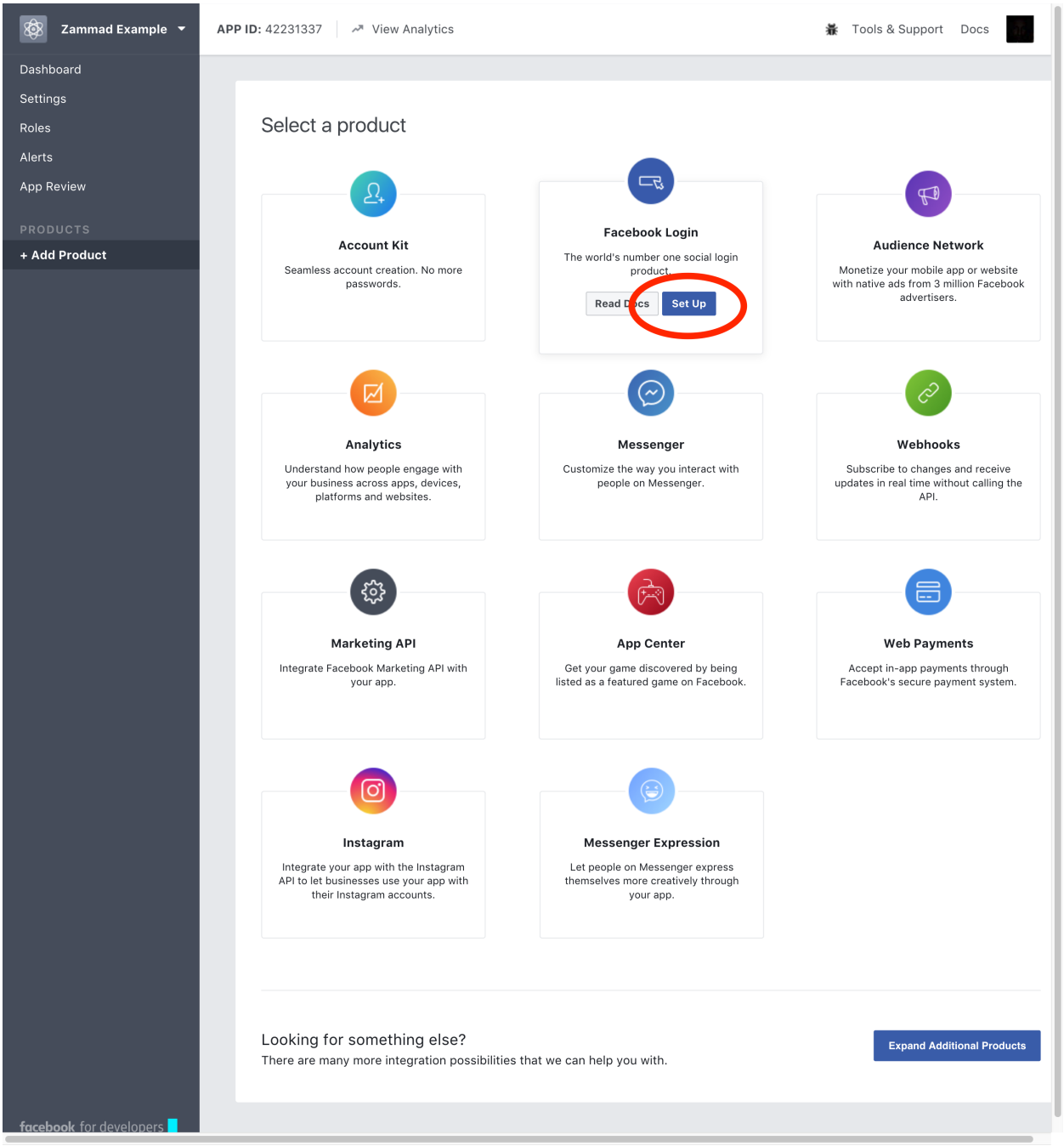
100%

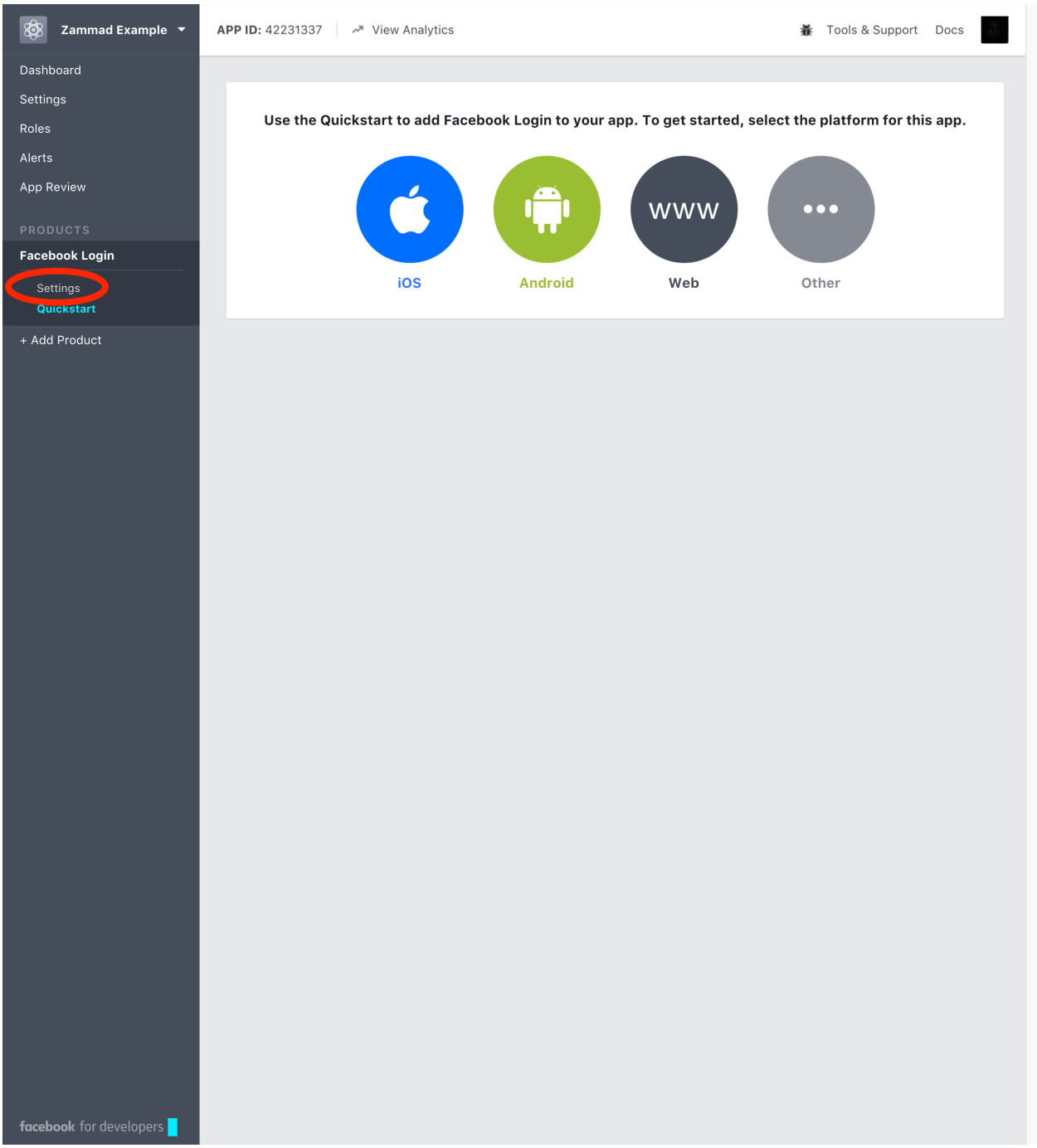
Your app has not made enough calls in last 24 hours to qualify for Rate Limiting

0%
Tue 2 PM Tue 5 PM Tue 9 PM Wed 12 AM Wed 4 AM Wed 7 AM Wed 11 AM Wed 2 PM


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
WWW
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

Yes — 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. [?]

Yes — Web OAuth Login
Enables web based OAuth client login for building custom login flows. [?]

No — Force Web OAuth Reauthentication
When on, prompts people to enter their Facebook password in order to log in on the web. [?]

No — Embedded Browser OAuth Login
Enables browser control redirect uri for OAuth client login. [?]

Yes — 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

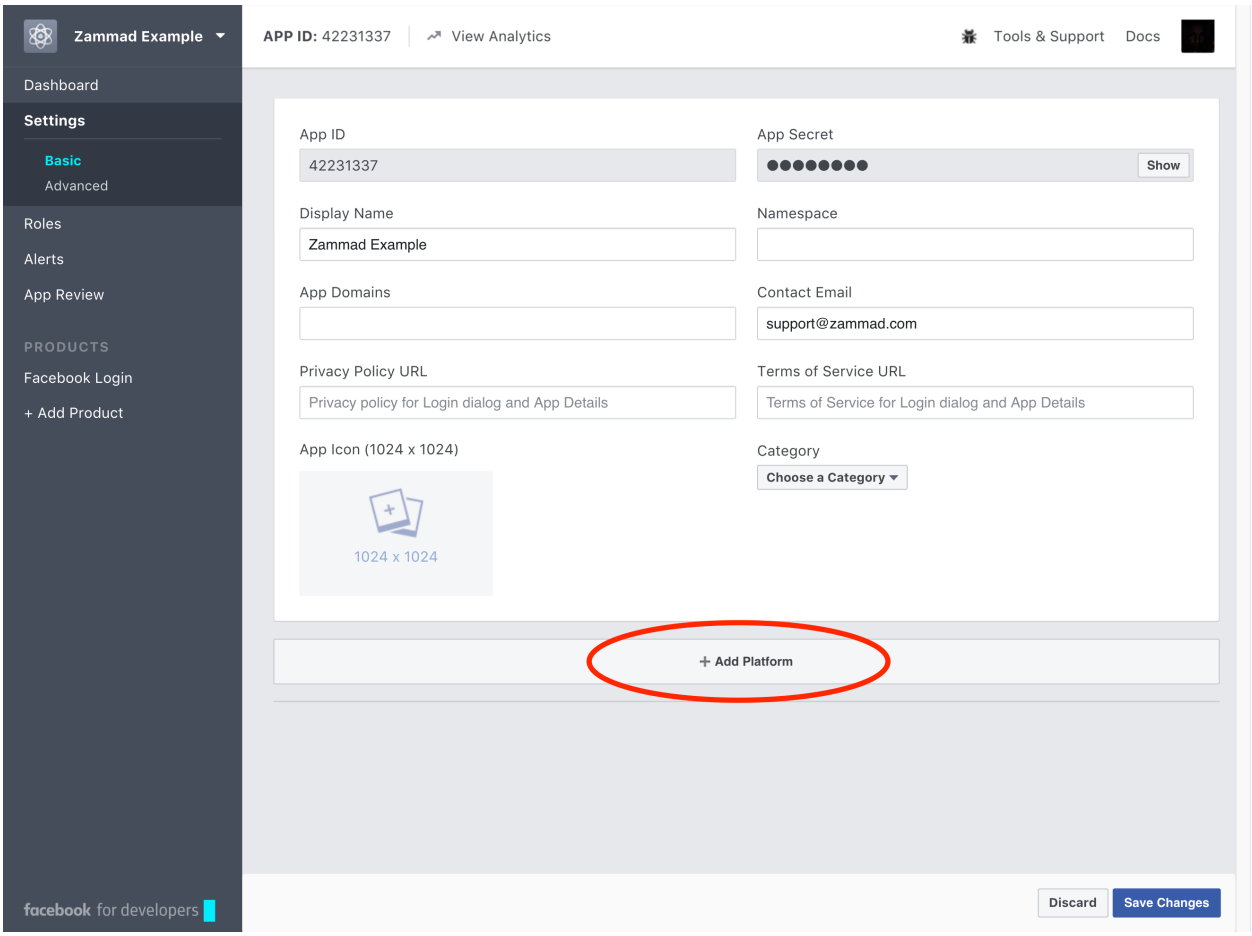
https://example.zammad.com/api/v1/external_credentials/facebook/callback

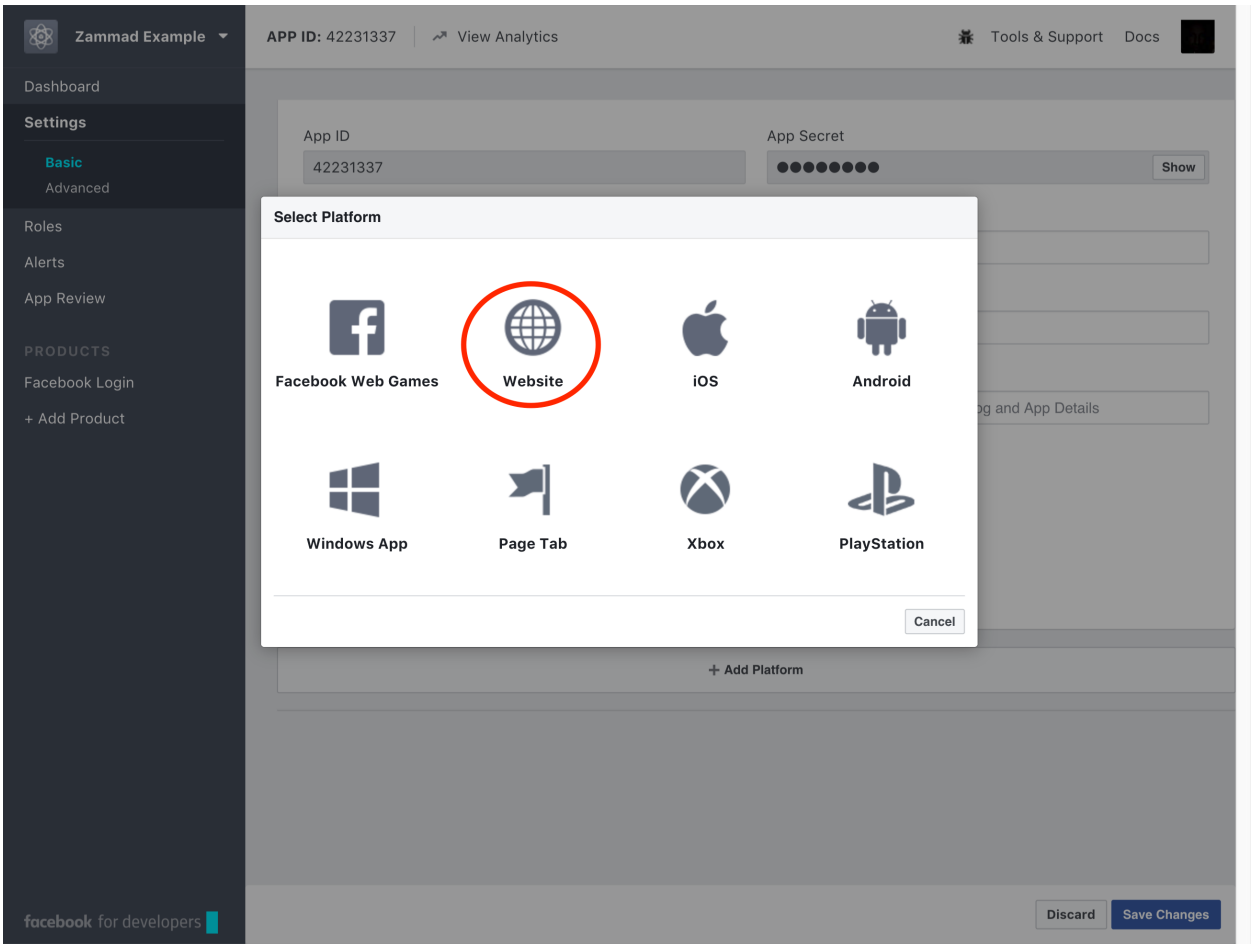
No — Login from Devices
Enables the OAuth client login flow for devices like a smart TV [?]

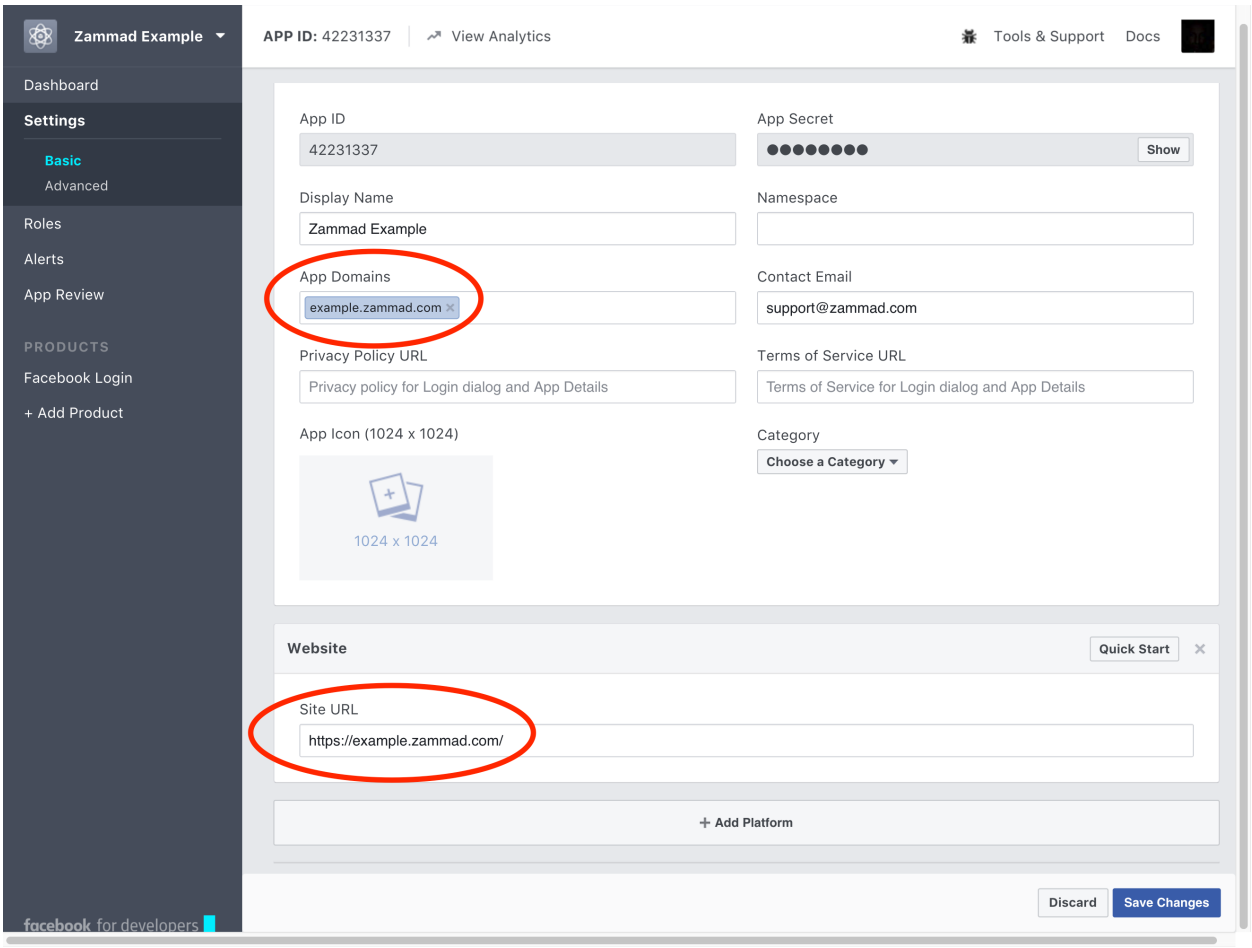
Deauthorize

Deauthorize Callback URL

What should we ping when a user deauthorizes your app?


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

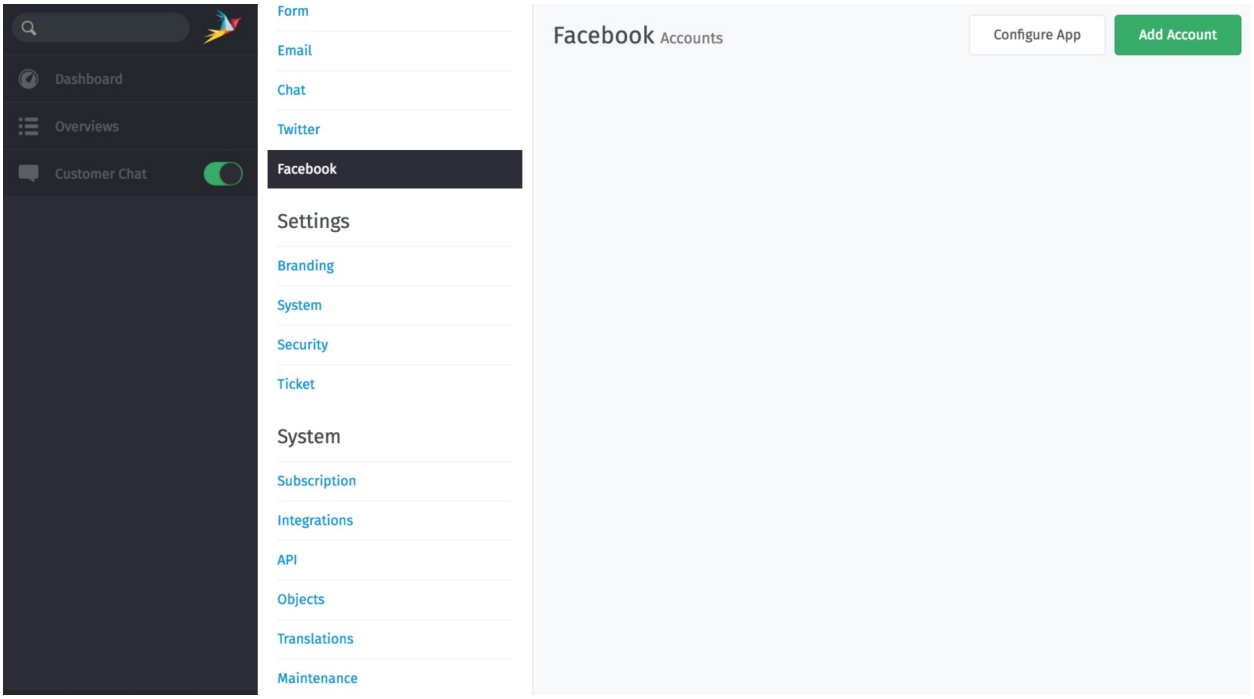
- Go to “Admin -> Channels -> Facebook”
- Click on “Connect Facebook App” and enter your “App ID”, “App Secret” and verify the “Callback URL”.

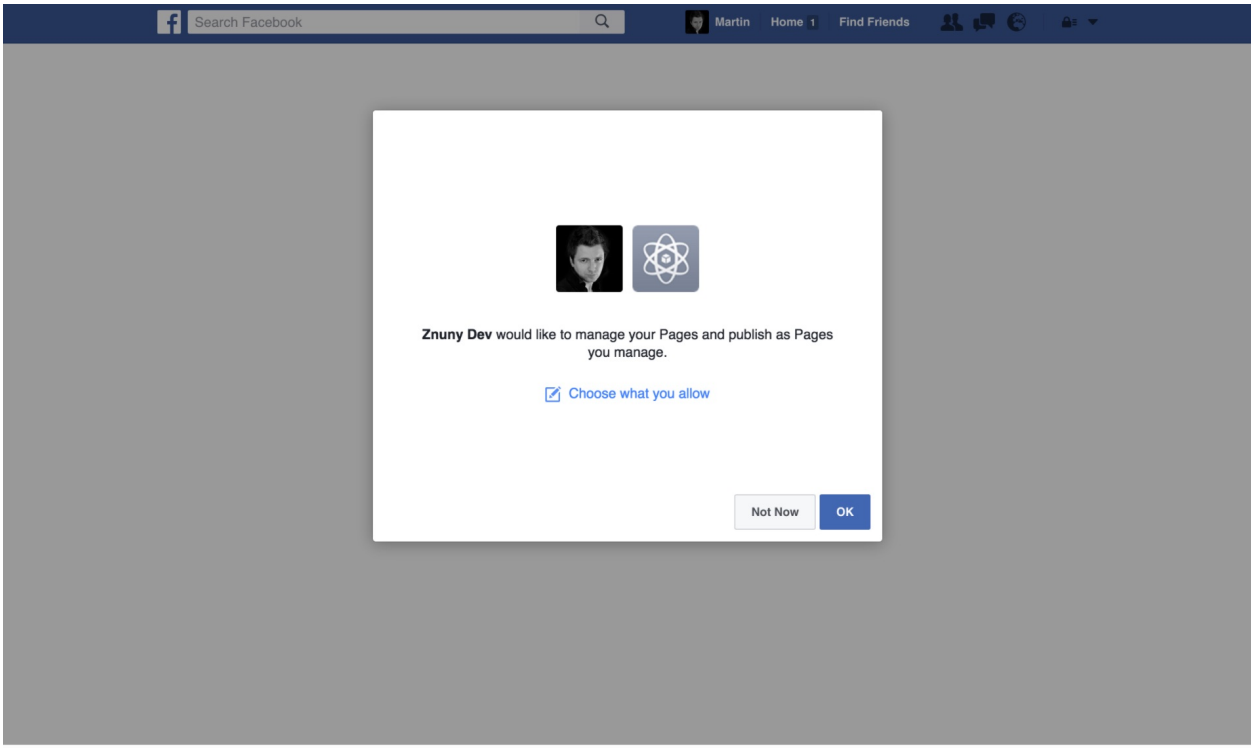
Done, your Zammad is configured as Facebook App now.

Link your Facebook Page to your Zammad Facebook app

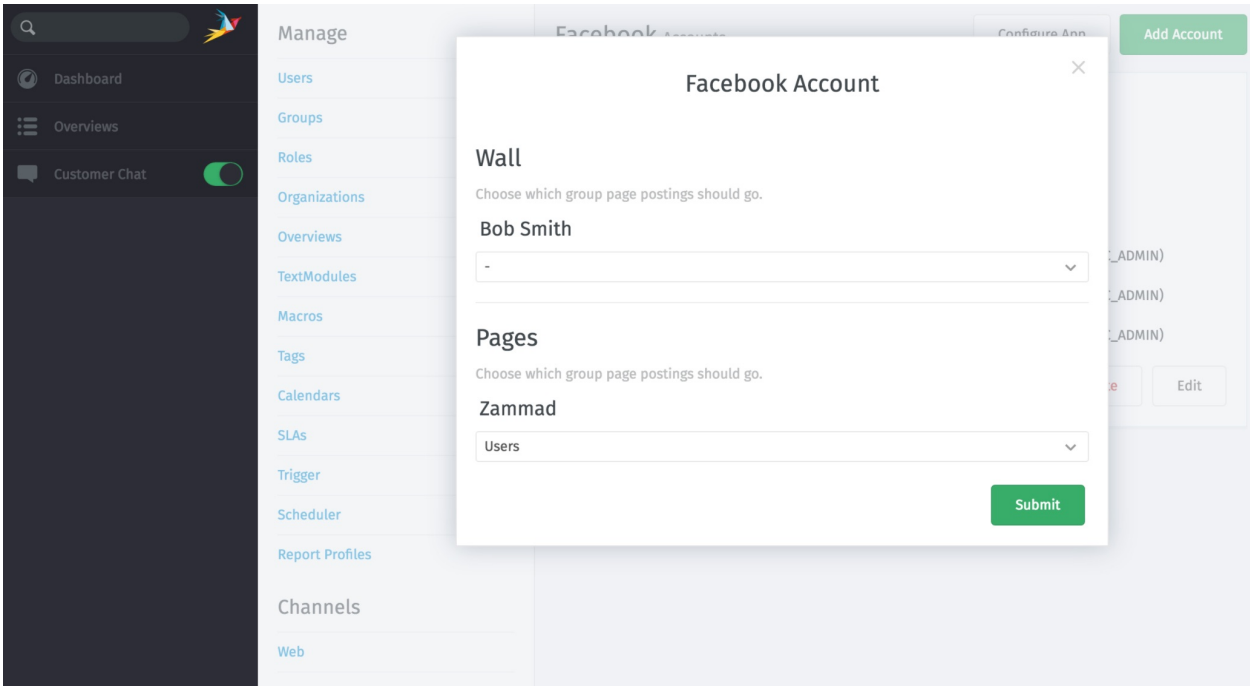
Now you need to link your Facebook Page from which you want to get posts and send out comments.

Click on “Add Account”, then you will see the authorize app page of Facebook. Click on “authorize app”.

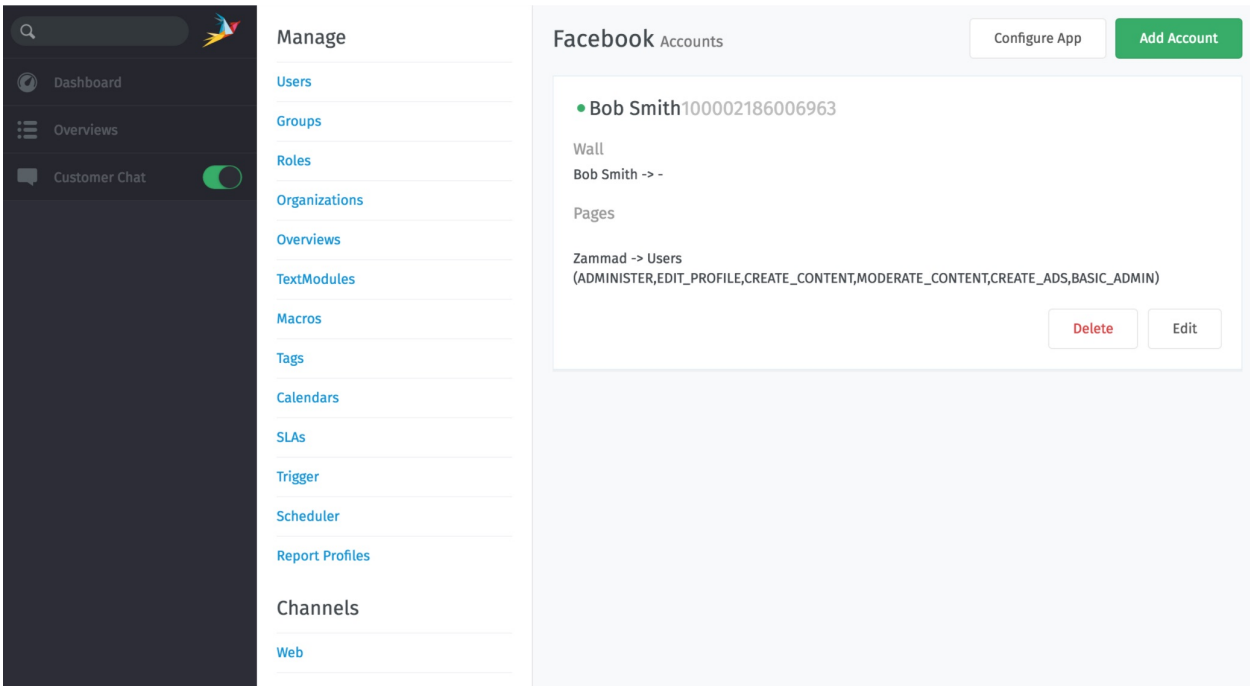




You will get redirected back to Zammad. Now you need to configure your search keys, where mentions should get routed.



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.

KAPITEL 24

Telegram

It's possible to put your Telegram bot communication into Zammad. To do so, you need to follow these steps.

Bemerkung: Your Zammad instance needs to be publicly available via HTTPS (we use Telegram WebHooks).

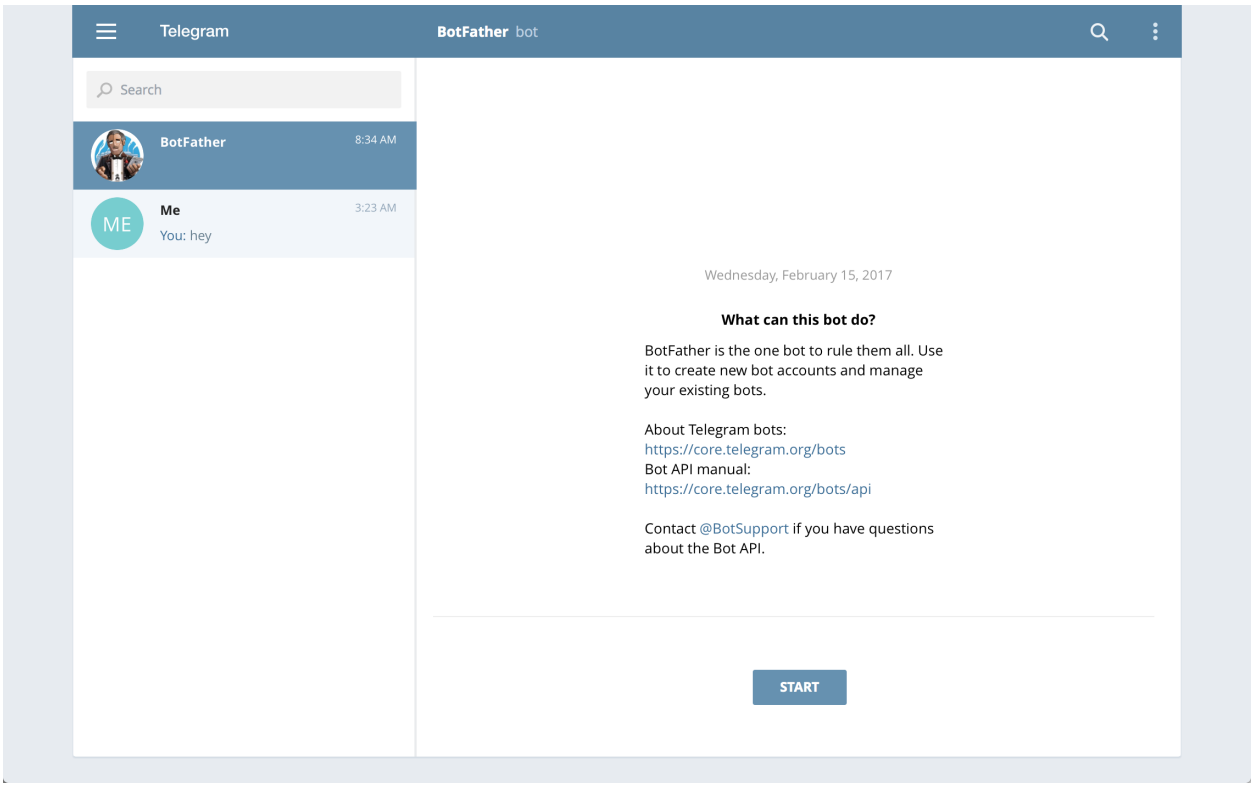
Warnung: Zammad **cannot receive file attachments larger than 20MB** in Telegram messages.

This is a hard limit imposed by the Telegram Bot API.

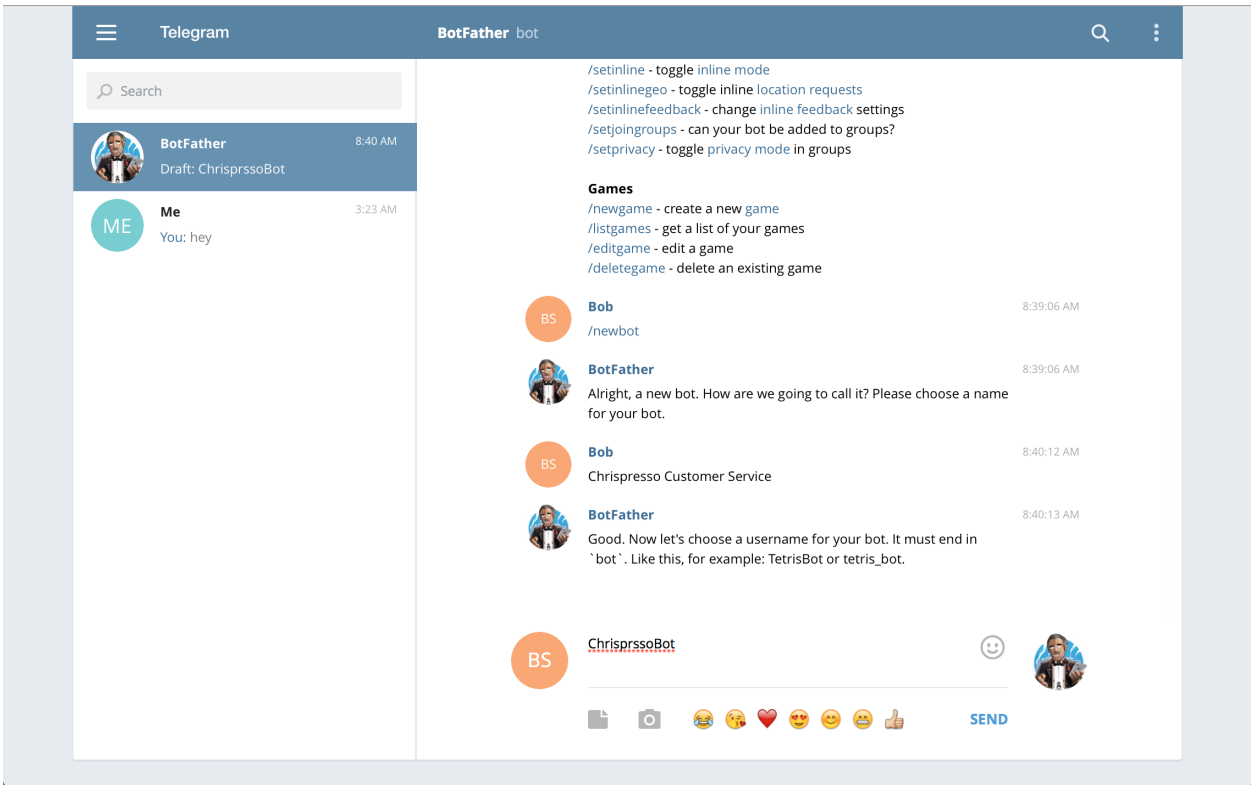
24.1 Register a Telegram bot app

Register your Telegram bot via your Telegram client, see also here: https://core.telegram.org/bots#3-how-do-i-create-a-bot

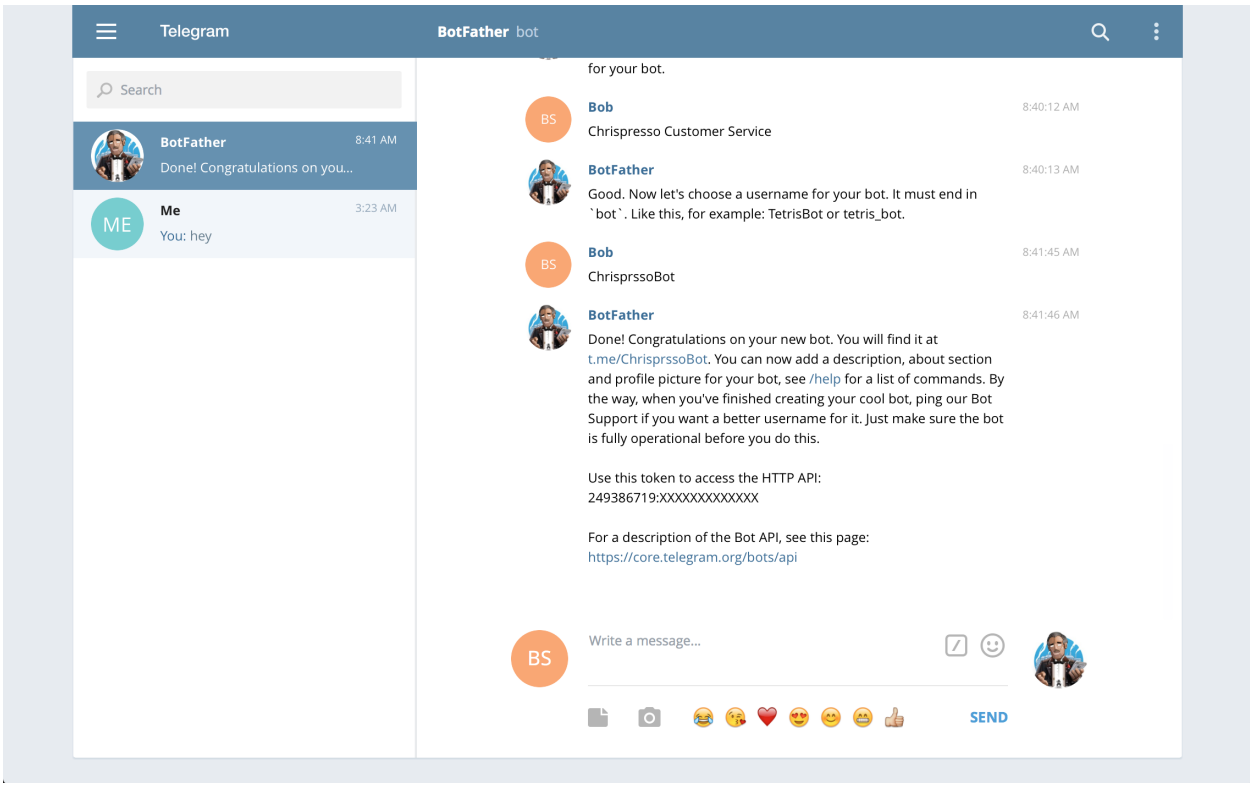
Go to BotFather



Register a new bot by using „/newbot“ and define its name and username

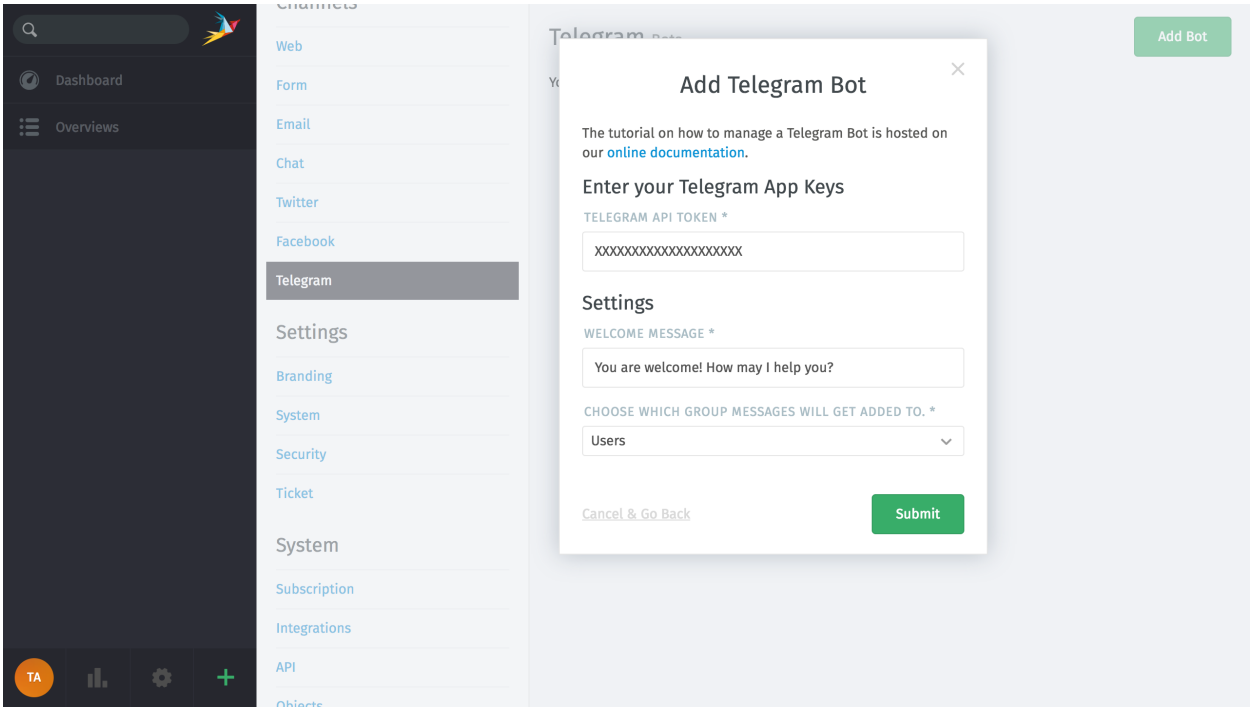


When you’re all done, you will get your Telegram bot API token

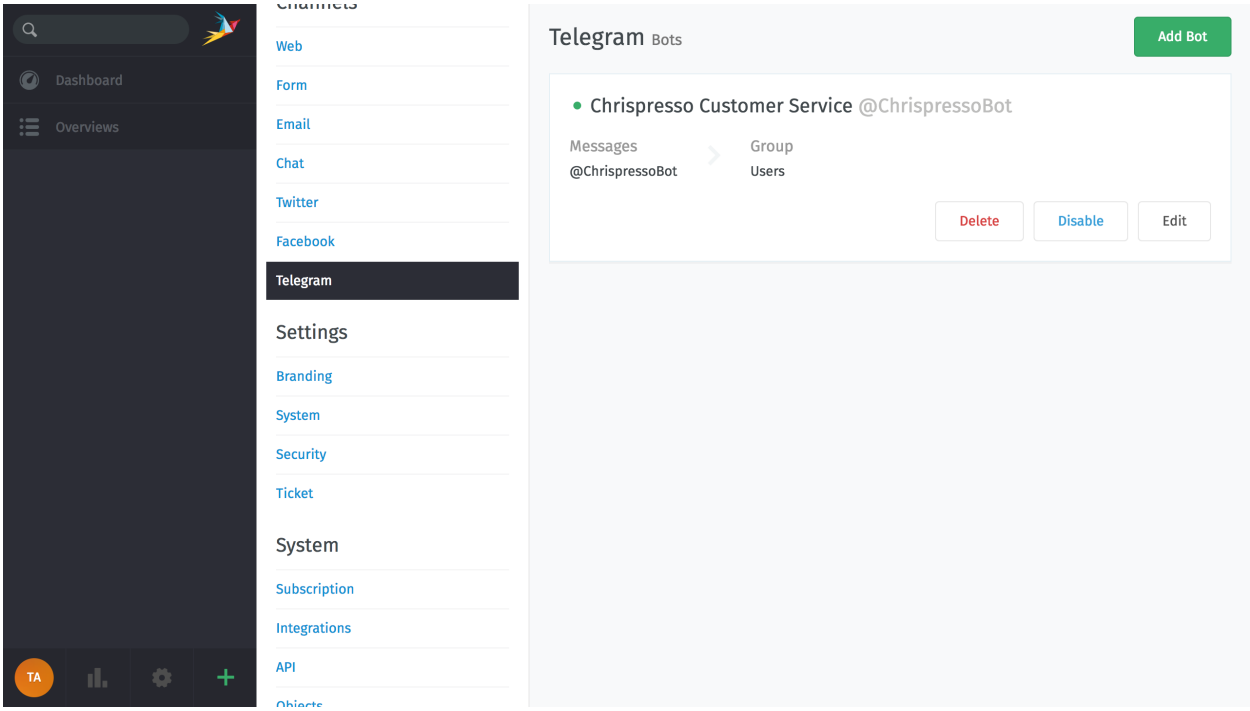


24.2 Configure Zammad as Telegram bot

Go to „Admin -> Channels -> Telegram“ and click „Add Bot“



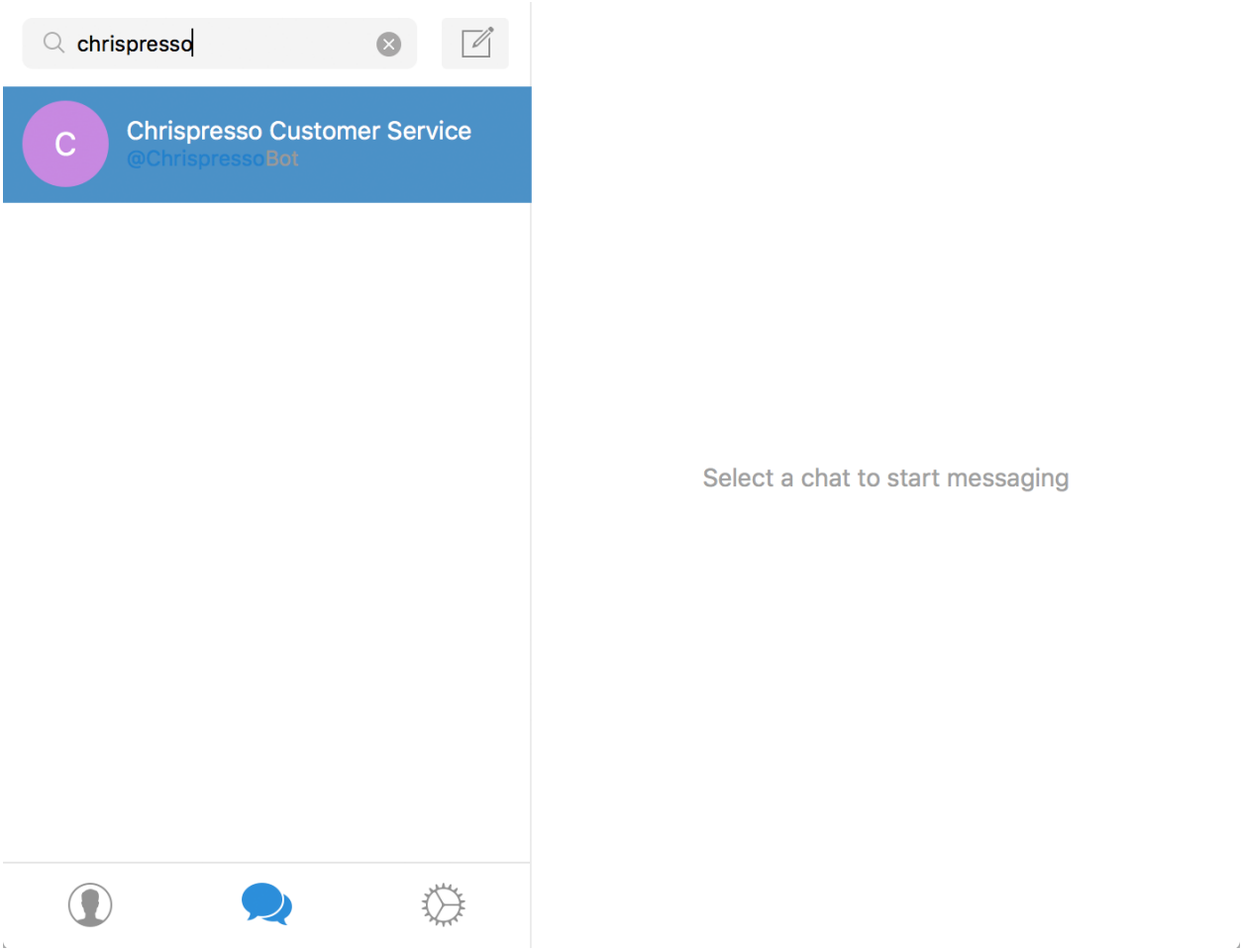
Enter your „API Token“, your „welcome message“ and set the incoming group.

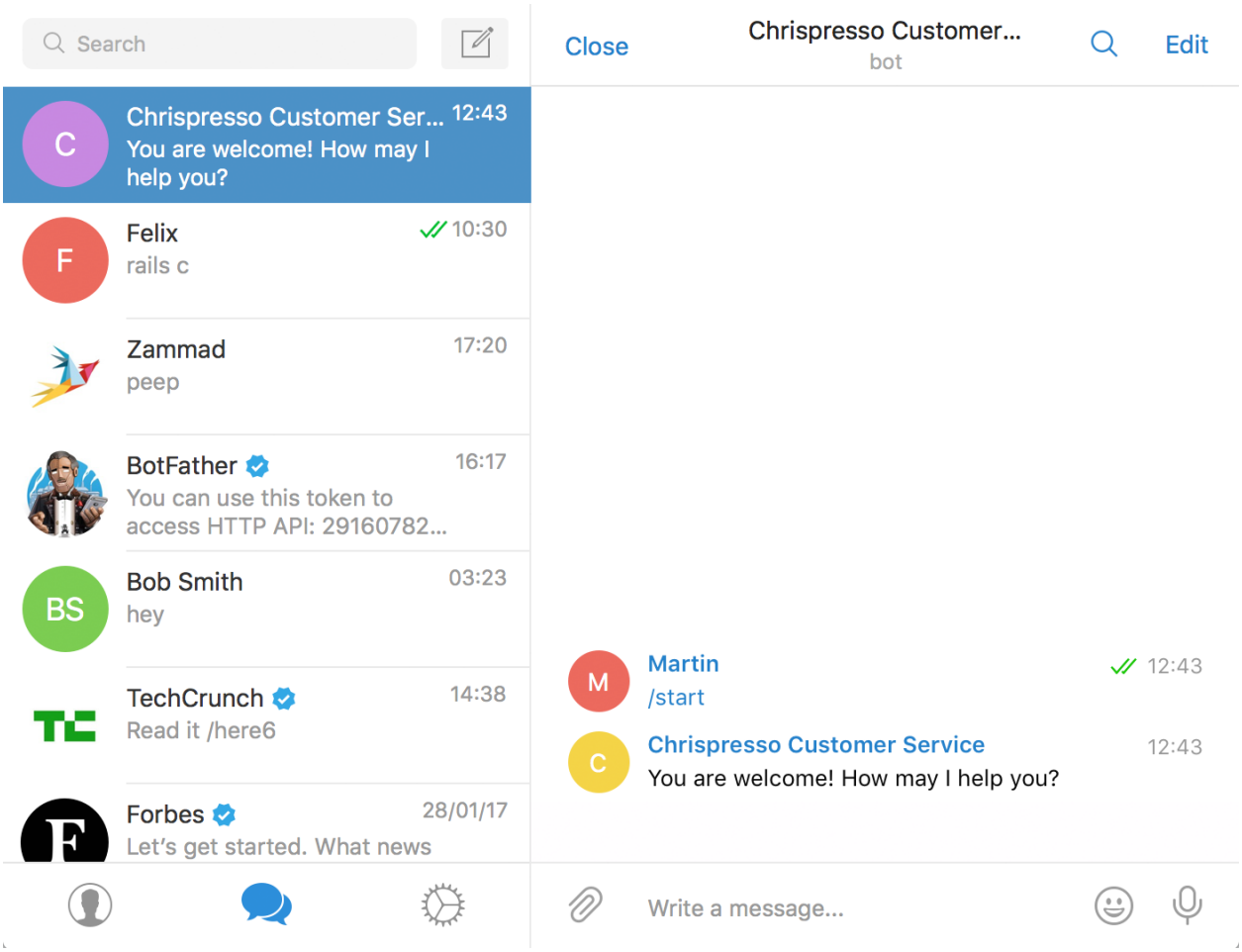


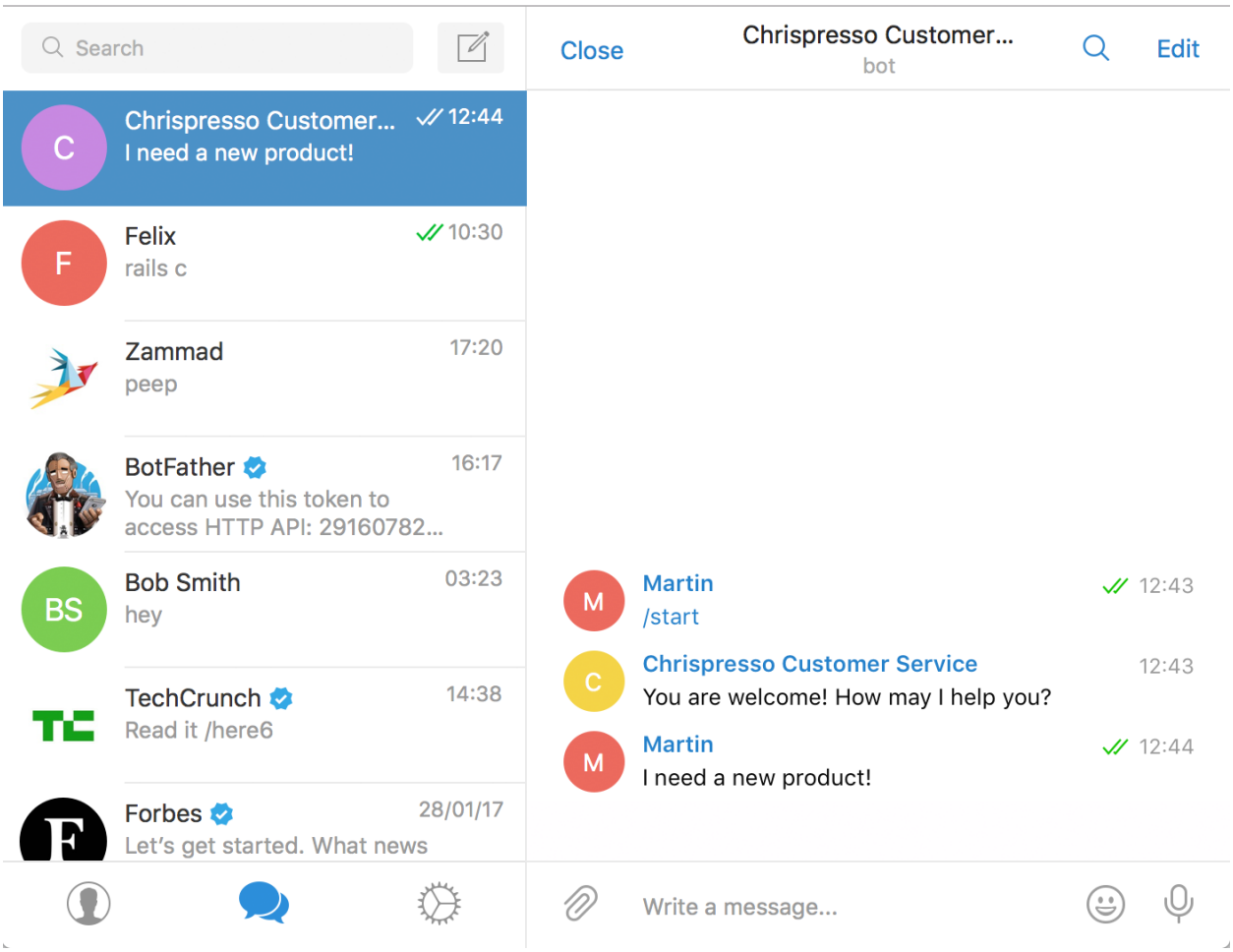
Done, your Zammad is now configured as a Telegram bot.

24.3 Start using your new channel

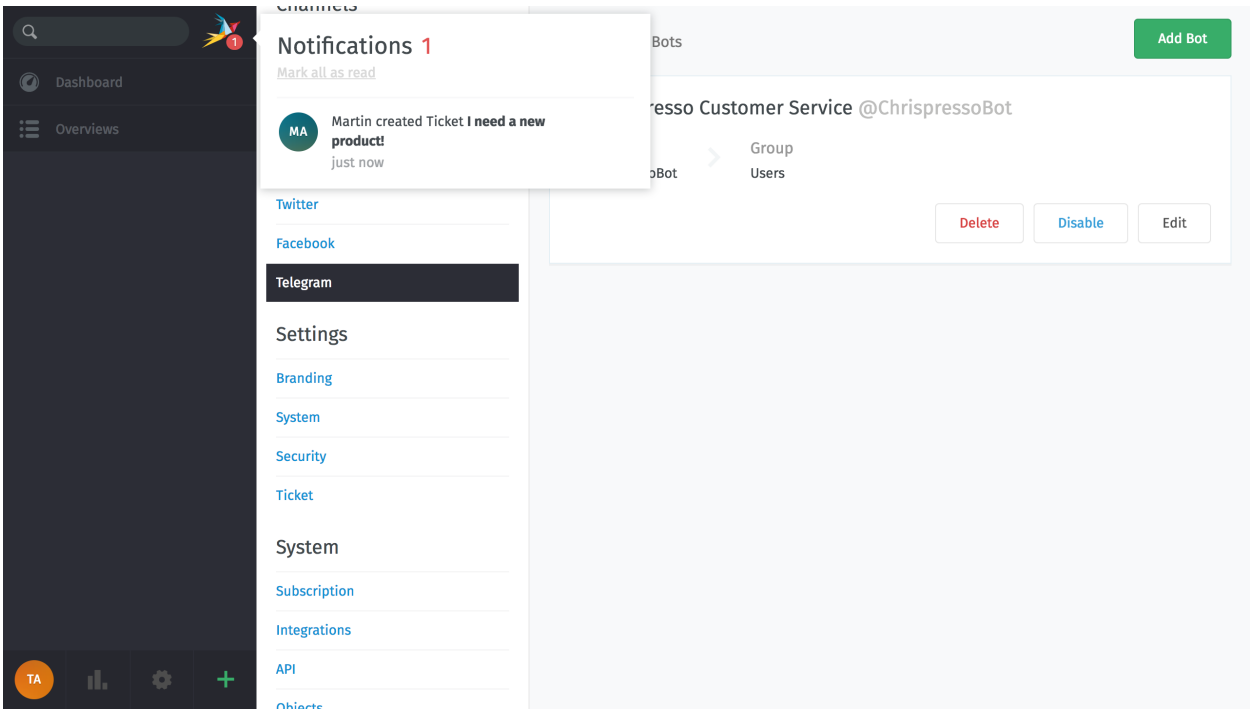
Go to your Telegram client, search for your new Telegram bot and start writing a message.


chrispresso
C
Chrispresso Customer Service
@ChrispressoBot
Select a chat to start messaging

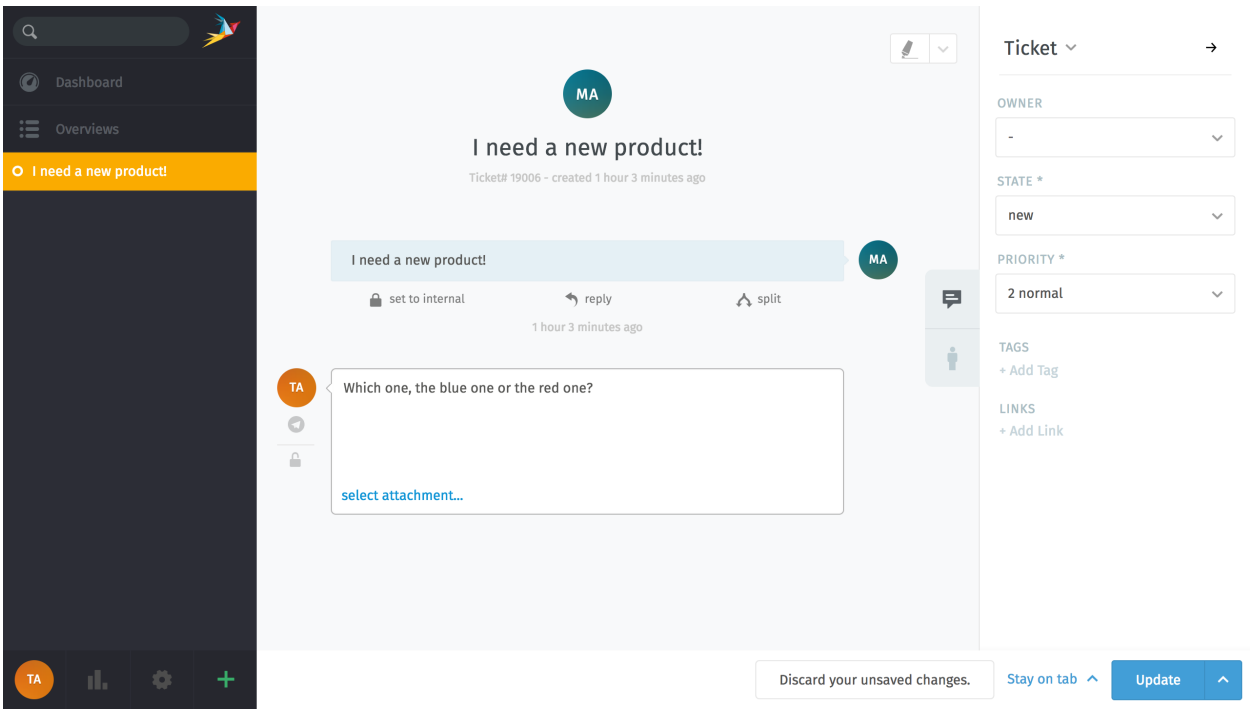

Search
Close
Chrispresso Customer...
bot
Edit
Chrispresso Customer Ser... 12:43
You are welcome! How may I help you?
Felix 10:30
rails c
Zammad 17:20
peep
BotFather 16:17
You can use this token to access HTTP API: 29160782...
Bob Smith 03:23
hey
TechCrunch 14:38
Read it /here6
Forbes 28/01/17
Let's get started. What news
Martin 12:43
/start
Chrispresso Customer Service 12:43
You are welcome! How may I help you?
Write a message...



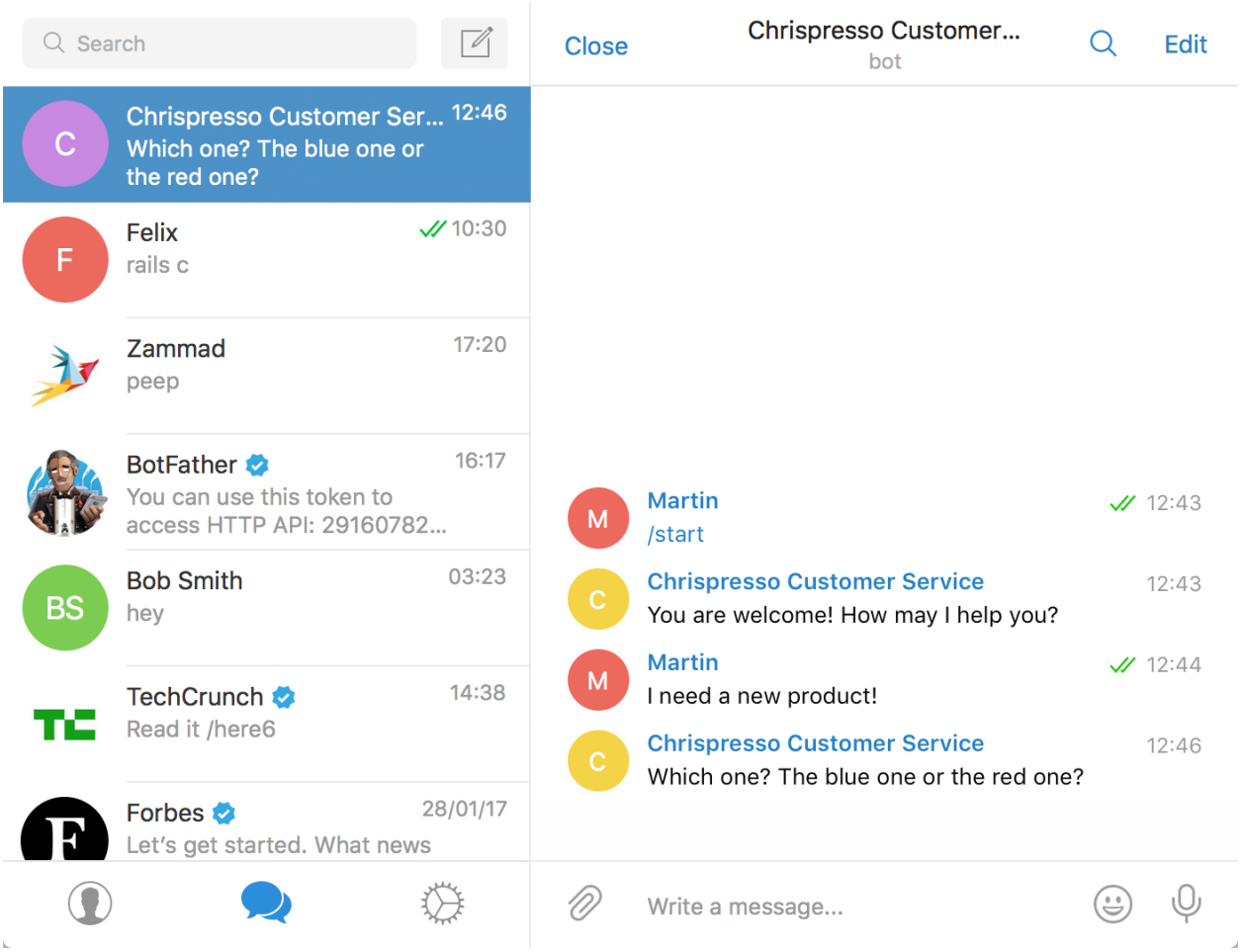
After a few seconds a new message in Zammad appears.



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...

KAPITEL 25

Branding

Product-name Defines the name of the application, shown in the web interface, tabs and title bar of the web browser.

Organisation Will be shown in the app and is included in email footers.

Logo Defines the logo of the application, shown in the web interface.

Pretty Date Defines date format you like the most.

KAPITEL 26

System

26.1 Service

Image Service Defines the backend for user and organization image lookups.

Geo Calendar Service Defines the backend for geo calendar lookups. Used for initial calendar succession.

Geo IP Service Defines the backend for geo IP lookups. Shows also location of an IP address if an IP address is shown.

Geo Location Service Defines the backend for geo location lookups to store geo locations for addresses.

Hinweis: You can find a detailed privacy information on what we store for how long on our Privacy-Appendix inside of our System-Administrator-Documentation.

26.2 Frontend

Send client stats Send client stats/error message to central server to improve the usability.

Client storage Use client storage to cache data to enhance performance of application.

KAPITEL 27

Security

27.1 Base

New User Accounts Enables users to create their own account via web interface.

Lost Password Activates lost password feature for users. If „no“ is selected here, the password can only be changed via the admin interface.

Hinweis: Normally, especially when using LDAP, you might want to consider to disable both above options, because you'd normally do that via LDAP.

27.2 Password

This section allows you to define password requirements for the local user accounts.

Bemerkung: Zammad does **not** allow you to change your LDAP password, instead, it will set a password in it's local database which might confuse your users. This will be addressed in the future by #1169 and #2389.

27.2.1 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. Please note that via UI the only way to unlock a user account is to change the password (either as admin or via password reset function (if enabled)). The default value is *10*.

Bemerkung: You can also unlock an account via console or API.

27.2.2 2 lower and 2 upper characters

You can add complexity into passwords by enforcing at least 2 upper and lower case characters. The default value is *no*.

27.2.3 Minimum length

This defines the minimum password length required (from *4* to *20*). The default value is *6*.

27.2.4 Digit required

This enforces your users to at least use one digit with his password. The default value is *yes*.

27.3 Third-Party Applications

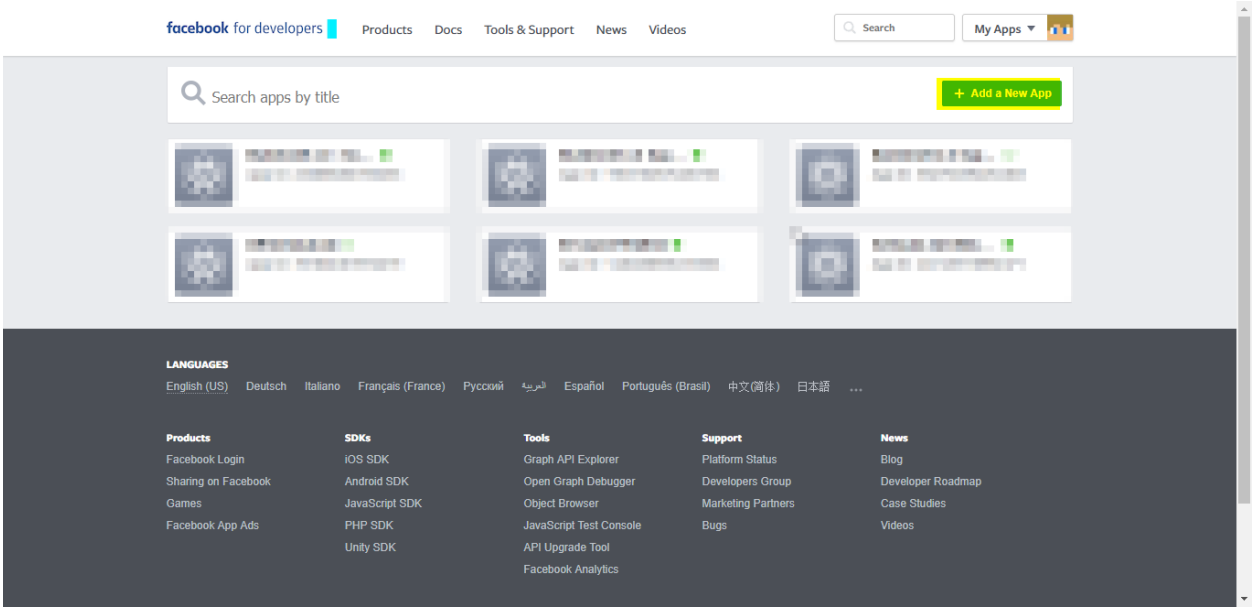
Third party authentication is a great way to help your users to login to Zammad easier. If the account is yet unknown, Zammad will create a new user automatically, without the user needed to interact (e.g. type in his name). Another big advantage of this feature is that your user doesn't need another password to remember.

27.3.1 Facebook

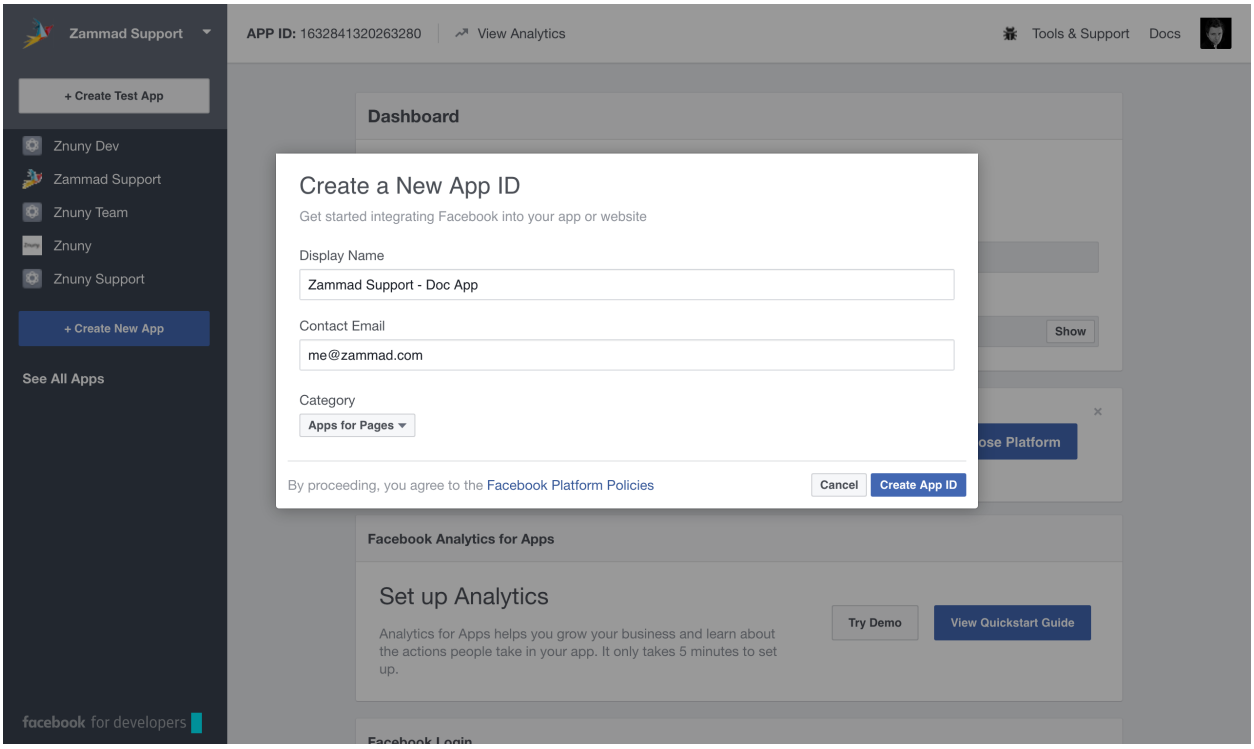
It is possible to create a quick login for your helpdesk via Facebook To do so, you need to follow these steps:

Register Facebook app

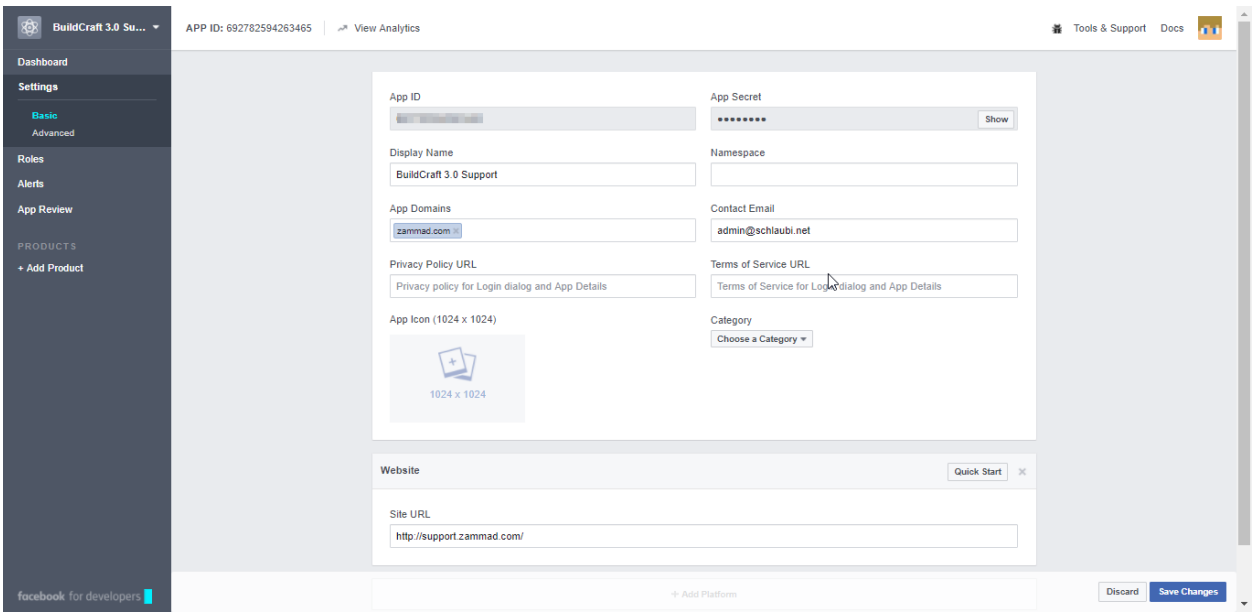
Visit [https://developers.facebook.com/apps/] and click on „Add a new App"



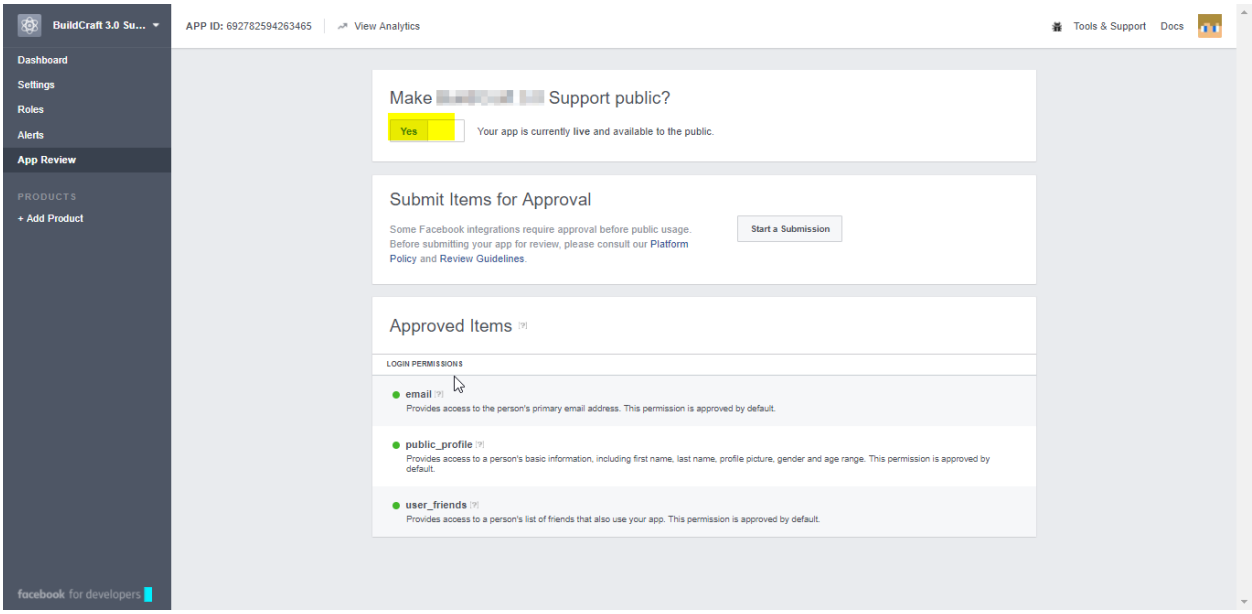
After that enter the app settings



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.

27.3.2 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[redacted]

Client Secret
1363[redacted]

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

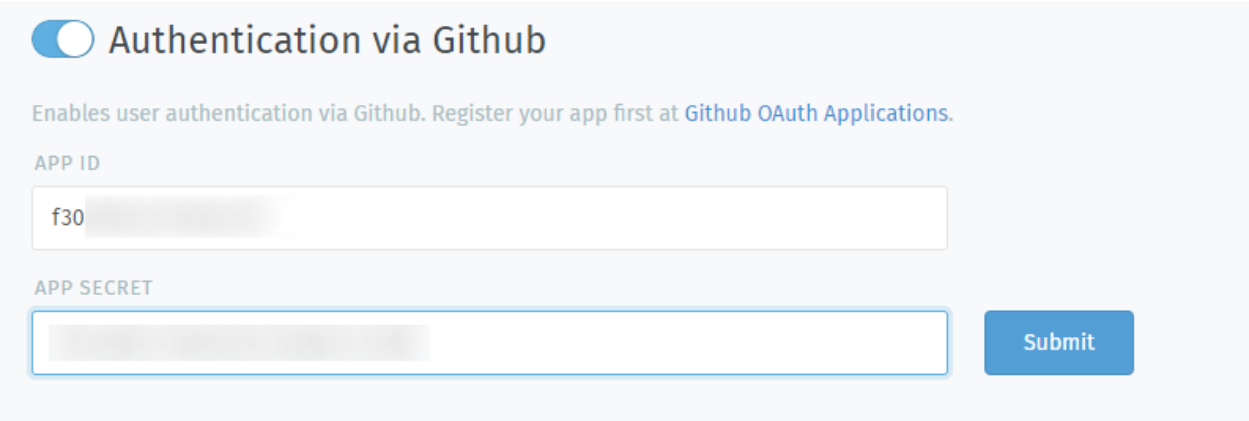
https://myzammad.zammad.com/auth/github/callback

Your application's callback URL. Read our OAuth documentation for more information.

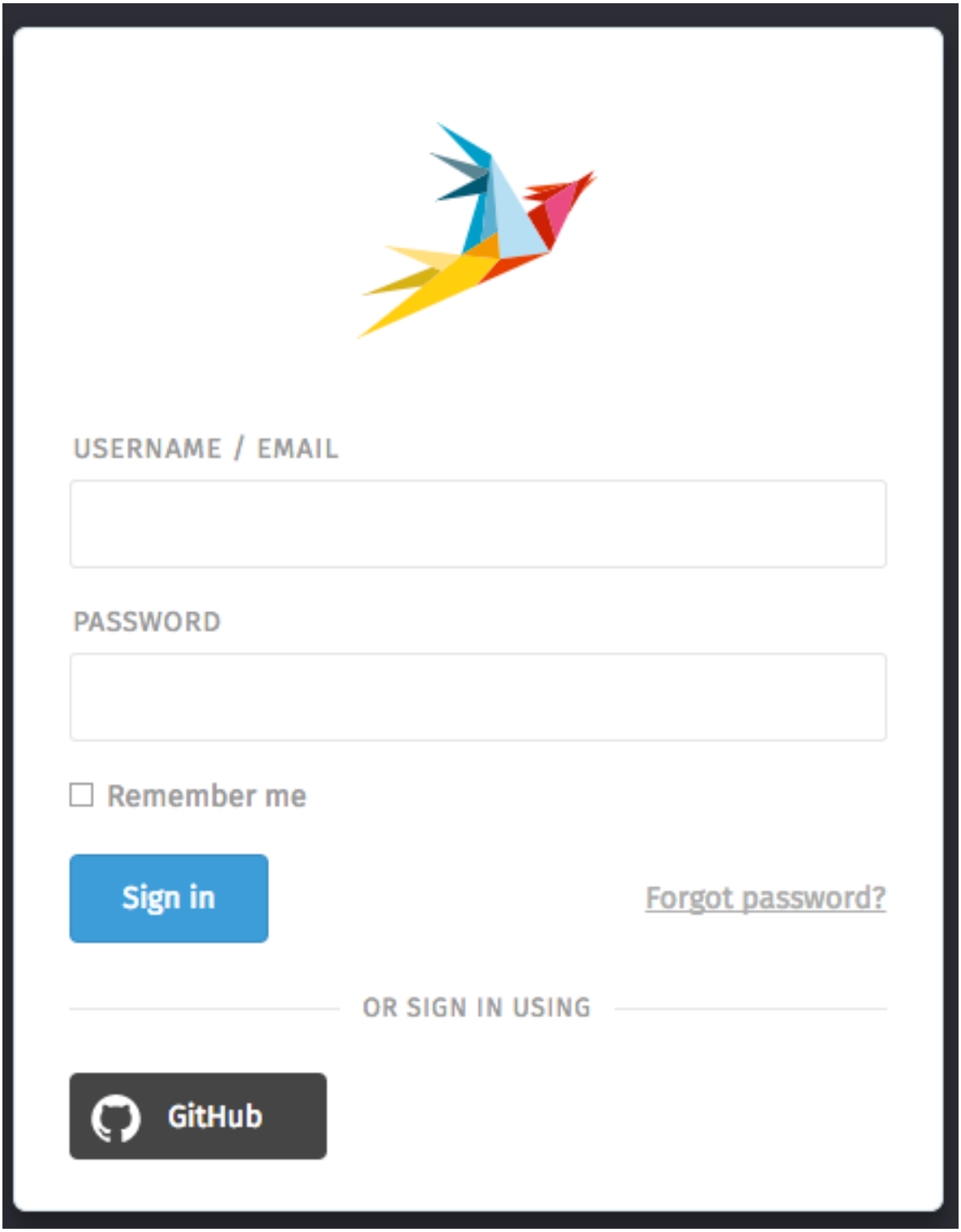
Update application | Delete application

Configure Zammad as GitHub app

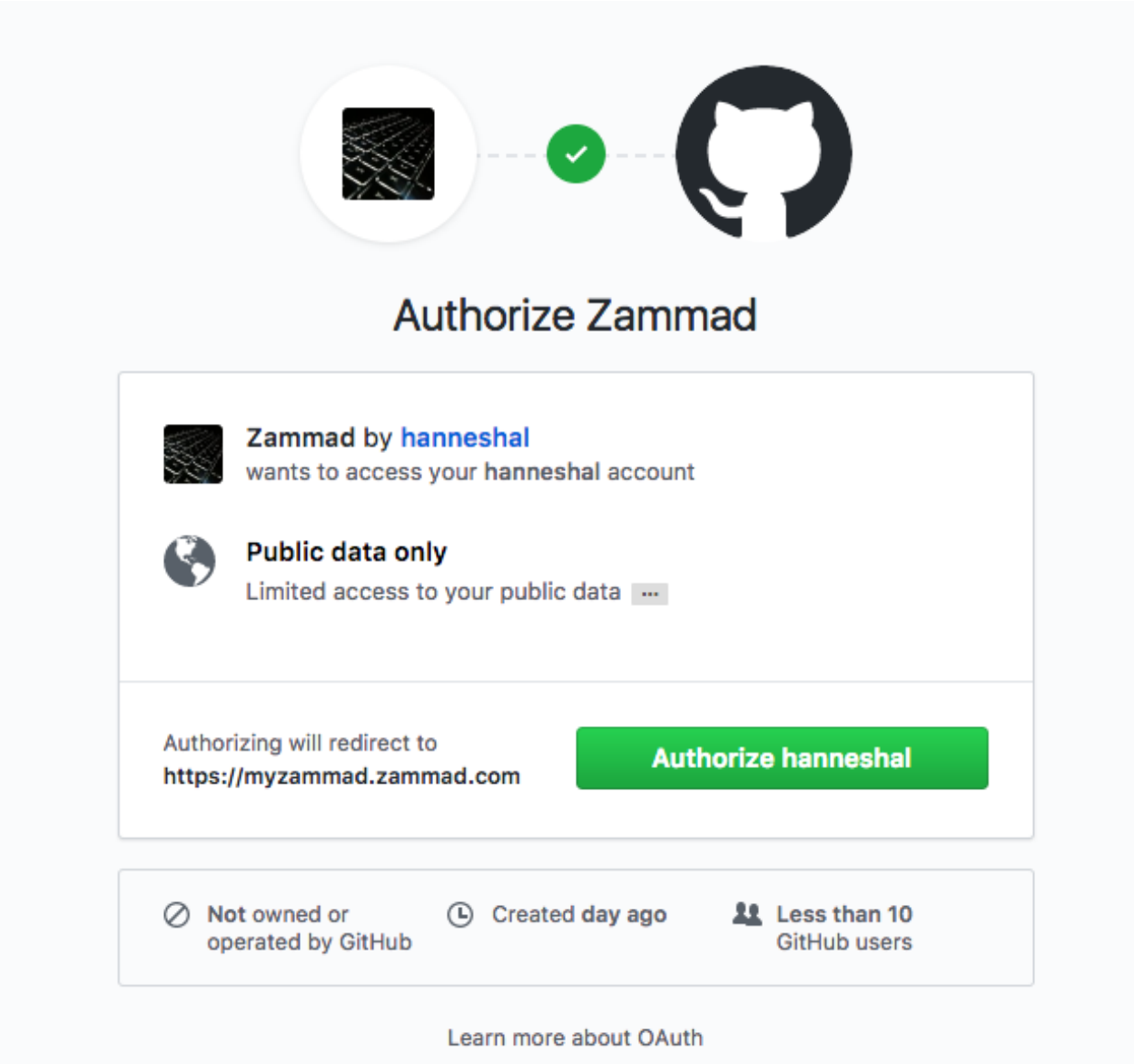
Enter the „APP ID“ and the „APP SECRET“ from the GitHub OAUTH Applications Dashboard



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.

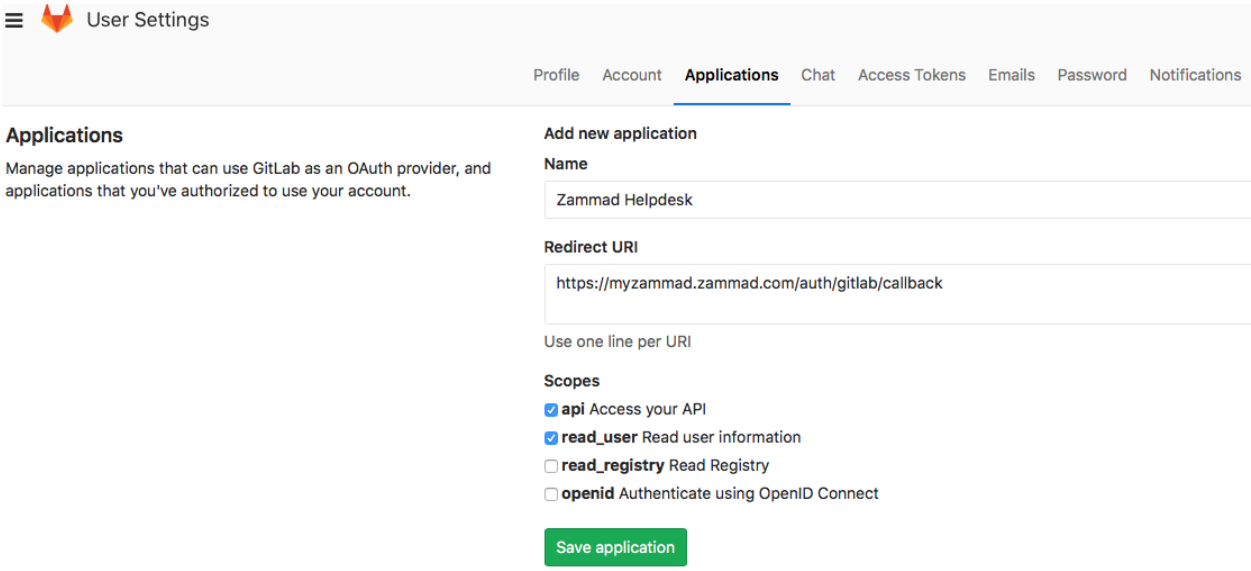
27.3.3 Gitlab

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

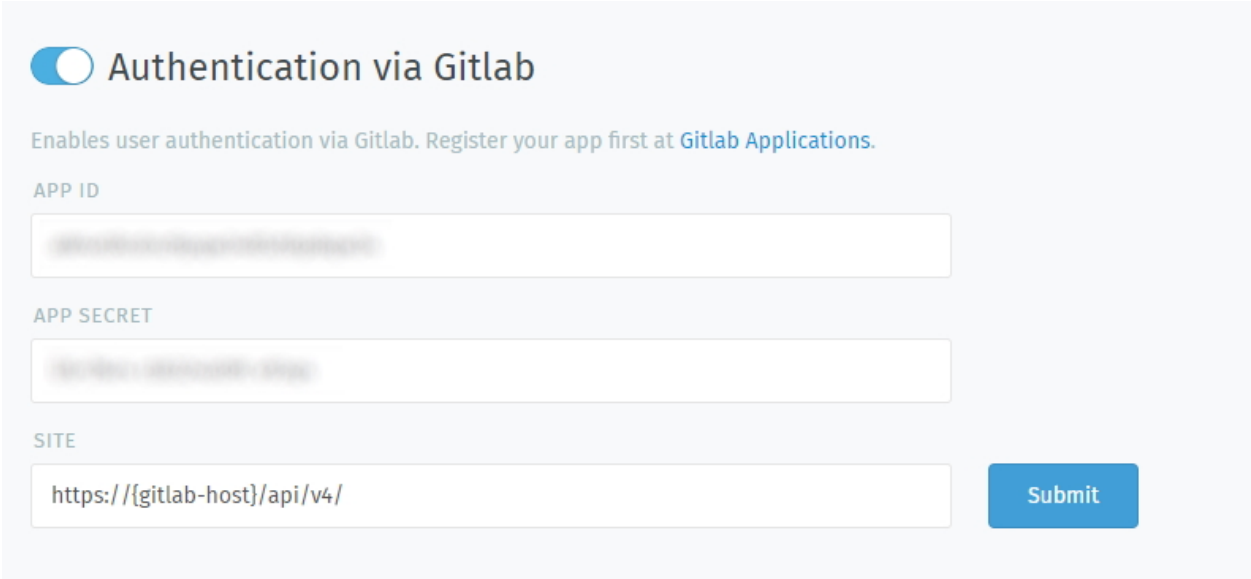


At the moment we need the „api“ scope. This is caused due a bug in Gitlab: https://gitlab.com/gitlab-org/gitlab-ce/issues/33022

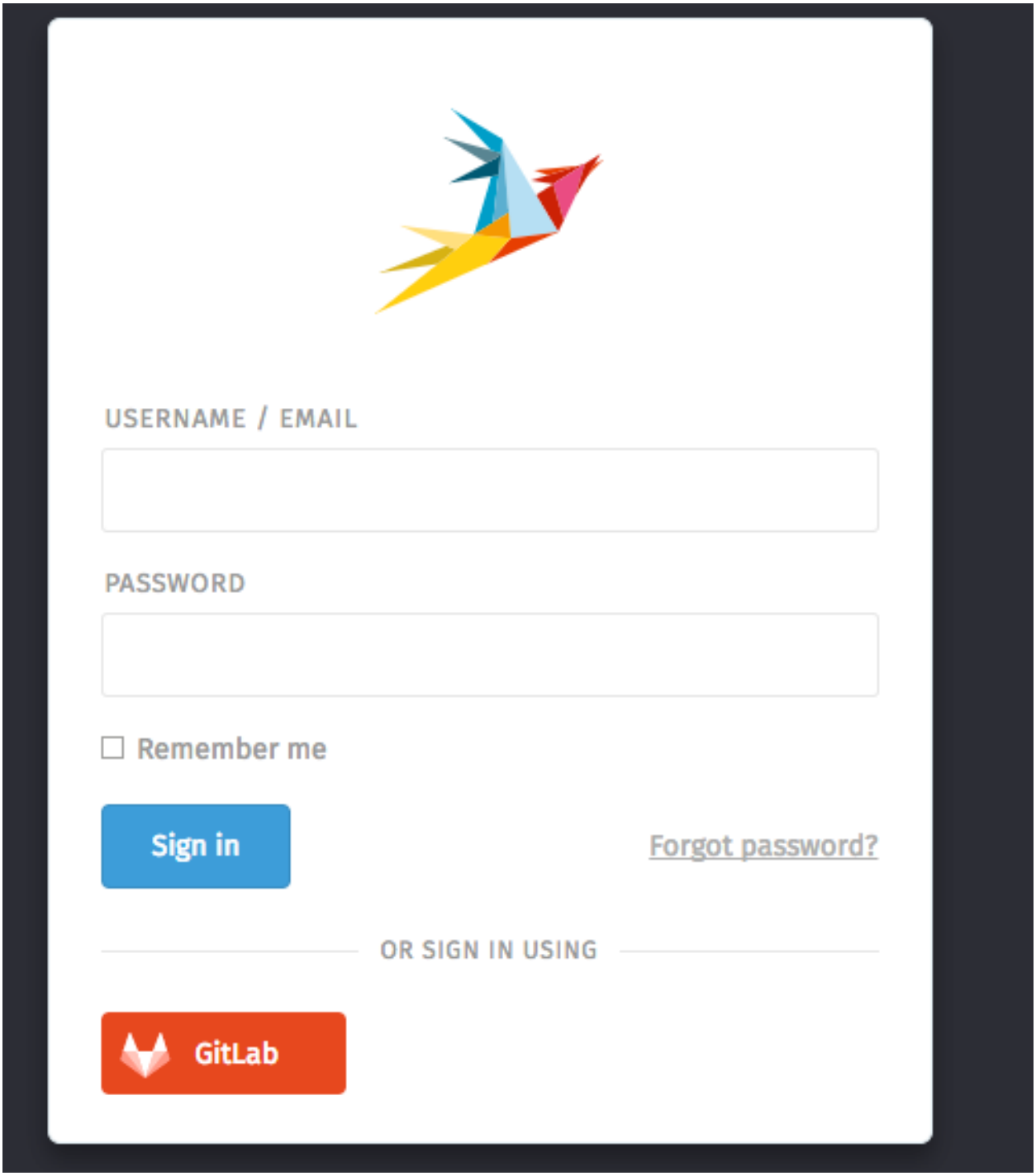
Configure Zammad as Gitlab app

Enter the „APP ID“ and the „APP SECRET“ from the Gitlab OAUTH Applications Dashboard.

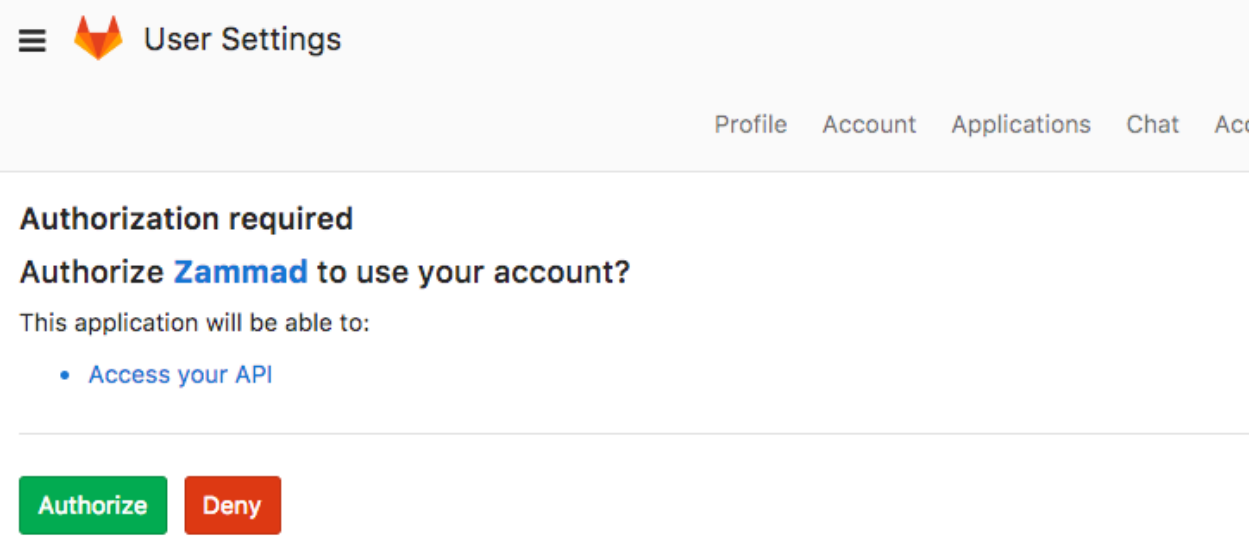
Bemerkung: Please ensure to use `https://{git_host}/api/v4/` for site.



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.

27.3.4 Google

With some easy and fast steps, you can enable Zammad to authenticate your users via Google.

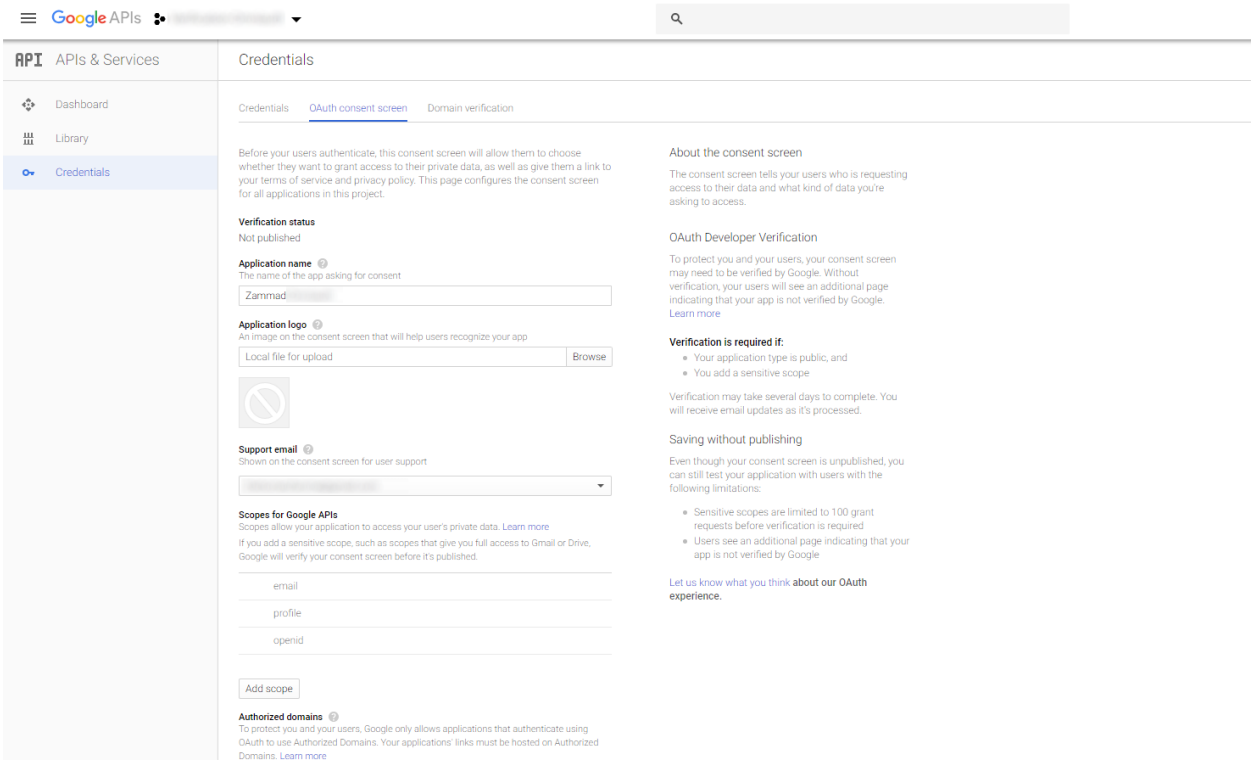
Hinweis: Prior Zammad 2.9 it was necessary to activiate the Google+ API - it's deprecated, thus this authentication will only work with Zammad 2.9 and later.

Register a Google app

First of all, we need to create a new project - you can skip this step if you already have one.

Hinweis: 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!



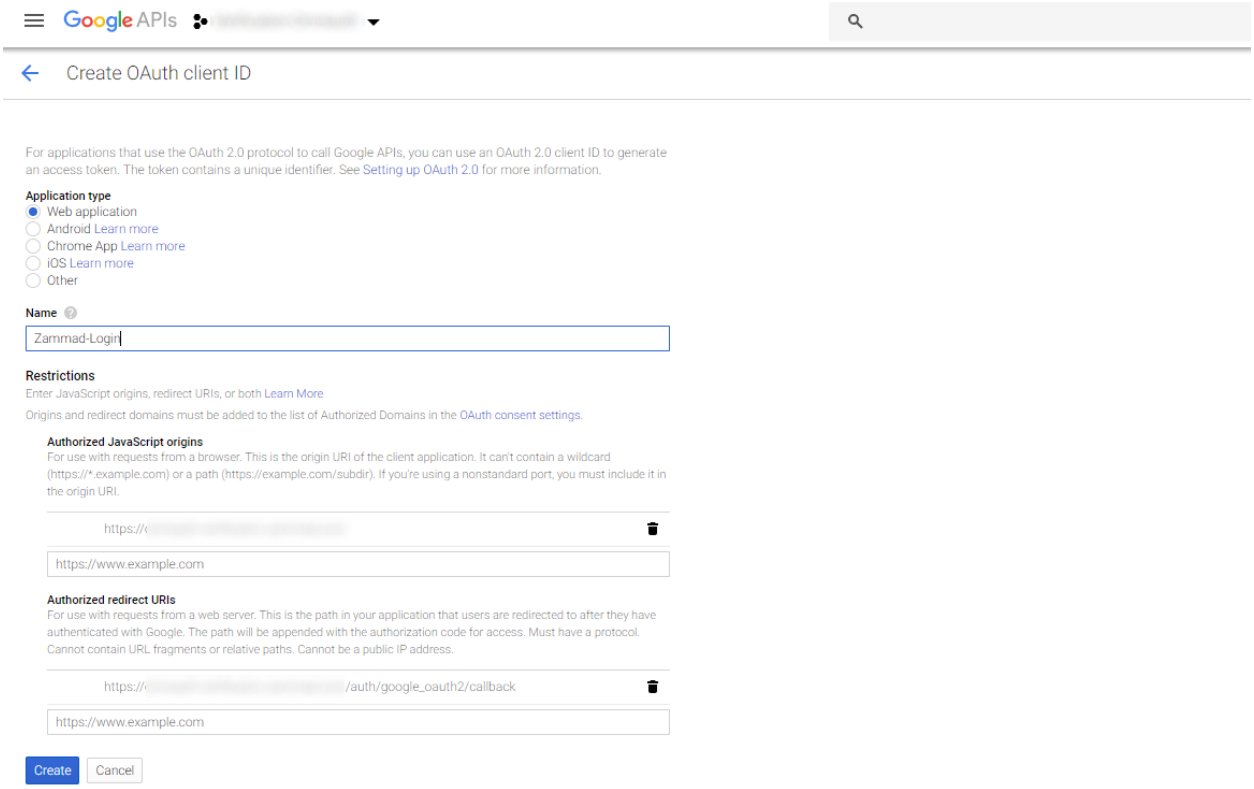
After filling in and savingthe consent screen information, you can change to „Credentials“ tab and create new „OAuth client ID“-Credentials.

Fill in the neceassary information, for restrictions you need the following (replace `zammad_host` with your FQDN):

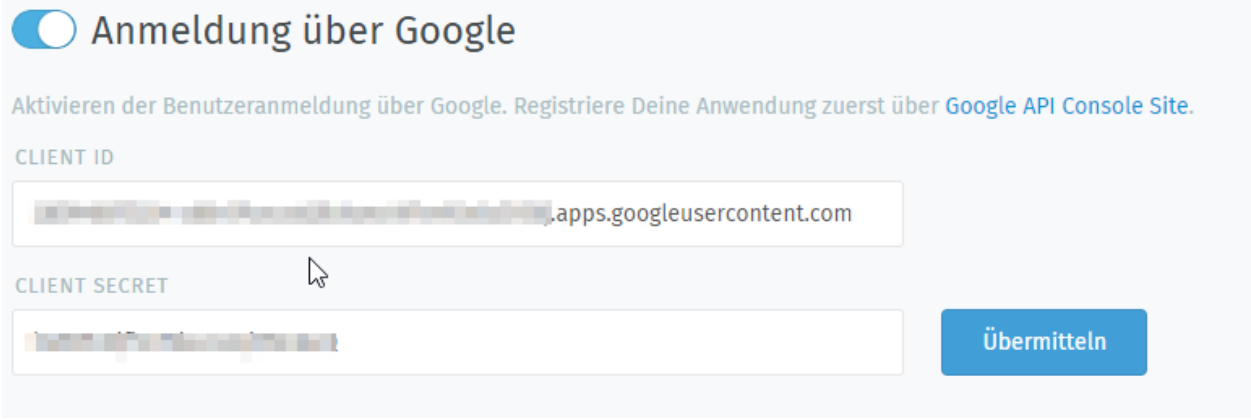
Aplication type `[x] Web application`

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.



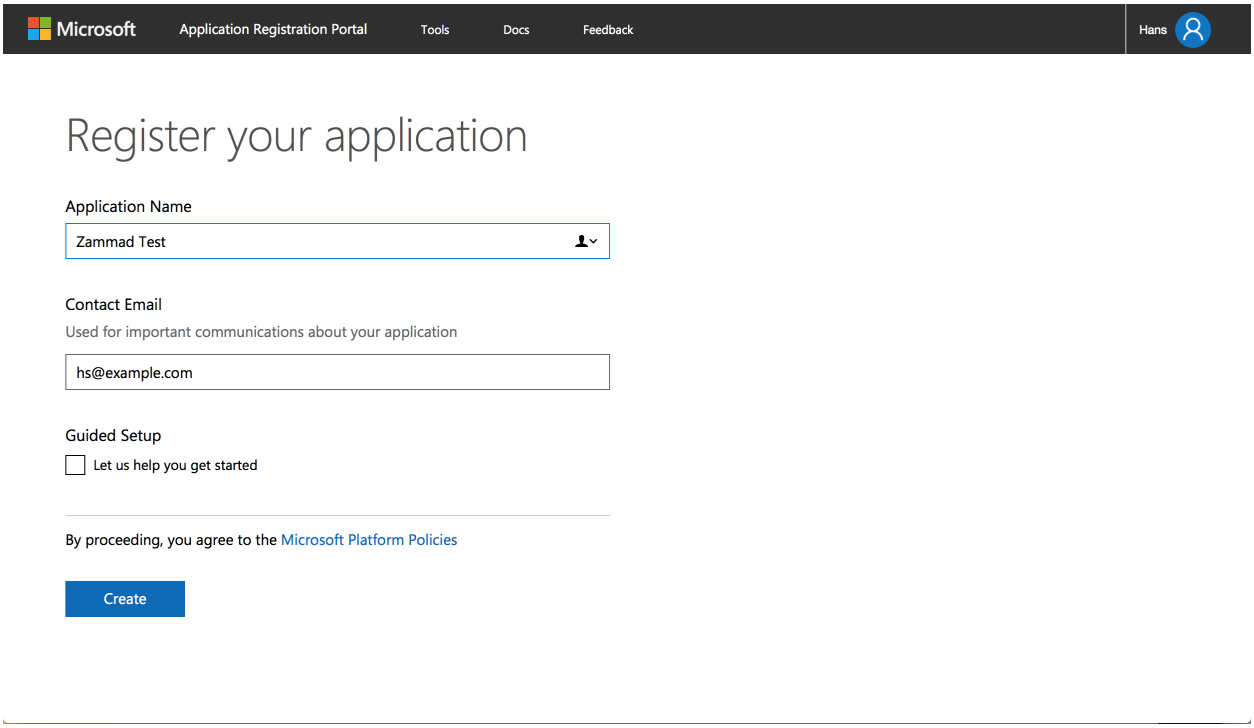
After submitting, the login via Google can be used.

27.3.5 Microsoft / Office365

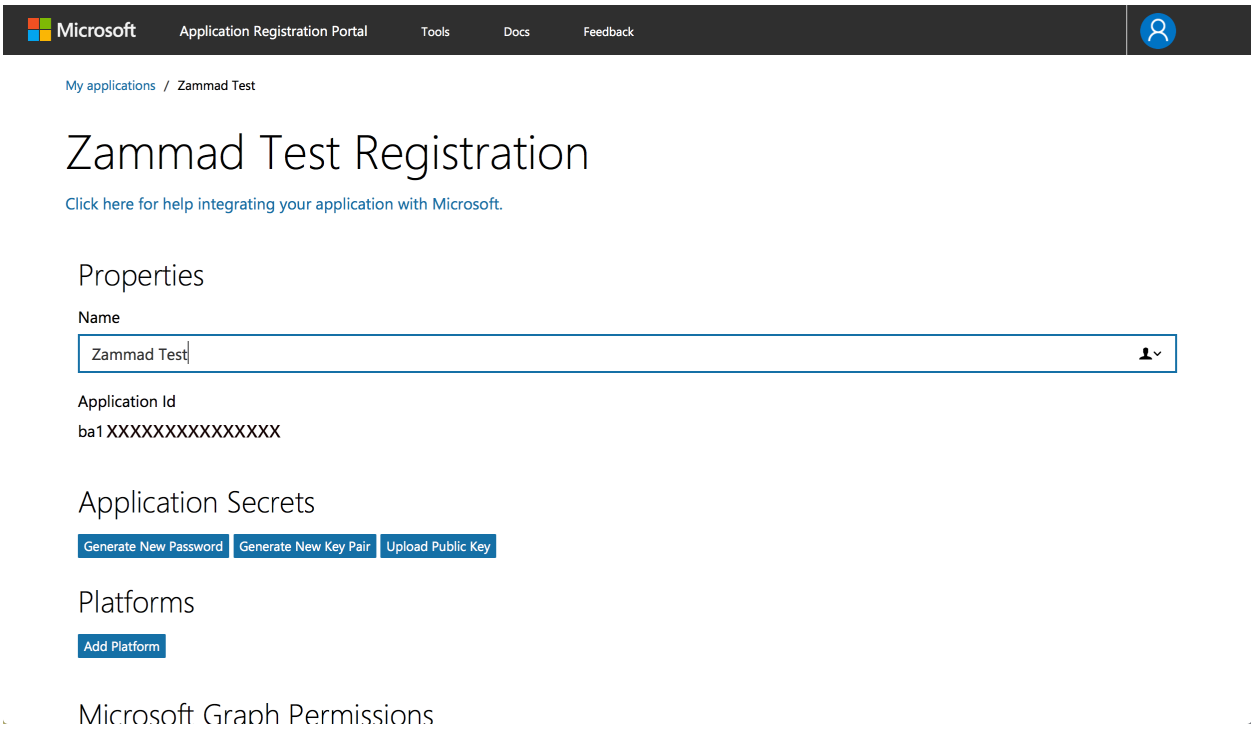
It is possible to create a quick login for your helpdesk via Microsoft (also Office 365) To do so, you need to follow these steps:

Register Microsoft app

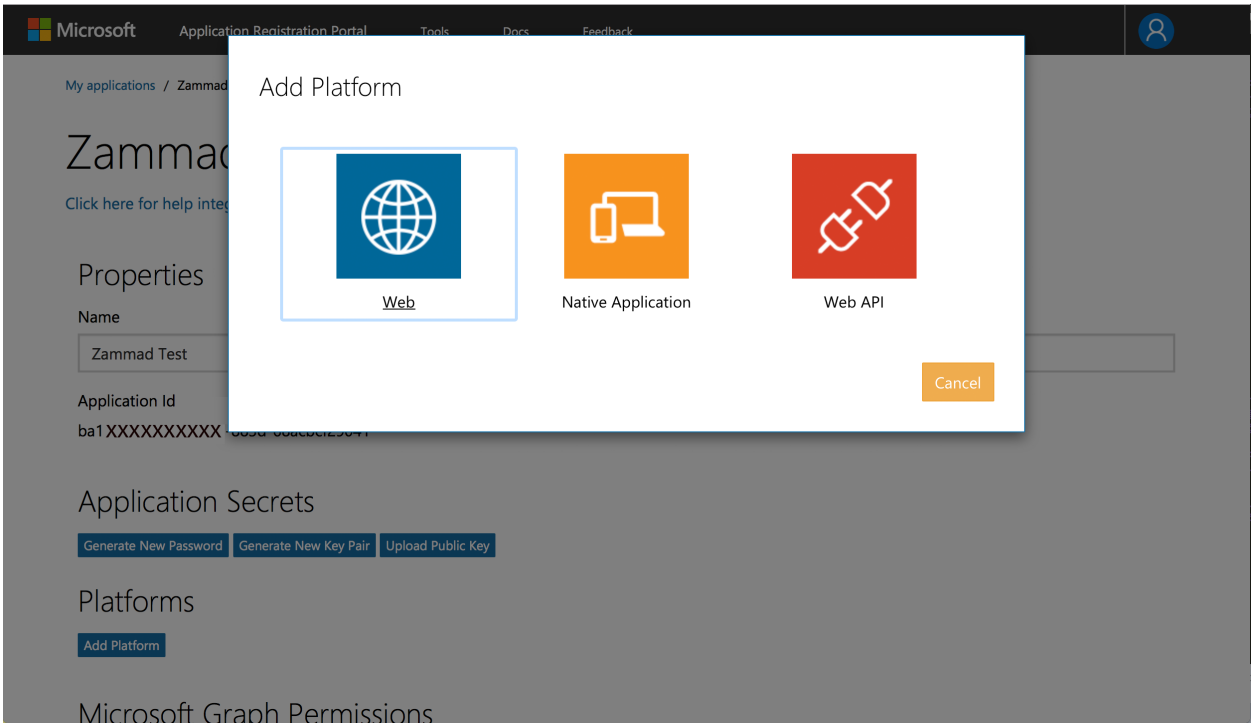
Go to https://identity.microsoft.com (or https://apps.dev.microsoft.com) and login with your account.

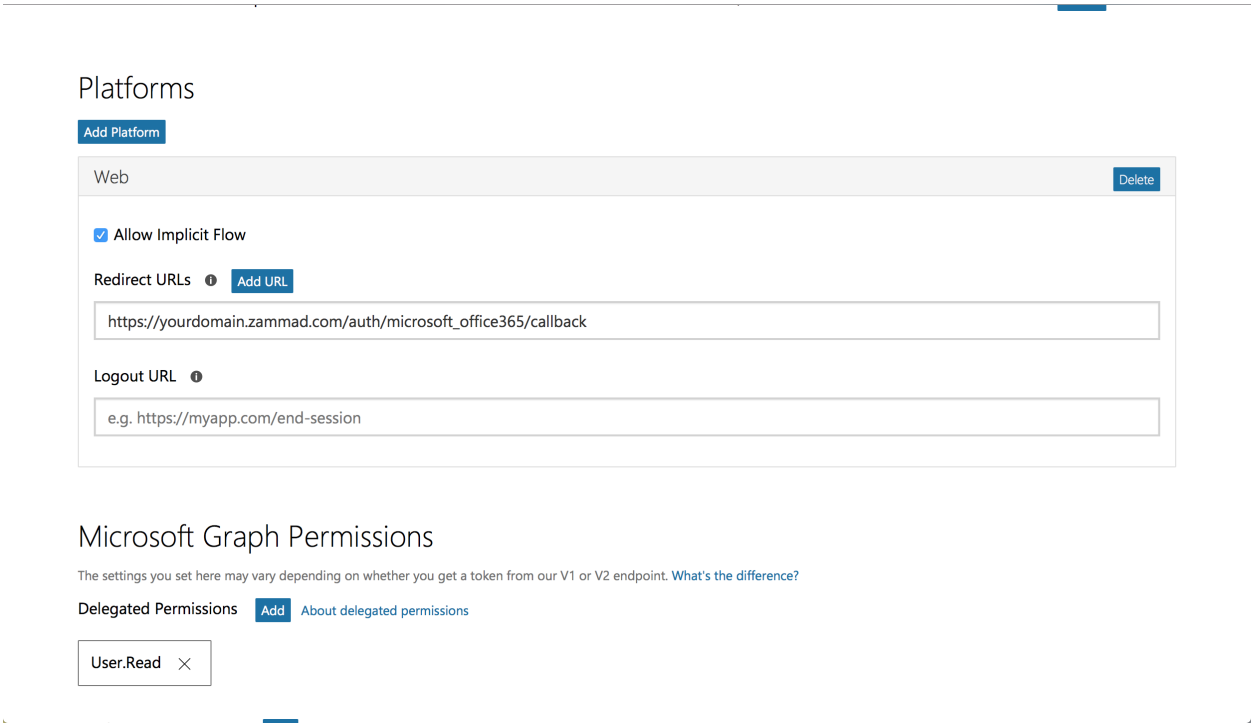


Click on „Create App“



Enter app settings. Then a new plattform and select „web“.





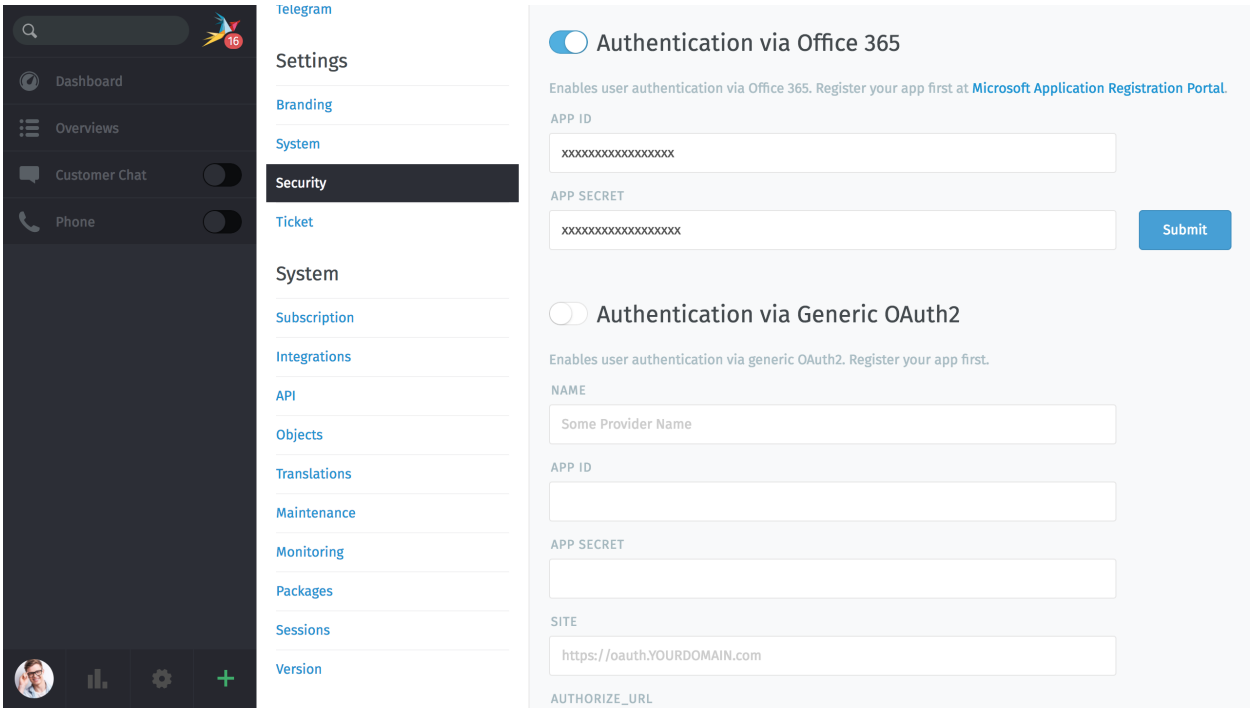
As „Redirect URL“ you need to enter „https://yourdomain.zammad.com/auth/microsoft_office365/callback“

After the app has been created, update the application icon and organization attributes.

Click on „Generate New Password“ button and note the „Password“ and „Application Id“.

Configure Zammad as Microsoft app

Go to „Admin -> Security -> Twitter -> Third Party Applications -> Office 365 Section“



Fill in the „APP ID“ and the „APP SECRET“ and click the „Submit“ button.

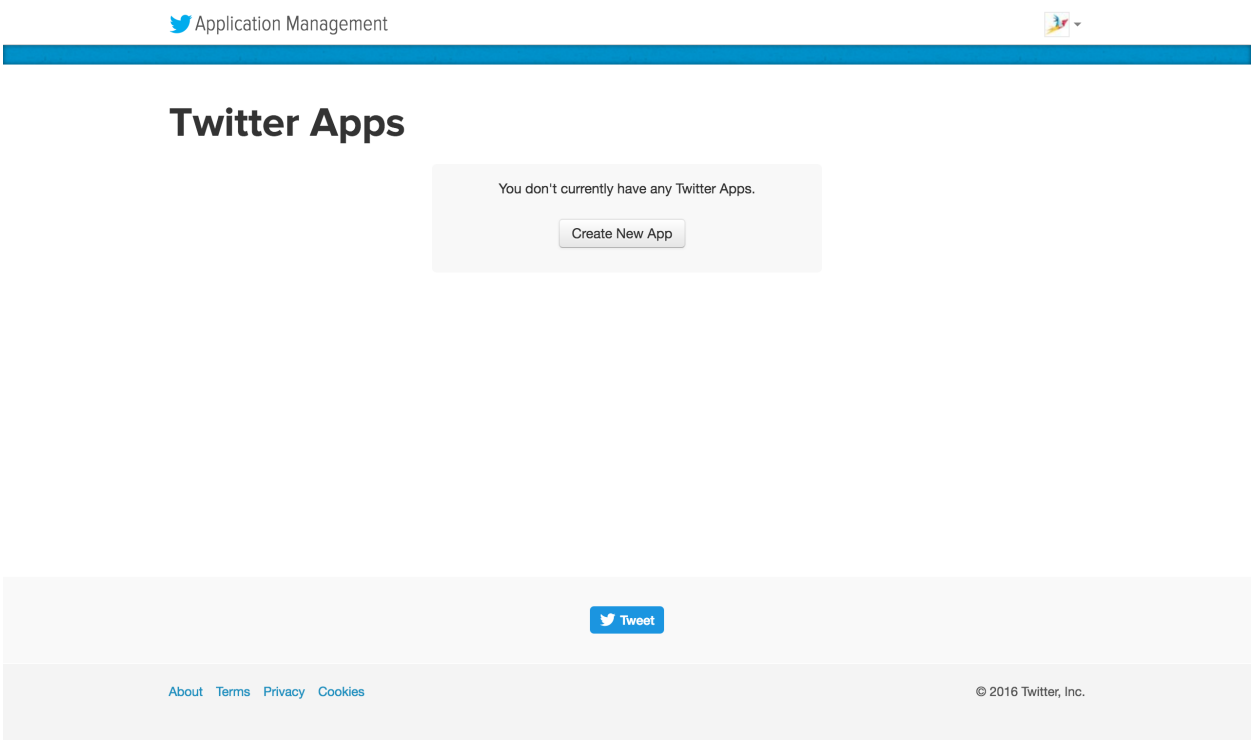
Now you can link accounts via „Avatar -> Profile -> Link Accounts“ or login via Zammad login page.

27.3.6 Twitter

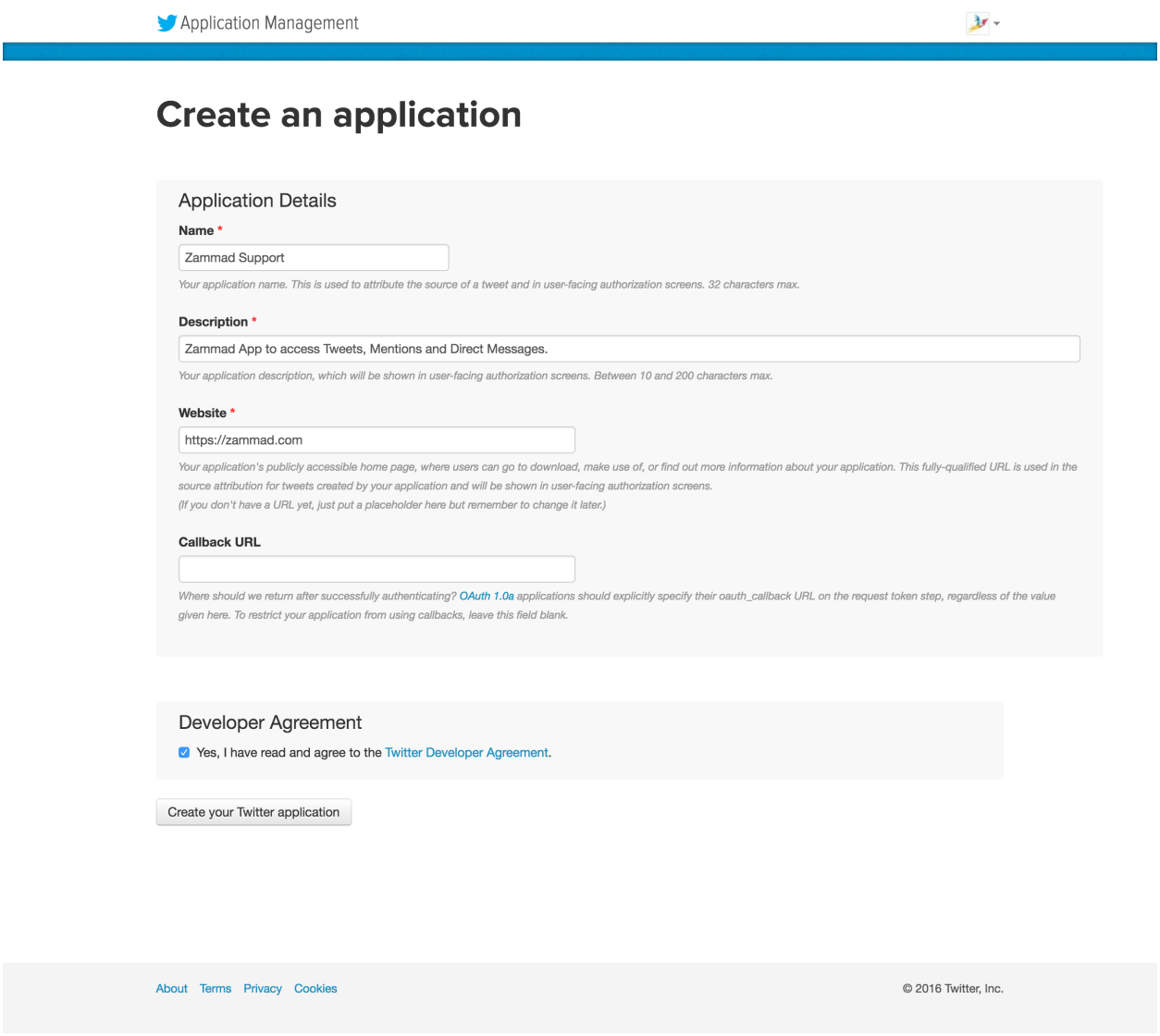
It is possible to create a quick login for your helpdesk via Twitter To do so, you need to follow these steps:

Register Twitter app

Go to https://dev.twitter.com/apps and login with your account.

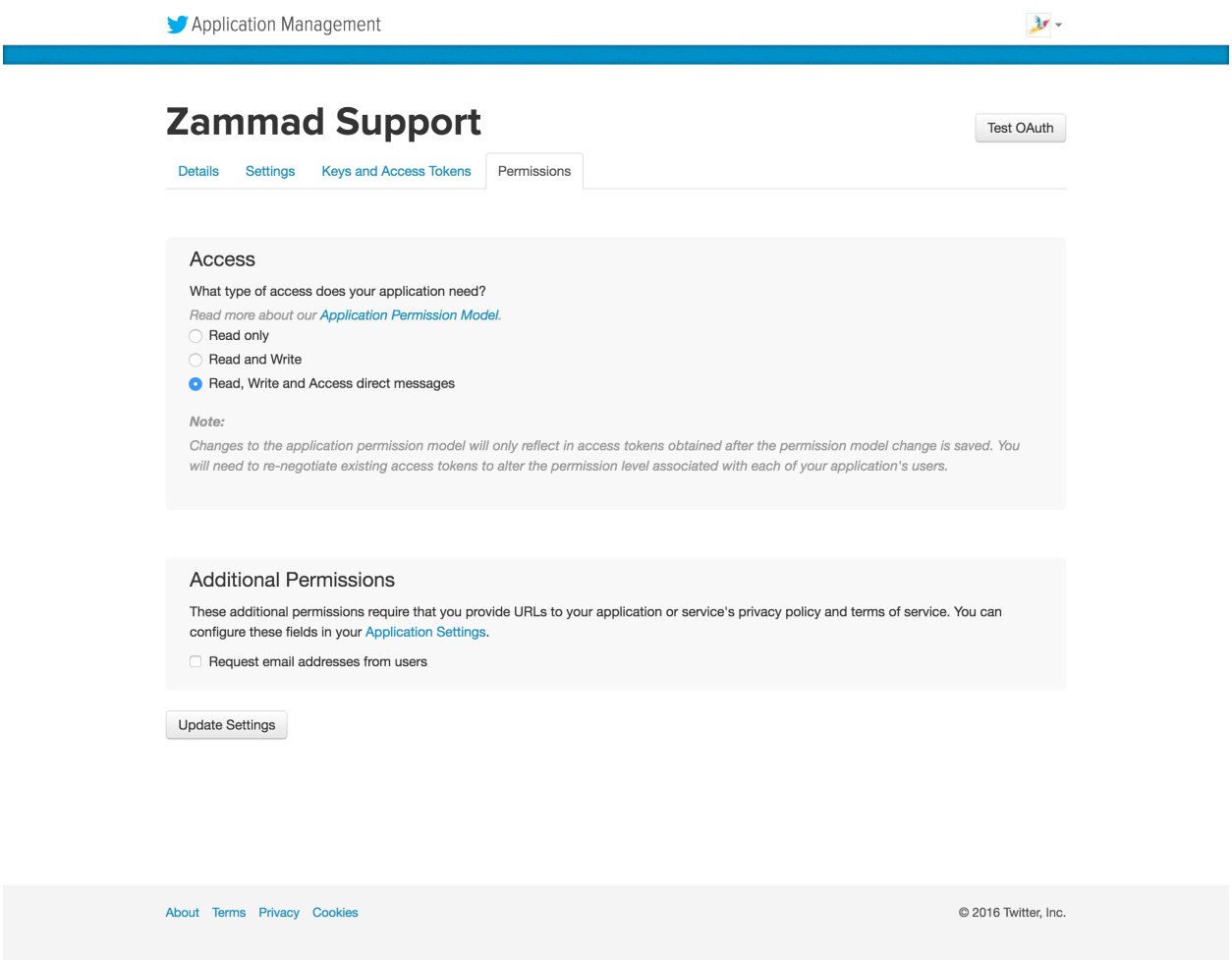


Click on „Create App“

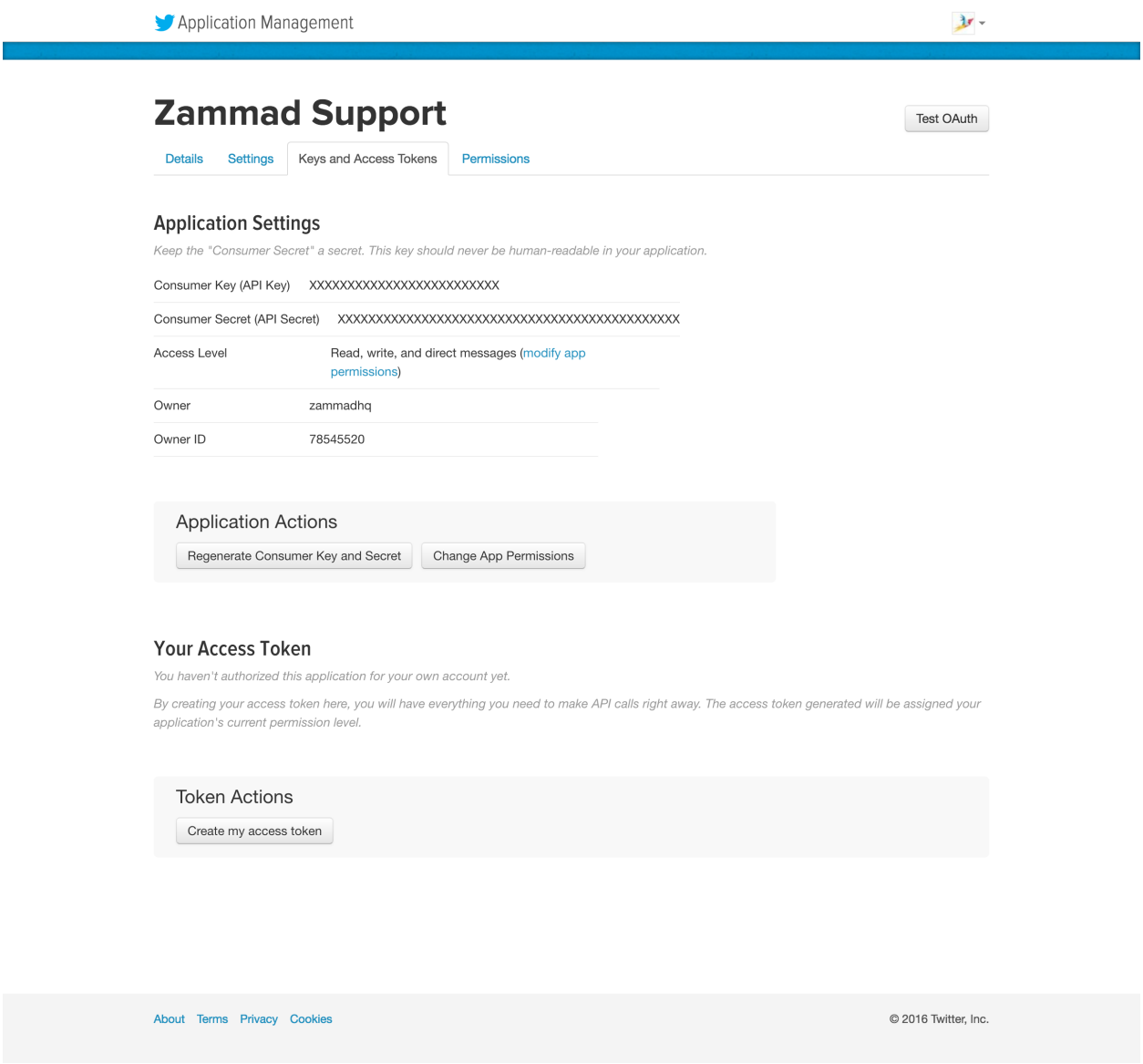


Enter app settings. As „Callback URL“ you need to enter „https://zammad_host/api/v1/external_credentials/twitter/callback“

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.

27.3.7 SAML

Connect your SAML identity provider as a single sign-on (SSO) method.

Bemerkung: 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`.

Bemerkung: 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:

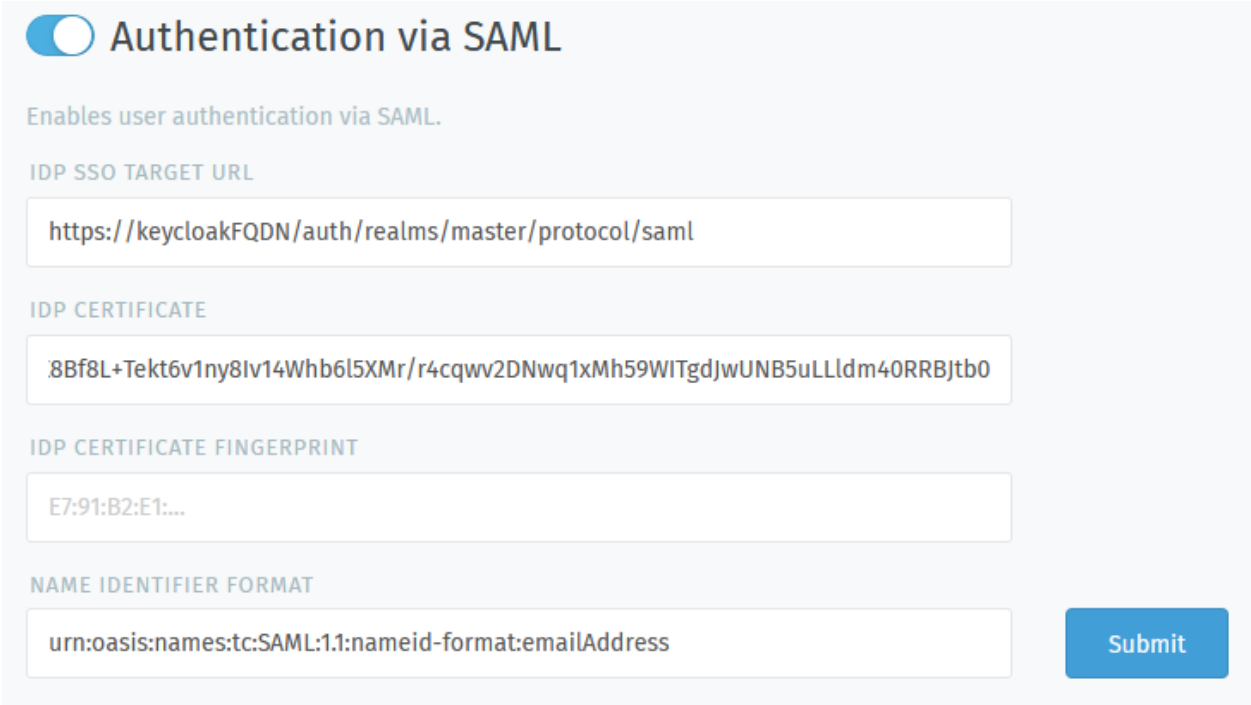
Clients > https://your.zammad.domain/auth/saml/metadata > Mappers > Create	
Name	EmailAddress-Email
Mapper Type	User Property
Property	emailAddress
SAML Attribute Name	email
SAML Attribute NameFormat	basic

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**:



Bemerkung: **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**.

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?

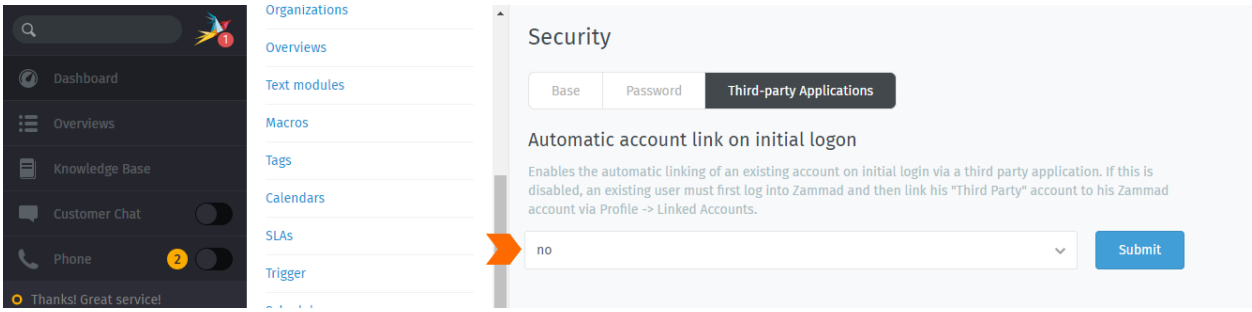
Bemerkung: We're currently missing documentation for the following login providers:

- LinkedIn
- Weibo

27.3.8 Automatic account link on initial logon

In general there's two ways two possible ways to handle how Zammad deals with users it already knows (from whatever source) when you 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 if you're trying to authenticate against a (yet) unknown third party authentication, Zammad will throw an error telling you, that the email address is already in use. If you want to ensure that no matter if your user links his account or not, you might want to enable „Automatic account link on initial logon"



KAPITEL 28

Ticket

Bemerkung: Additional settings for the ticket composer interface can be found in the *Composer Settings*.

28.1 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.

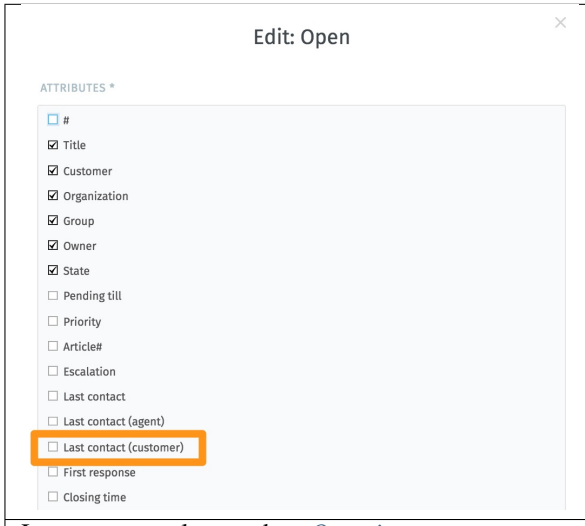
> **Warnung:** Please ensure to take a look at *Settings* within the email channel to ensure you have at least one reference that helps Zammad to assign follow-ups correctly.
>
> Disabling this and not setting up any further follow up search will lead to unexpected results!

Ticket Last Contact Behaviour (default: `Last customer contact (with consideration an agent has replied`
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.

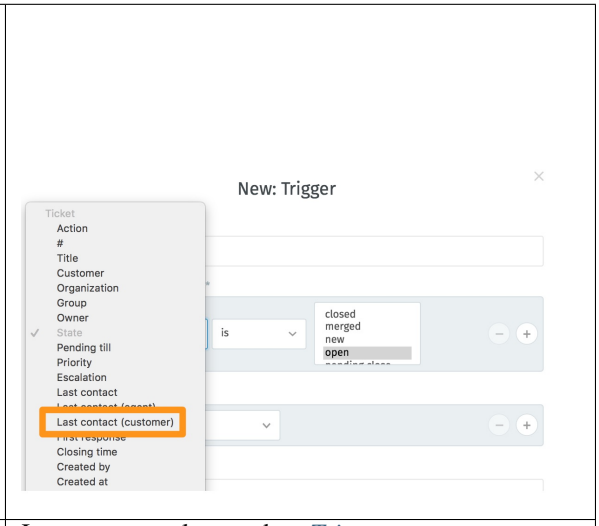
Last customer contact (with consideration an agent has replied to it) If this option is chosen, Zammad will update the last updated value just once if a customer replies. After that, Zammad will wait for an agent to react.

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 „bumb“ the ticket affected.

Last customer contact (without consideration an agent has replied to it) 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*

28.2 Number

Ticket Number Format (default: `Increment (SystemID.Counter)`) This setting defines the way Zammads 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 detections 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.

Bemerkung: This option is only available if ticket number format is set to `Increment`!

28.3 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.

Bemerkung: 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 requirenment. 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.

Bemerkung: The search function in this area is only affective, if you have too many agents to display at once to help you with the configuration.

KAPITEL 29

Integrations

Zammad offers numerous integrations that add rich features to your instance.

Bemerkung: We're still working on this part of our documentation, stay tight!

29.1 Integrations for phone systems

29.1.1 CTI (generic)

Zammad offers phone functionality for users with access to a CTI (computer telephony integration) system (available since version 2.6).

Using the CTI integration enables you to greet customers by name, get their tickets up on the screen with a single click. Or, identify missed calls at a glance, and return calls right from within Zammad. Manufacturers of telephone systems or developers can use this interface to connect their telephone system to Zammad.

What do I need to get started?

Zammad requires the following information to support the agent:

- individual call events (e.g., call incoming/answered/ended)
- caller ID data

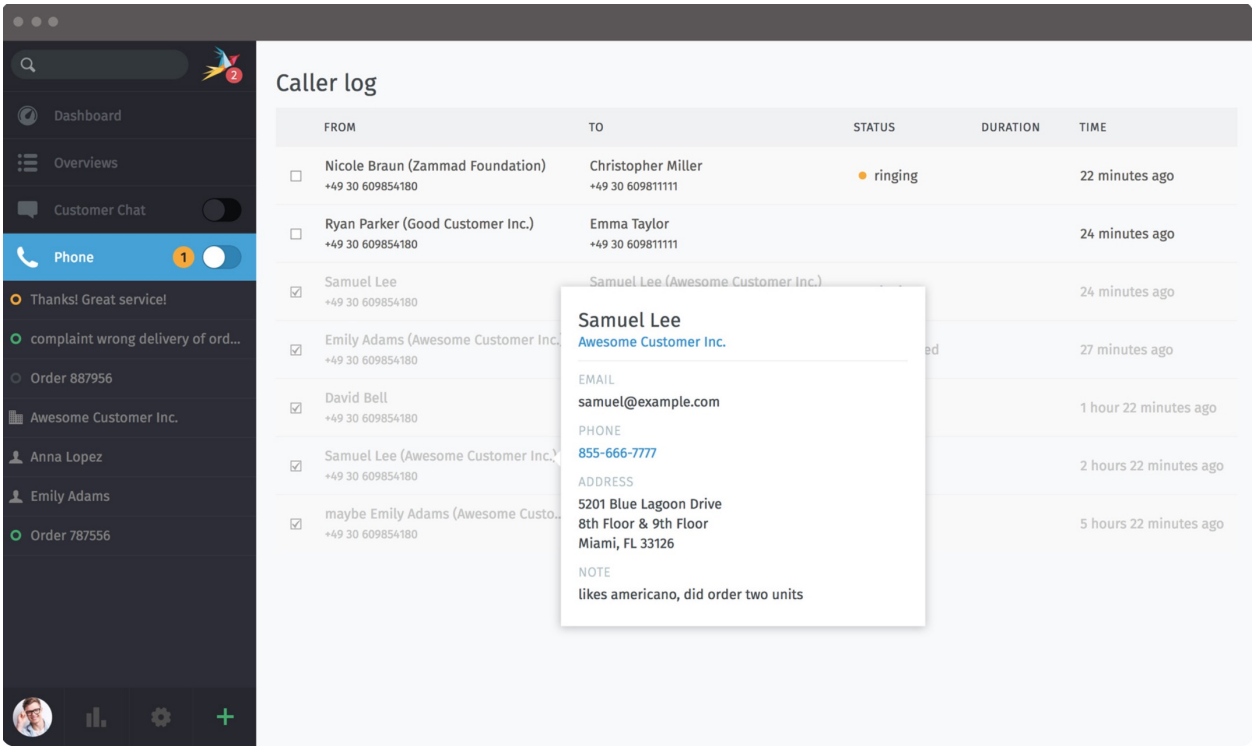
To use the new CTI interface, your CTI system must transmit this information (and only this information) to Zammad via HTTP.

What can it do for me?

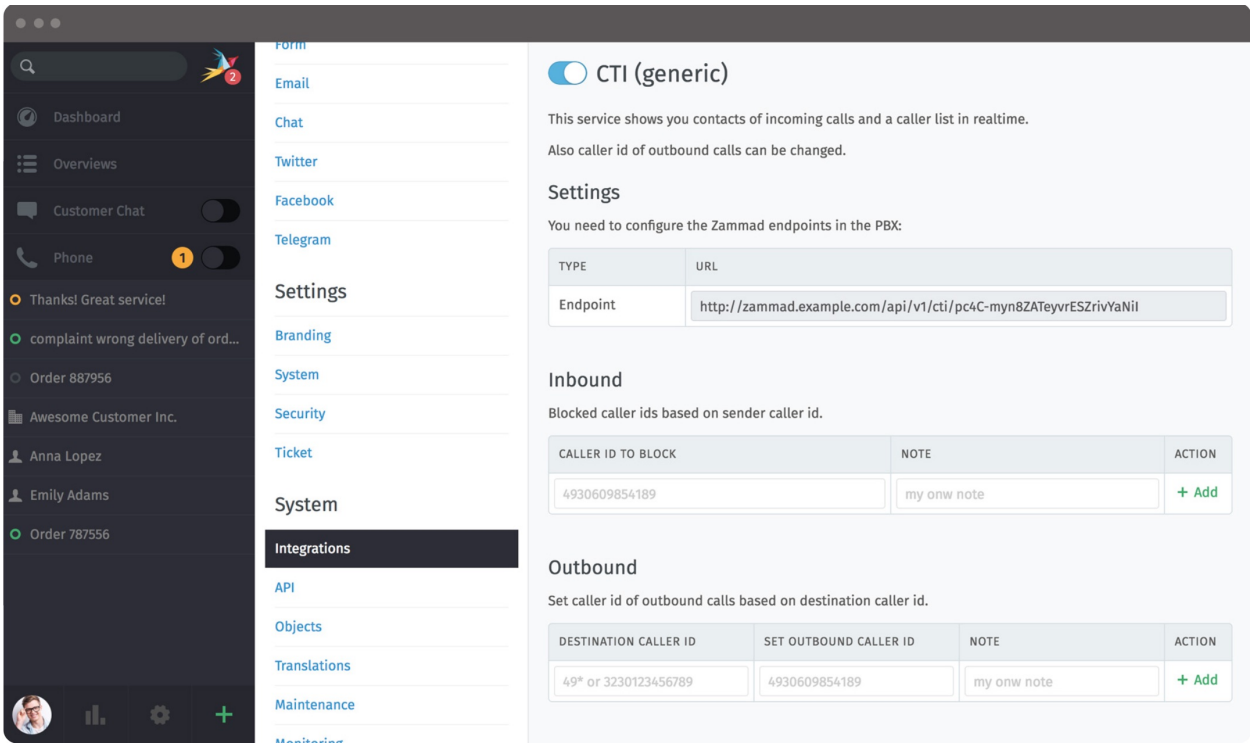
Inbound calls - Caller ID display: Open a caller's customer profile with just one click - Intelligent caller ID search: Automatically scans tickets for caller ID data (e.g., in email signatures) if customer account data is missing - Caller overview: See a caller's entire ticket history, or instantly create a new ticket - Call journal: See all calls at a glance, along with their status (e.g., which ones require a call back?) - Agent overview: See who's currently on a call - Selective call blocking[1] - Do-not-disturb mode[1]

Outbound calls

- Direct dialing: Initiate calls from within Zammad[1]
- Dynamic caller ID: Set your caller ID based on, e.g., the destination country of the call*



[1] requires PBX/telephone system support



More information can be found on our CTI API intro.

29.1.2 Placetel CTI

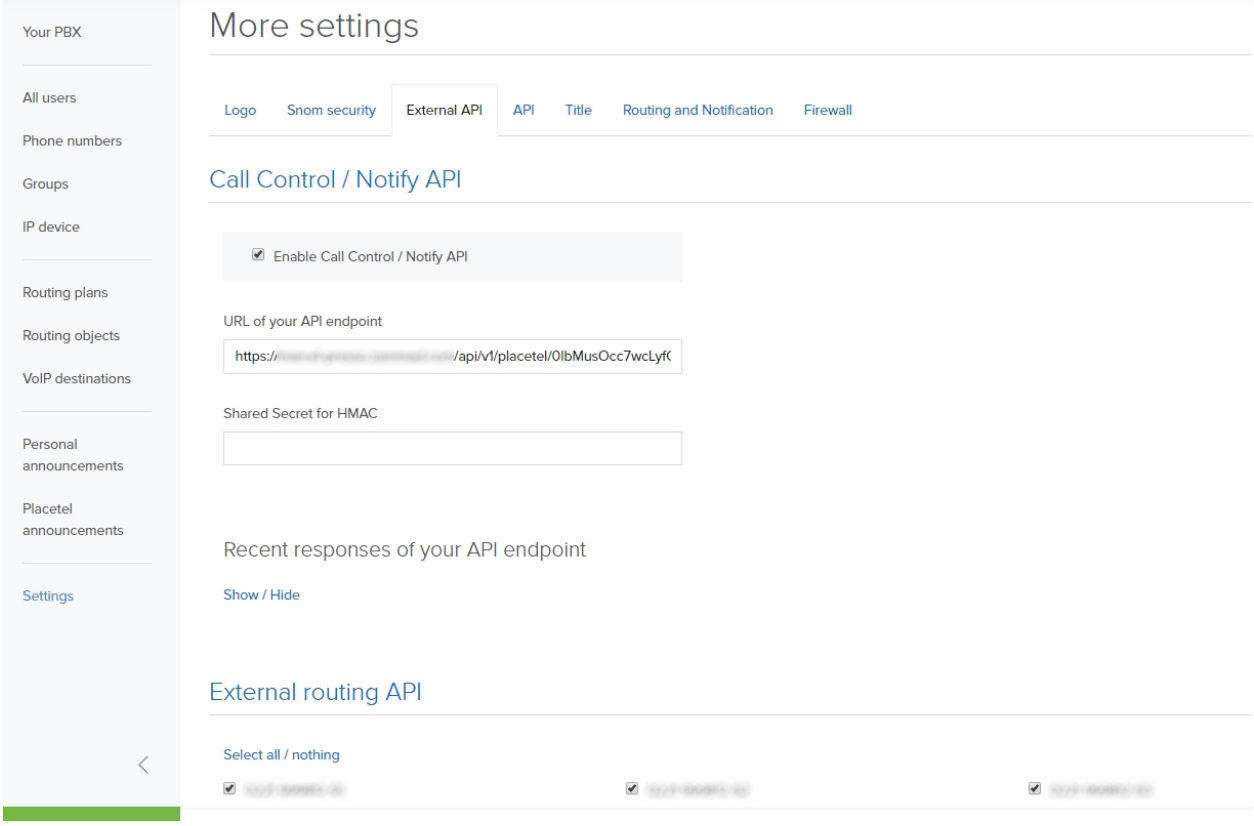
Starting with Zammad 2.8, we're supporting the CTI for placetel. For configuration, please go to Settings -> Integrations -> Placetel in the admin panel.

Limitations

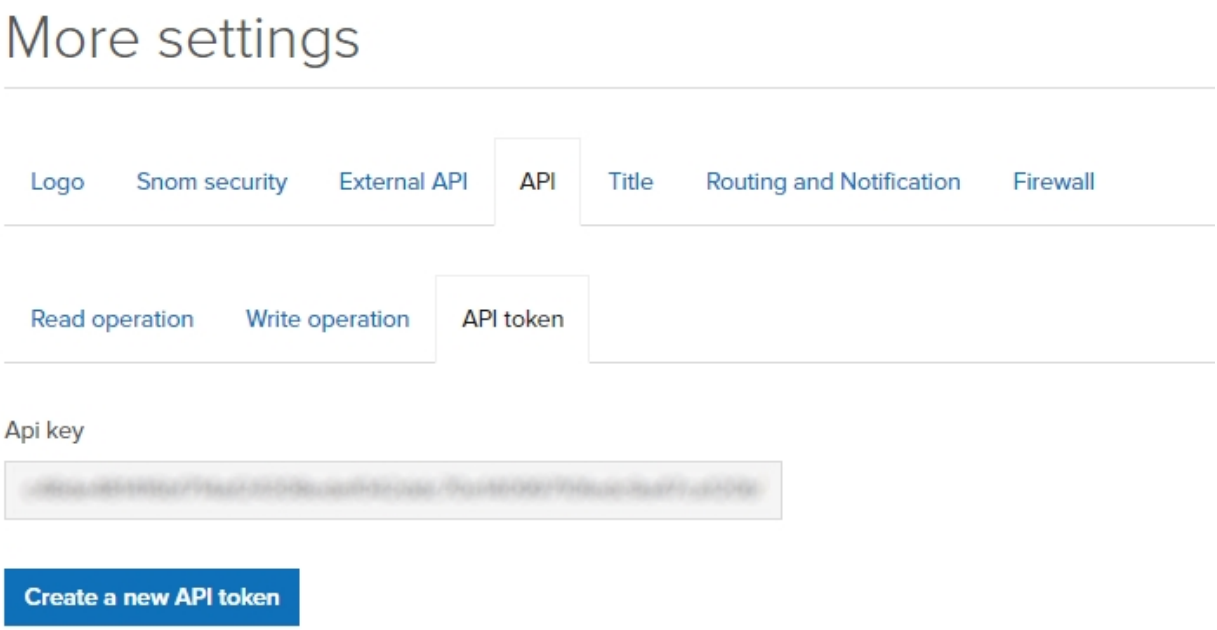
Please note, that it's not possible to log outbound calls. This is due to a limitation of the Placetel API.

Getting needed information from placetel

Before starting with Zammad, you'll need to login to your placetel Account, go to Administration -> Settings. Here you'll find the tab „External APIs" (open it). Activate „Call Control/Notify API" and enter the URL of the Zammad API endpoint (can be found within the Placetel integration page). Further below you can tick the phone numbers you want Zammad to get notified on.



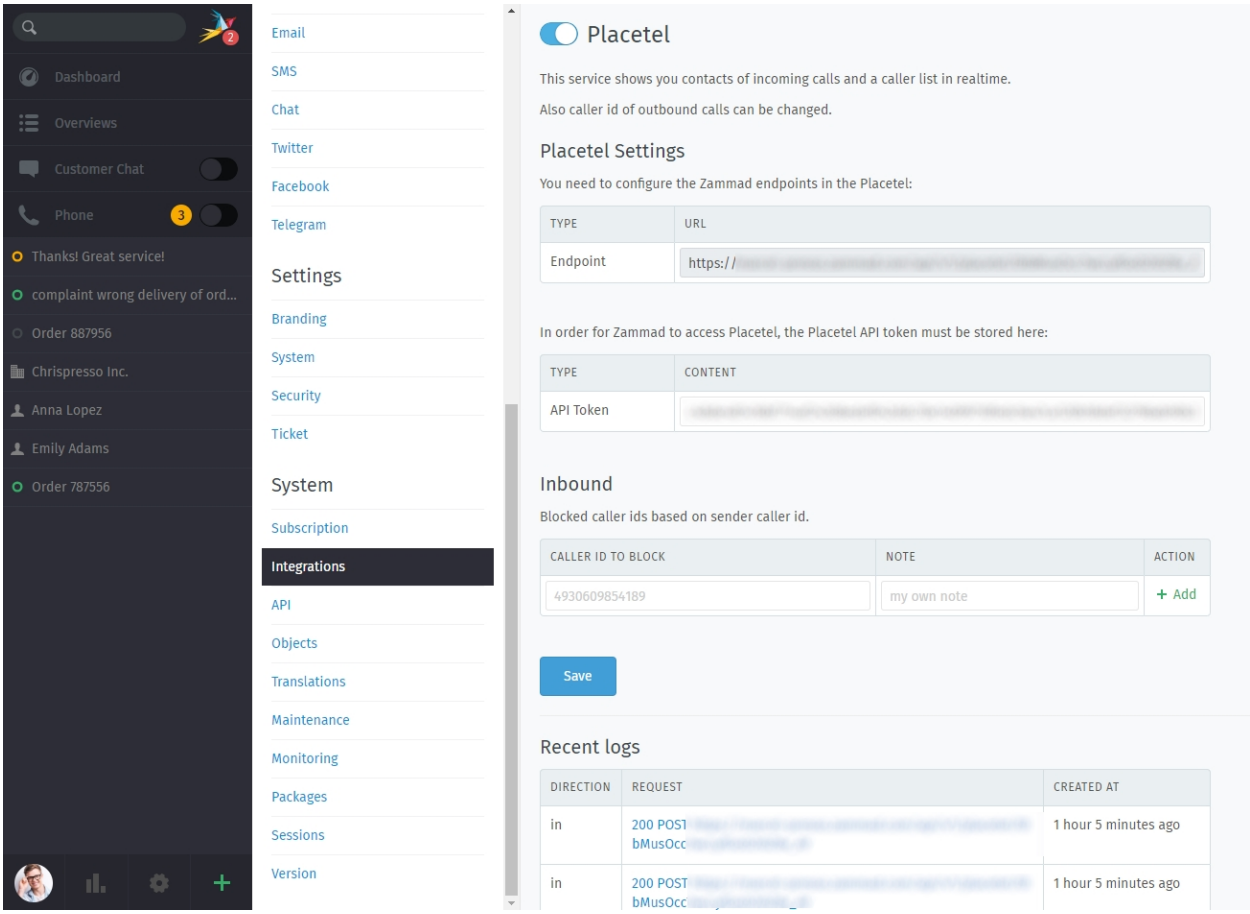
Now change to the API tab and grab your API token. If you don't have an API token yet, you can simply create one.



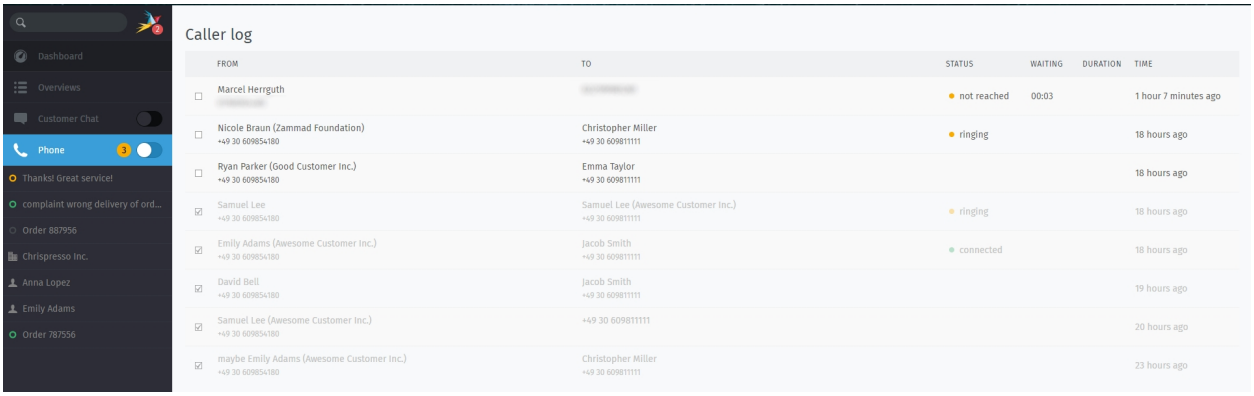
Activating Placetel Integration within Zammad

Enabling the Zammad placetel integration is really easy. Simply paste your API token into the text field, activate the integration and save your changes. You're all set up! If needed, you can add Inbound caller IDs to a ignore list. If you do this, Zammad will not show any notifications from those numbers.

Further below on the same page you'll have a log of recent notifications that came to the API, in case you need to debug something.



As soon as you start calling (in and out) on activated numbers, you'll see them in your Caller log with their current state.



29.1.3 Sipgate

Bemerkung: In order to use this feature, please go to the Feature Store in your sipgate account and activate `sipgate.io`. This feature is available for all account types of sipgate, the store option might differ in name and API calls.

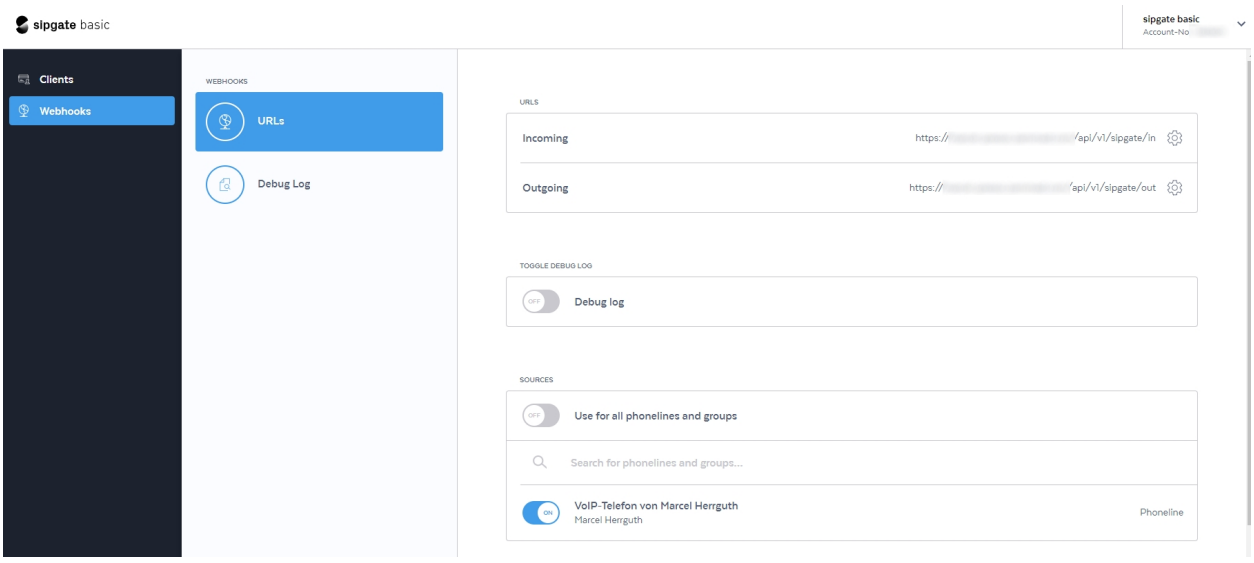
Knowing who's calling is quite important, if you want to bring your support to the next level! Our integration for sipgate enables you to see whos calling who. If you have a customer that wants to get connected to Emmily, you can even check if she's free in the caller log.

For configuration, please go to Settings -> Integrations -> sipgate in the admin panel.

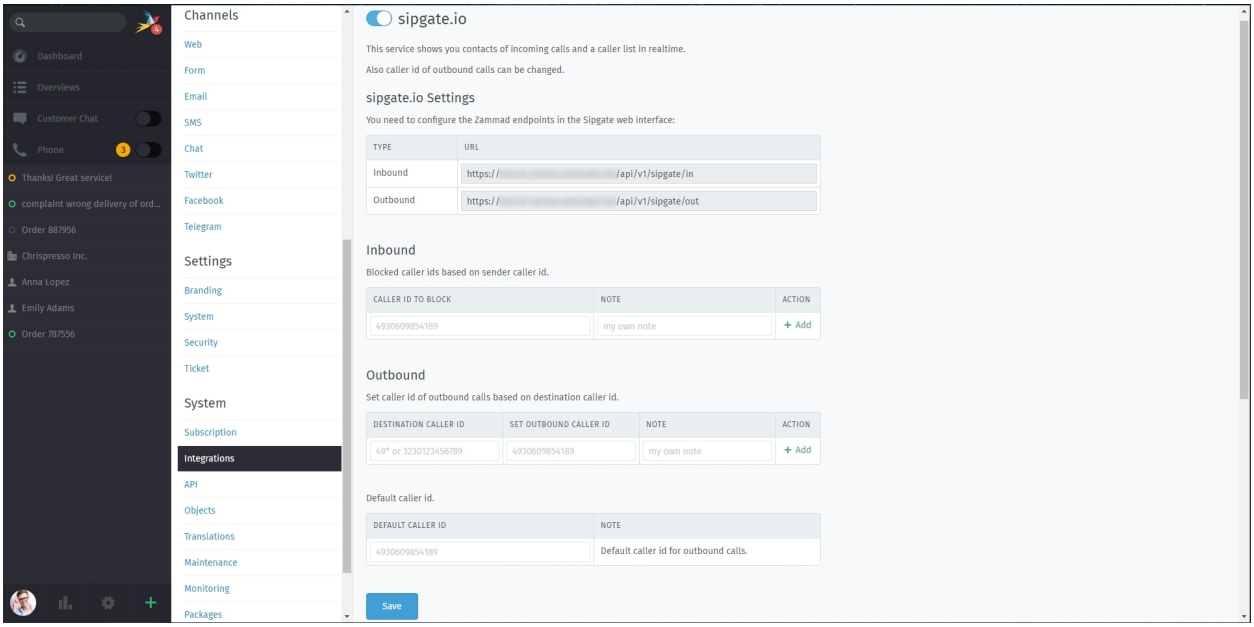
Get your incoming and outgoing URL of your Zammad instance:

```
Incoming: https://zammad.your.tld/api/v1/sipgate/in
Outgoing: https://zammad.your.tld/api/v1/sipgate/out
```

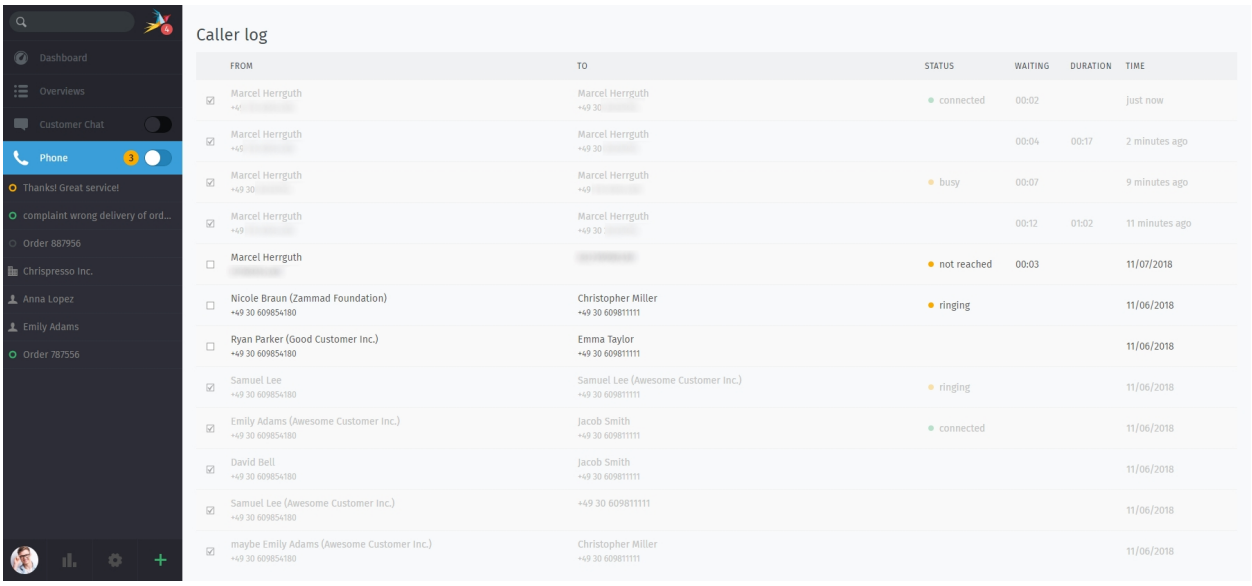
You now need to visit the Console website from sipgate. Go to Webhooks -> URLs and enter your incoming and outgoing URLs there. Further below you can also choose what extensions or groups should appear in your Zammad instance later.



Now you're all set to activate the sipgate integration within Zammad. If you want to, you can inbound and outbound caller IDs to block. This ensures that you can concentrate on your work instead of answering unwanted calls all the time. Setting the default Caller ID is optional and makes sense if you have several numbers. This ensures that your group number for e.g. support is always displayed so your agent don't get called back directly.



You can now switch to your caller log. You'll see missed, in- and outgoing calls here. Pretty easy, right?



Debug Log

In case something doesn't work as expected or you're just curious, Zammad also offers a log of the last API calls on the integration page. If you click on the request, a detailed view will open showing you the exact request.

Recent logs

DIRECTION	REQUEST	CREATED AT
in	200 POST https://[illegible]/api/v1/sipgate/in	just now
in	200 POST https://[illegible]api/v1/sipgate/in	1 minute ago
in	200 POST https://[illegible]/api/v1/sipgate/in	1 minute ago
in	200 POST https://[illegible]api/v1/sipgate/out	8 minutes ago
in	200 POST https://[illegible]/api/v1/sipgate/out	8 minutes ago
in	200 POST https://[illegible]/api/v1/sipgate/in	8 minutes ago
in	200 POST https://[illegible]/api/v1/sipgate/in	9 minutes ago
in	200 POST https://[illegible]/api/v1/sipgate/in	10 minutes ago

29.2 Integrations for authentication and customers

29.2.1 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)

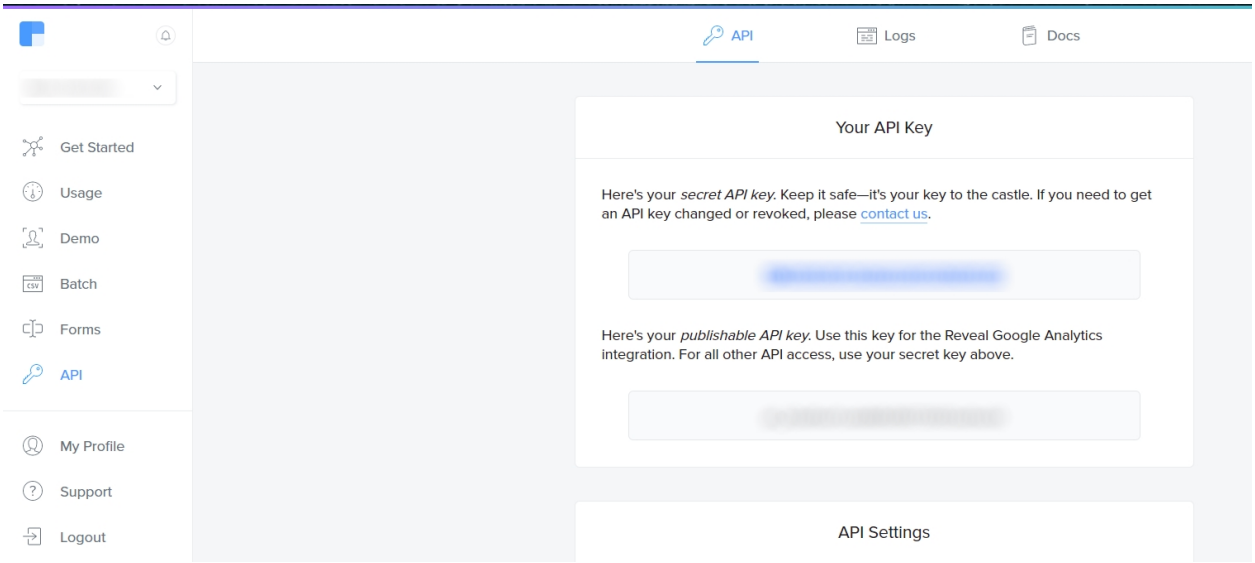
If needed, you can add further custom objects and add mappings to them, so the clearbit information can be filled within the database. In general you can get any information from clearbit, as long as you have a mapping to an Zammad object.

Hinweis: 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 lastname field filled in already and Clearbit has other information on that, it will not be updated.

However: If let's say the lastname has been set by Clearbit and Zammad notices that the lastname on the Clearbit source has changed, Zammad will also update this value.

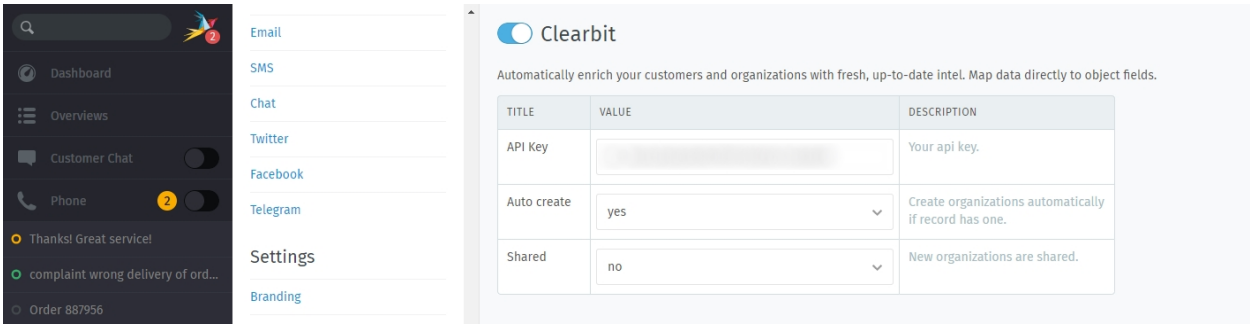
Configuration

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.

Bemerkung: If you’re unsure what option to choose, better stick with „no“. You can also learn more about *Organizations*.



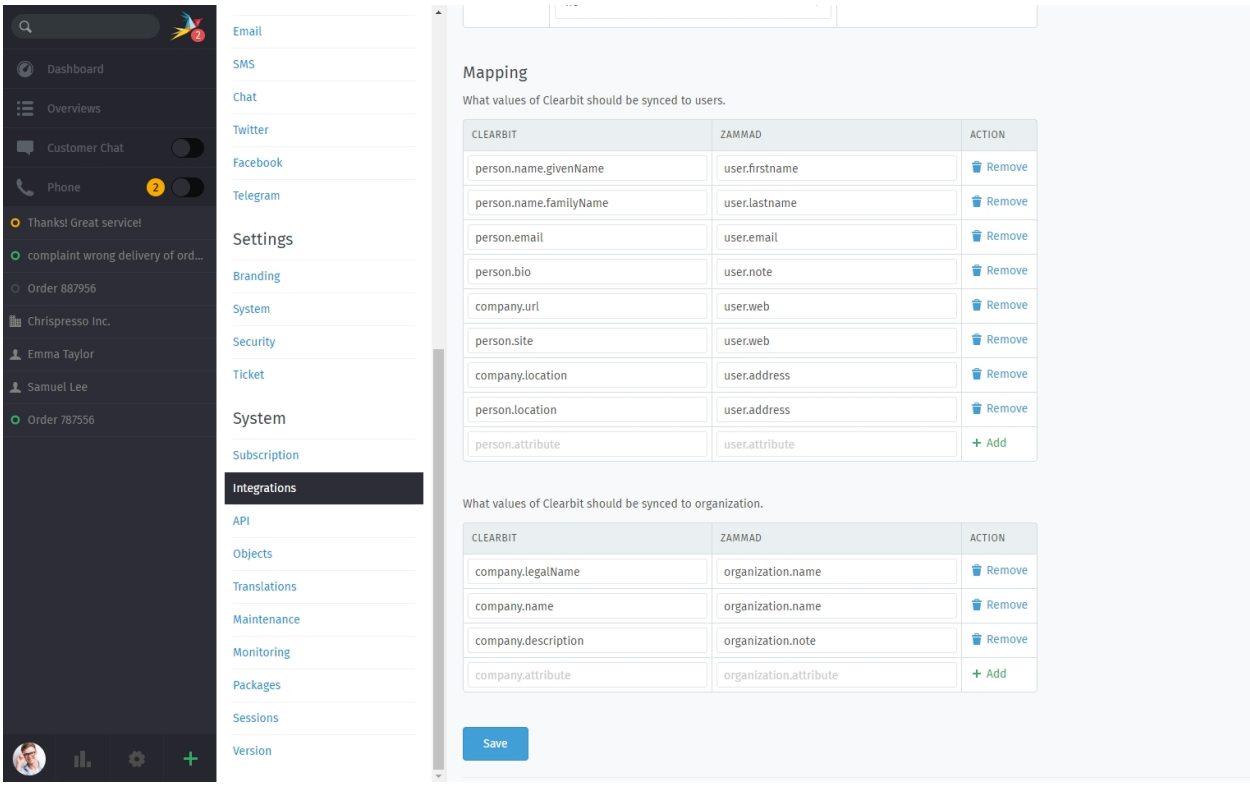
The Mapping option works similar to the mapping within the Exchange and LDAP sync. You can add further mappings for e.g. custom fields if you need more information that Clearbit can provide.

Bemerkung: 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 .

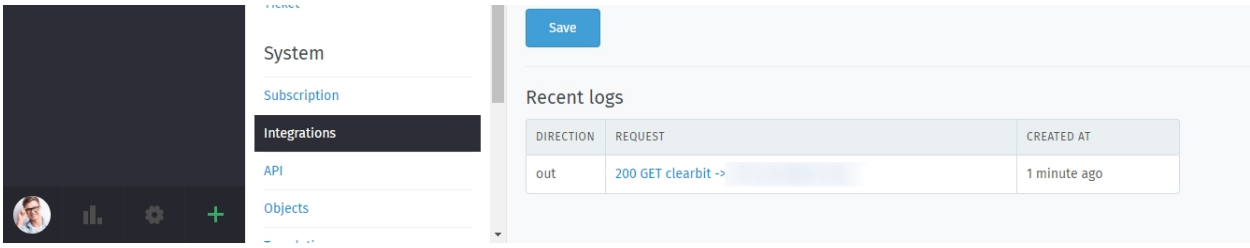
If you’re happy with the above chosen Settings and your mapping, just save the changes and enable Clearbit integration. Zammad will now start polling the Clearbit API as users contact you.

Bemerkung: Zammad does not synchronize with Clearbit on a regular basis, but on demand if needed. This saves

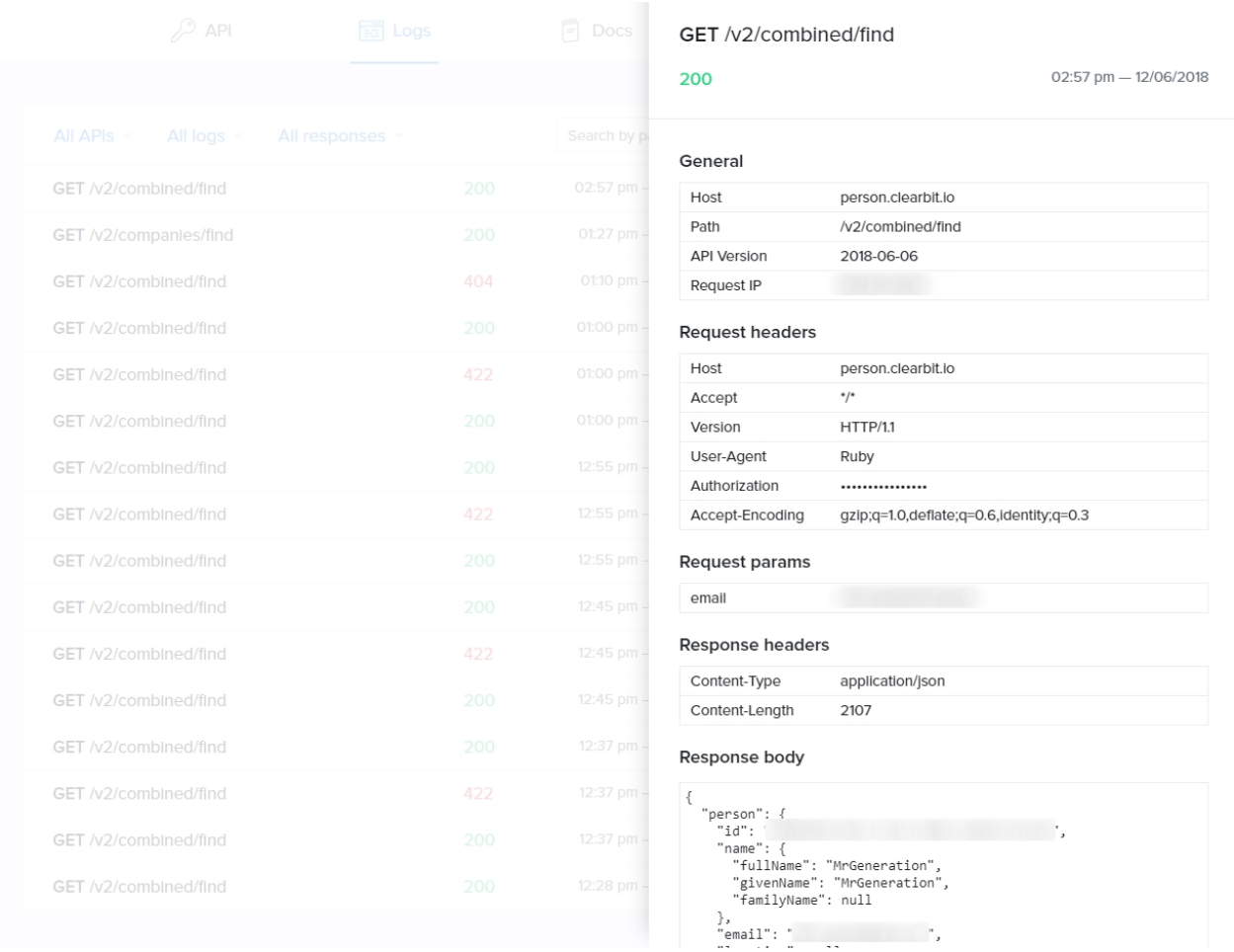
API calls.



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 lastname of Alex to „X“, Zammad did not update it. What Zammad did was to add further information it received from Clearbit. Really cool, right?

29.2.2 S/MIME

New in version 3.4, Zammad supports S/MIME for high-security email communication.

Abb. 1: Once S/MIME has been enabled, **Encrypt** and **Sign** buttons will appear in the ticket composer.

What is S/MIME?

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**.

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**.)

Bemerkung: 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.**

Setup

S/MIME is **disabled by default**. Enable it to start adding certificates.

Abb. 2: 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 Import public-key certificates for both your own organization and your contacts.

> **Warnung: 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.

Bemerkung: 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.

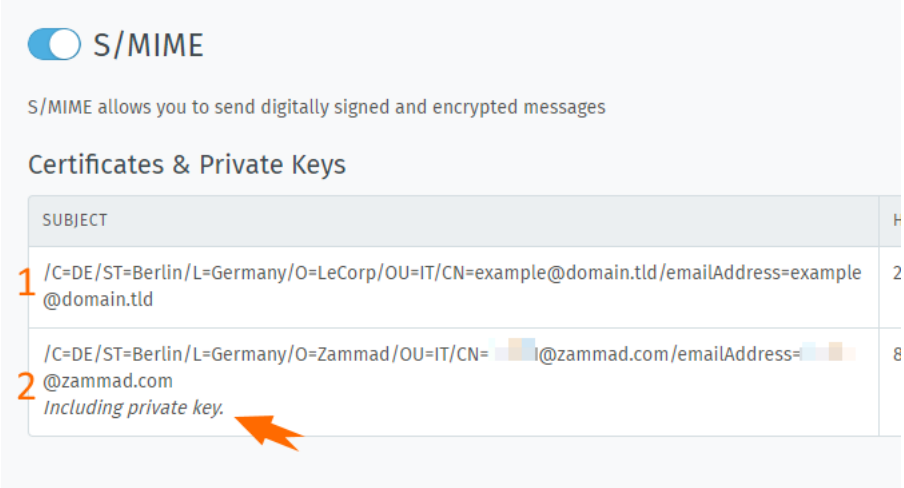


Abb. 3: A note is displayed on certificates with a matching private key (see line 2).

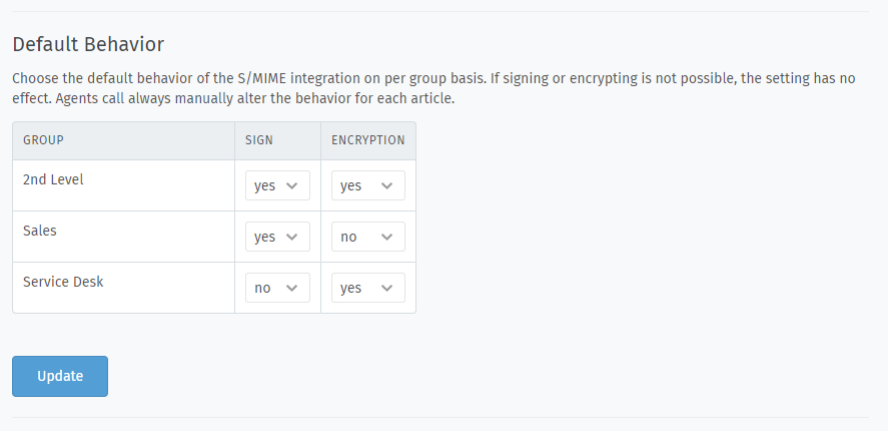
Bemerkung: Certificates and private keys must be uploaded separately.

If your certificate and private key are bundled together in the same file or PEM block, import it twice (once using each button).

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.

Recent Logs

All of the system's latest S/MIME activity is displayed in the **Recent Logs** section.

Abb. 4: Logs report the status and details of all mail, both incoming and outgoing, that used signing/verification or encryption/decryption.

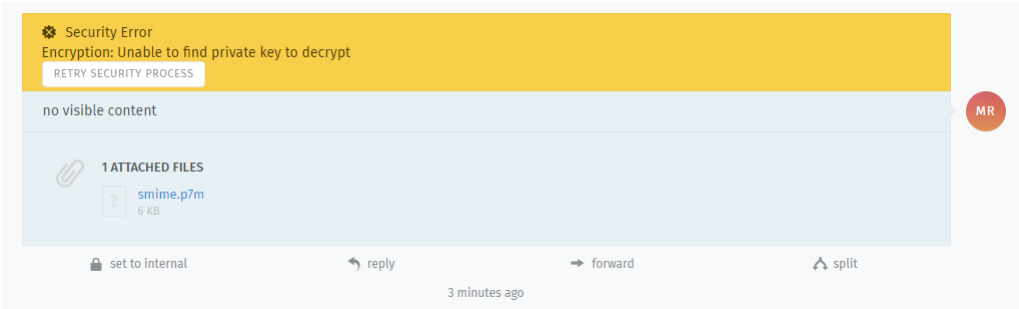
Bemerkung: This log does **not** include email sent by *triggers* or the *scheduler*. For those, check your `production.log`.

Usage

For more details on how S/MIME integration works in practice, see the user docs.

Troubleshooting

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?

Warnung: 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?

29.3 Integrations for Monitoring Systems

Bemerkung: This section currently misses the following integrations:

- Icinga
- Monit
- Nagios

29.3.1 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:

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:

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

curl -X POST \
  -F "event_id=$NOTIFY_SERVICEPROBLEMID" \
  -F "host=$NOTIFY_HOSTNAME" \
```

(Fortsetzung auf der nächsten Seite)

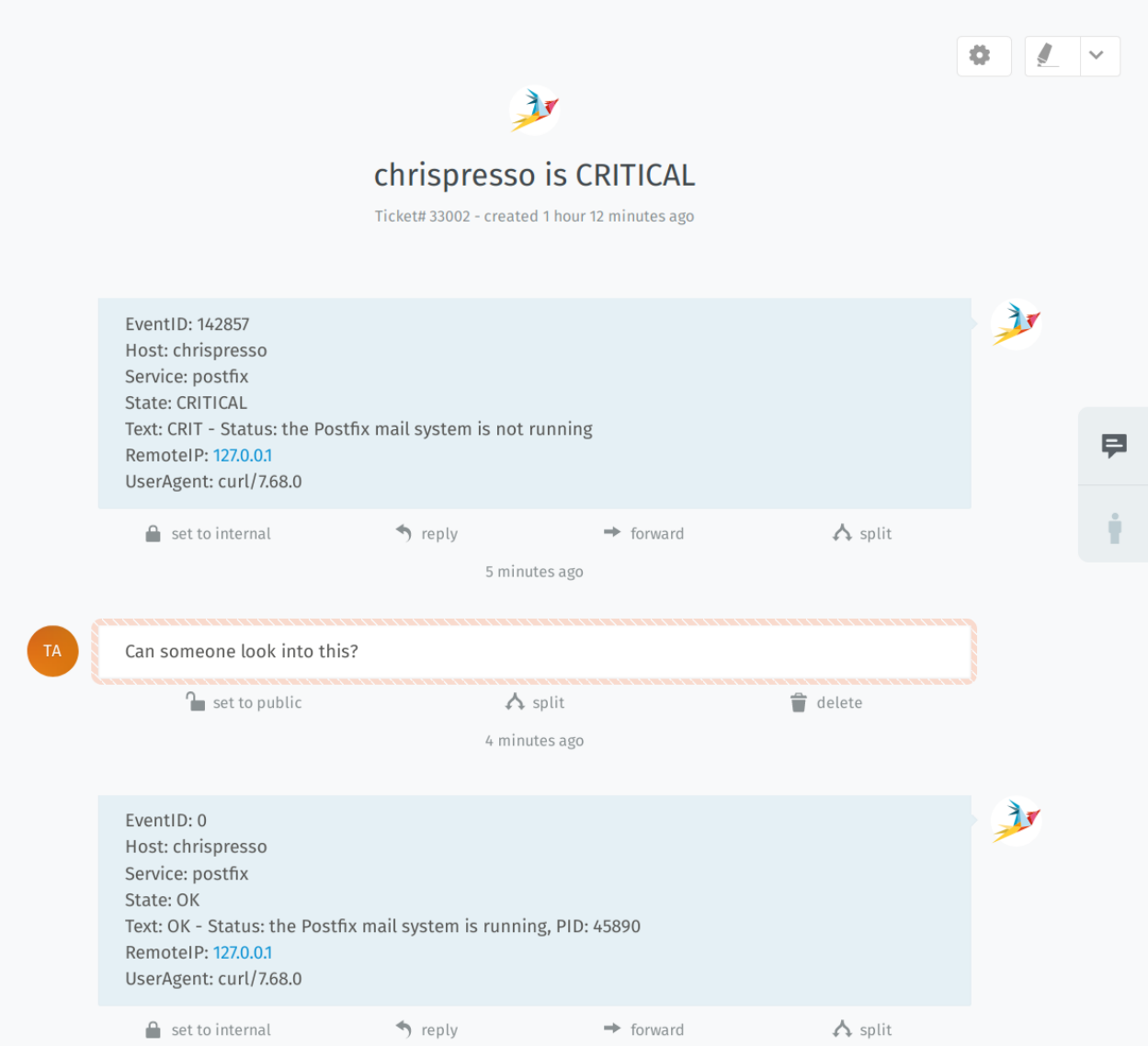


Abb. 5: Multiple alerts, same ticket.

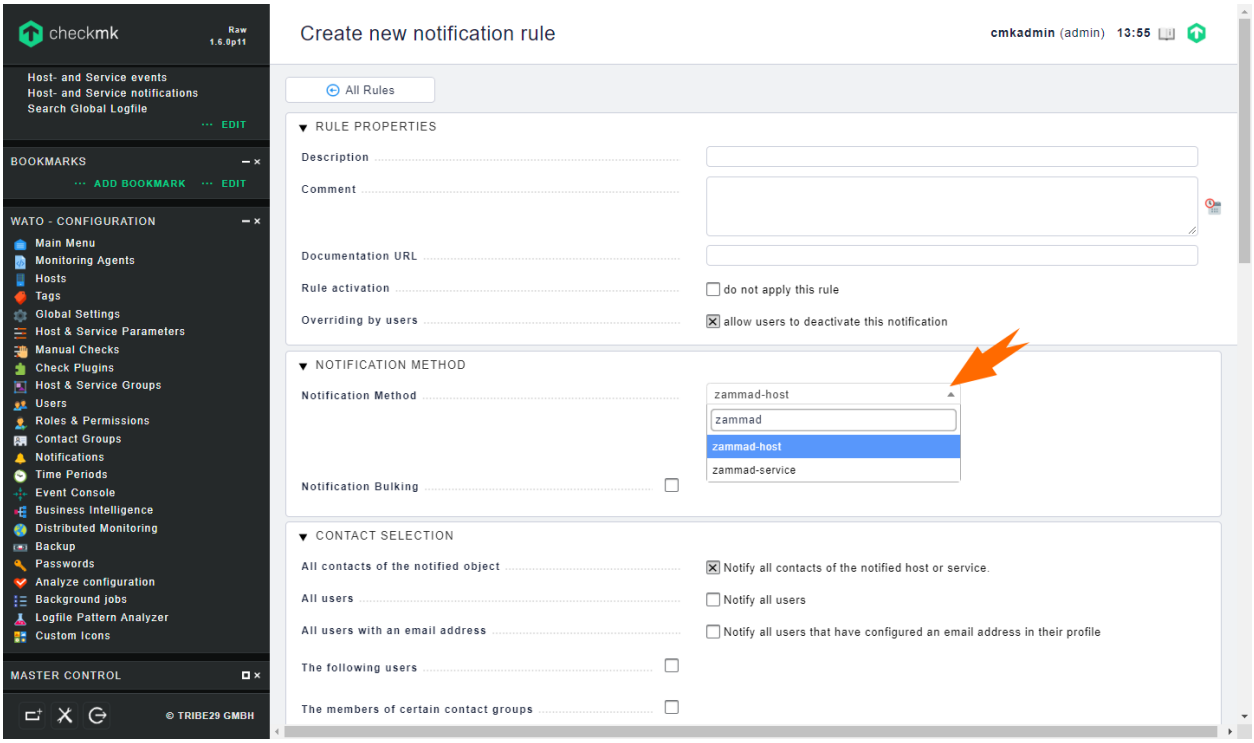


Abb. 6: Find your custom scripts in the Checkmk WATO under **Notifications > New Rule > Notification Method**.

(Fortsetzung der vorherigen Seite)

```
  -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
```

Bemerkung: 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

> **Warnung:** **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.

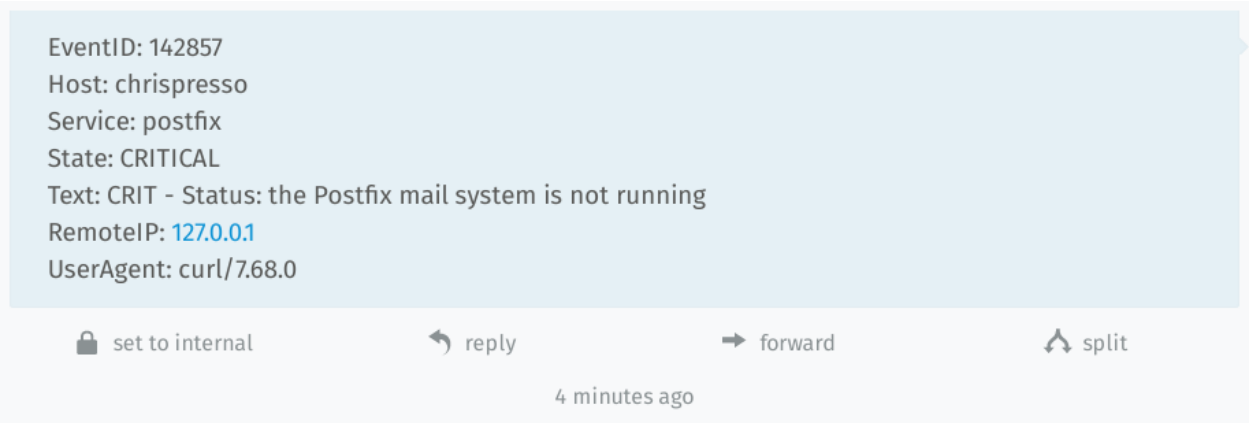
Hinweis: **It's just an API endpoint!**

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

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

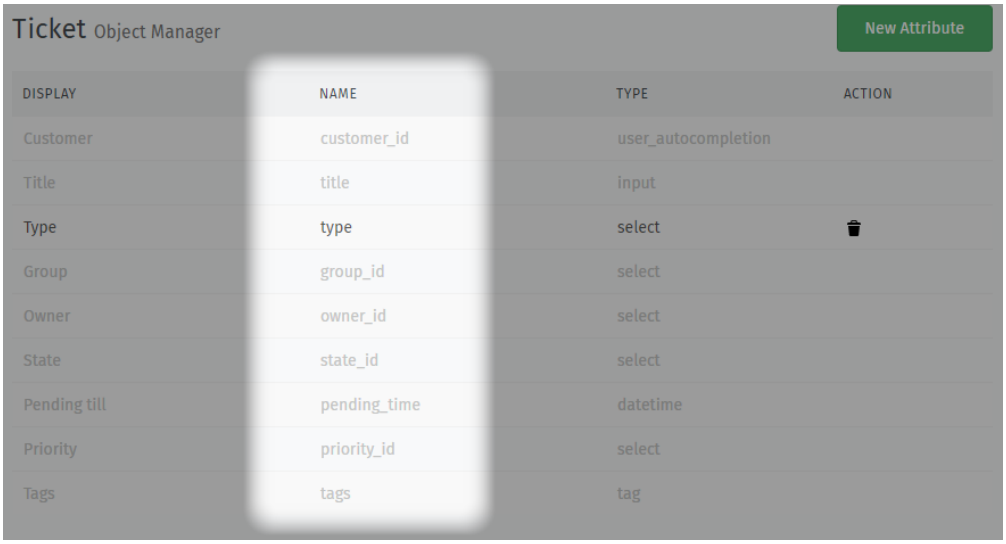


Abb. 7: Find a complete list of ticket attributes in the Object Manager.

- priority

but in practice, you can set almost any attribute, including *custom ones you created through the Object Manager*.

Bemerkung: 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?

Warnung: 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!

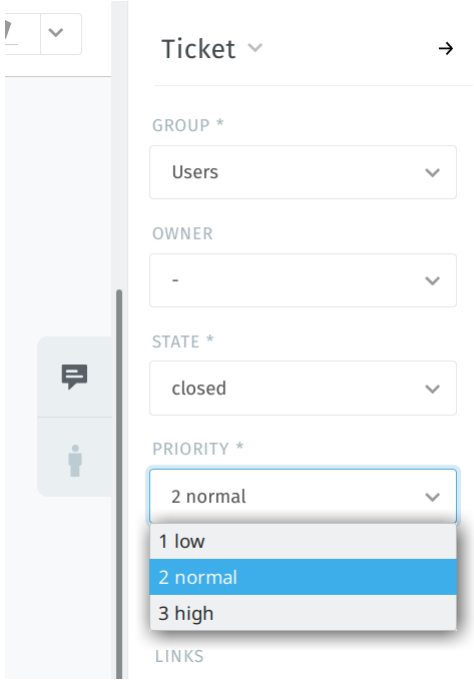
So what values does Zammad understand, then? Well, it depends…

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"
```



Bemerkung: 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:

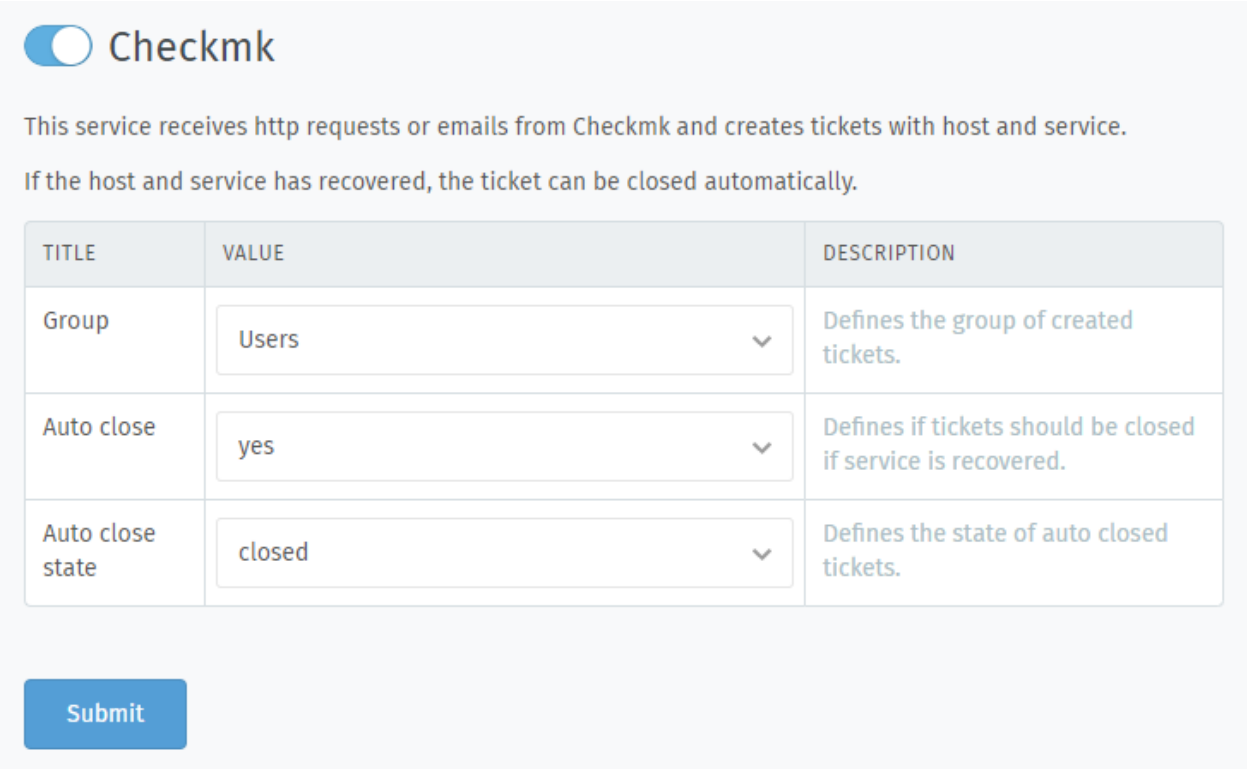
```
# valid
>> Ticket.first.update(note: "You're gonna need a bigger boat")
=> true
>> Ticket.first.note
=> "You're gonna need a bigger boat"

>> 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.

Recent logs

DIRECTION	REQUEST	CREATED AT
in	200 POST https://verification-20200807.zammad.com/api/v1/integration/check_mk/030e4816bec37e5909cca98d6c1beb7c	50 minutes ago
in	200 POST https://verification-	50 minutes ago

29.3.2 Zabbix Integration

This guide describes how to integrate your Zabbix 5.0 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 5.0 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

29.4 Other Integrations

This section will hold any other integration that can't be grouped up (yet).

29.4.1 Slack

Bemerkung: 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

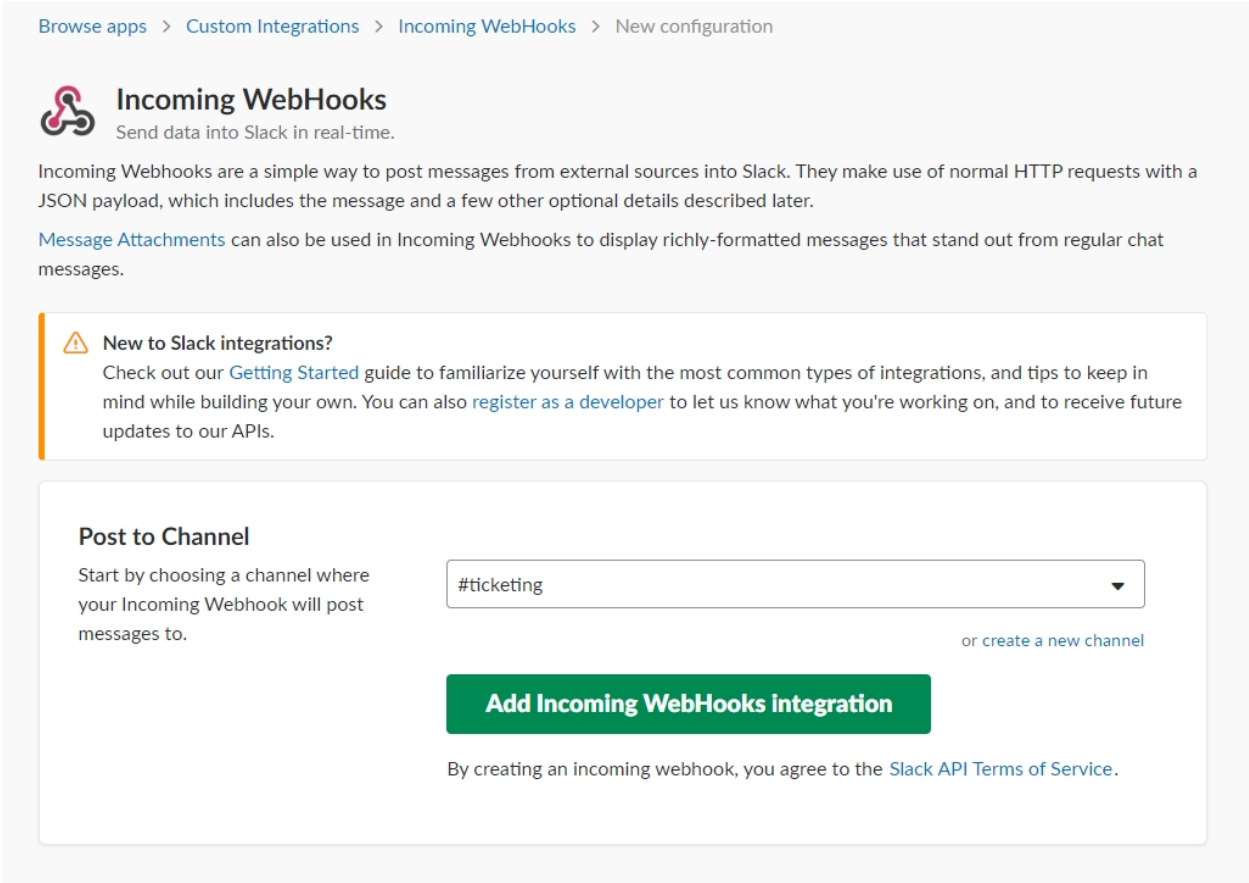
Zammad will provide the Ticket title, a direct link to the Ticket, the event type (creation, updated, escalation), the customer, time and the last article that has triggered the notification. This will give you further options, as you can see e.g. escalating tickets that are assigned to an agent that's e.g. absent. You can interact faster, as you might see problems earlier (or even before the problem gets one).

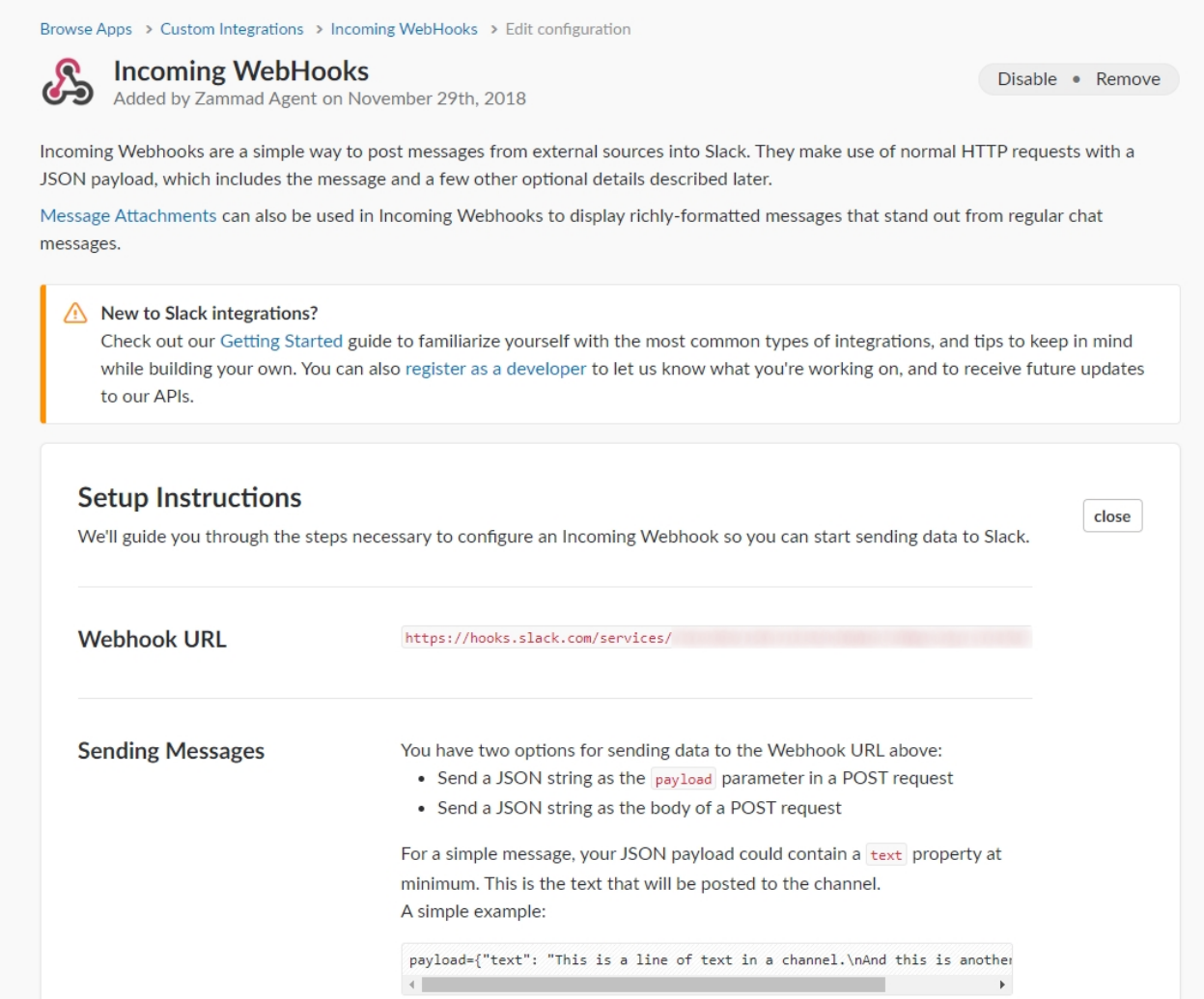
If needed and wanted, you can even discuss directly about the topic before sending an answer to the customer. Another possible use case would be a agent monitoring new agents and their answers to tickets.

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.

Hinweis: You need administrative rights on the Slack Workspace. The link to the app directory is normally `https://[workspace-name].slack.com/apps`.

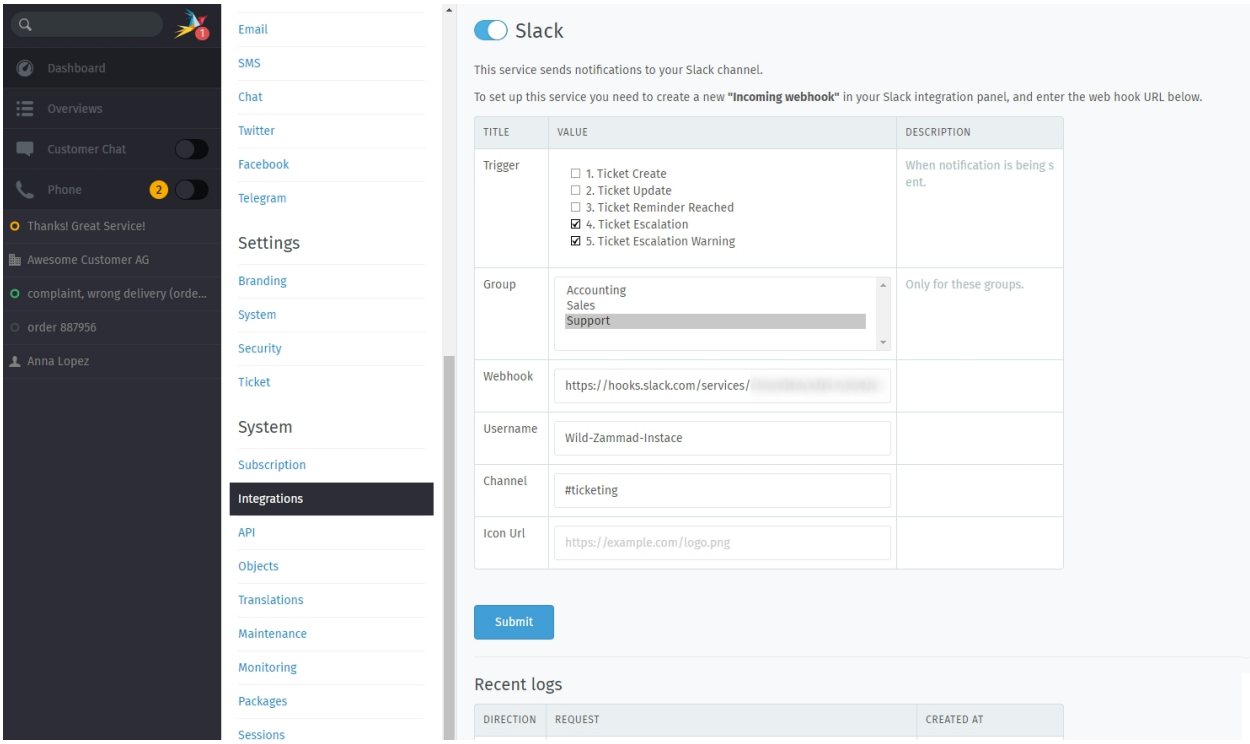

Browse apps > Custom Integrations > Incoming WebHooks > New configuration
Incoming WebHooks
Send data into Slack in real-time.
Incoming Webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details described later.
Message Attachments can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.
New to Slack integrations?
Check out our Getting Started guide to familiarize yourself with the most common types of integrations, and tips to keep in mind while building your own. You can also register as a developer to let us know what you're working on, and to receive future updates to our APIs.
Post to Channel
Start by choosing a channel where your Incoming Webhook will post messages to.
#ticketing
or create a new channel
Add Incoming WebHooks integration
By creating an incoming webhook, you agree to the Slack API Terms of Service.



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.

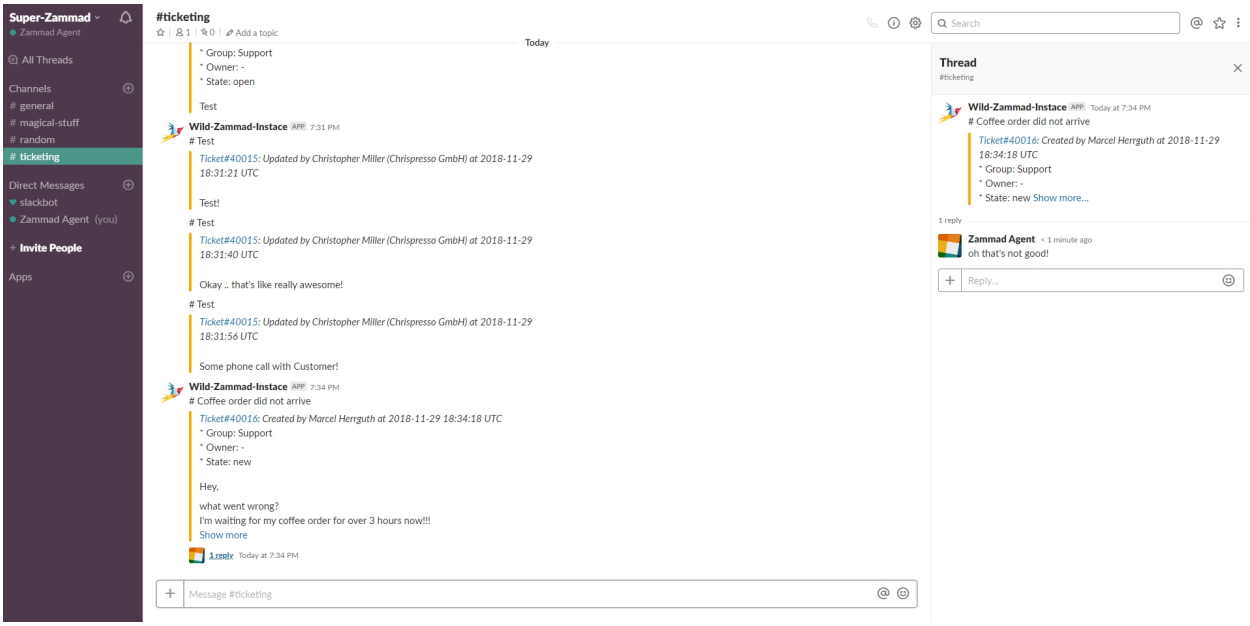
When you're ready, just hit „Submit" and enable the integration. Zammad will now post new ticket information based on the trigger you chose. Below the options you have the recent log that shows the latest requests to Slack for debugging if needed.

Bemerkung: If you leave the icon URL empty, Zammad will use the Zammad logo instead. The icon should be a square PNG file.

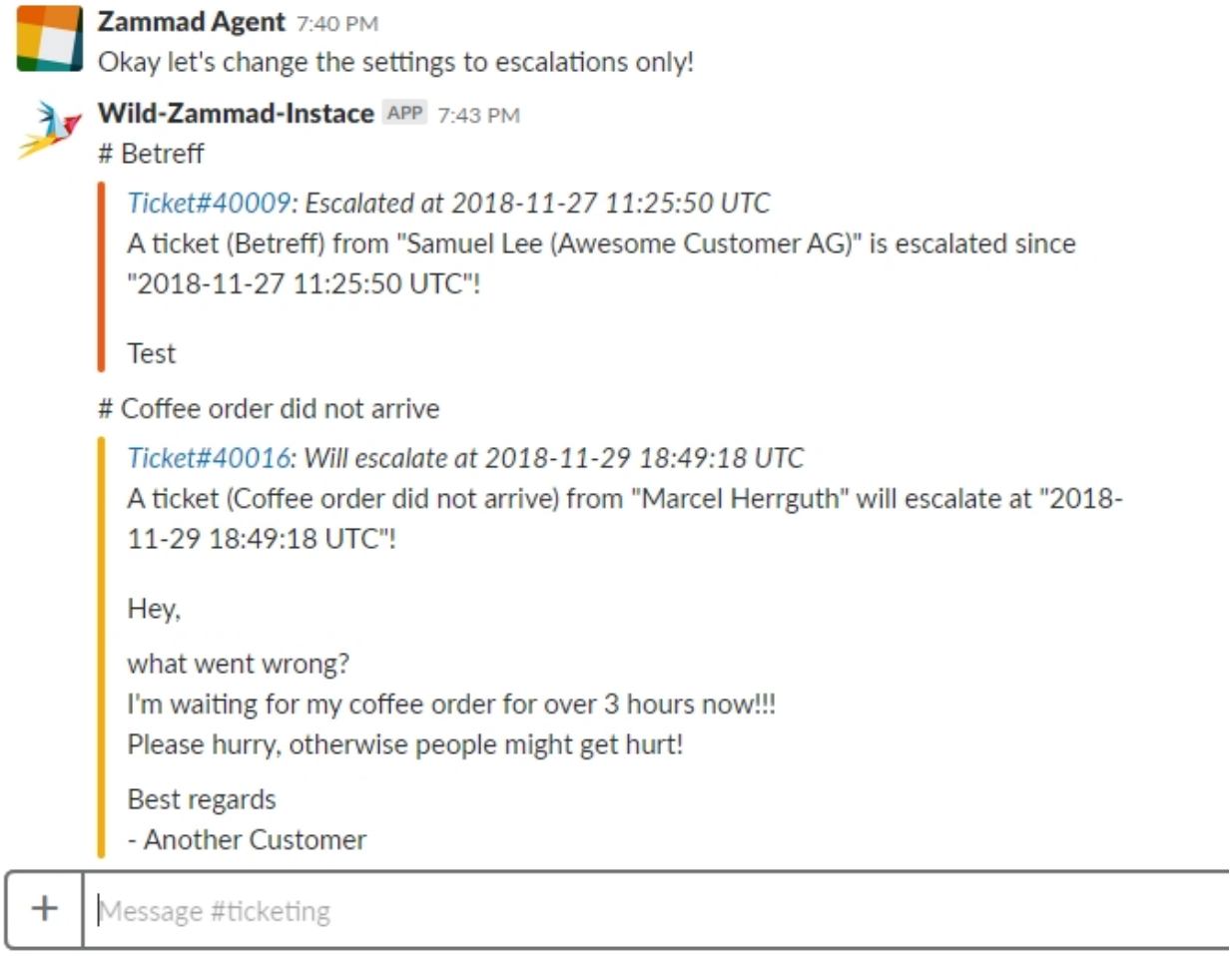


The result

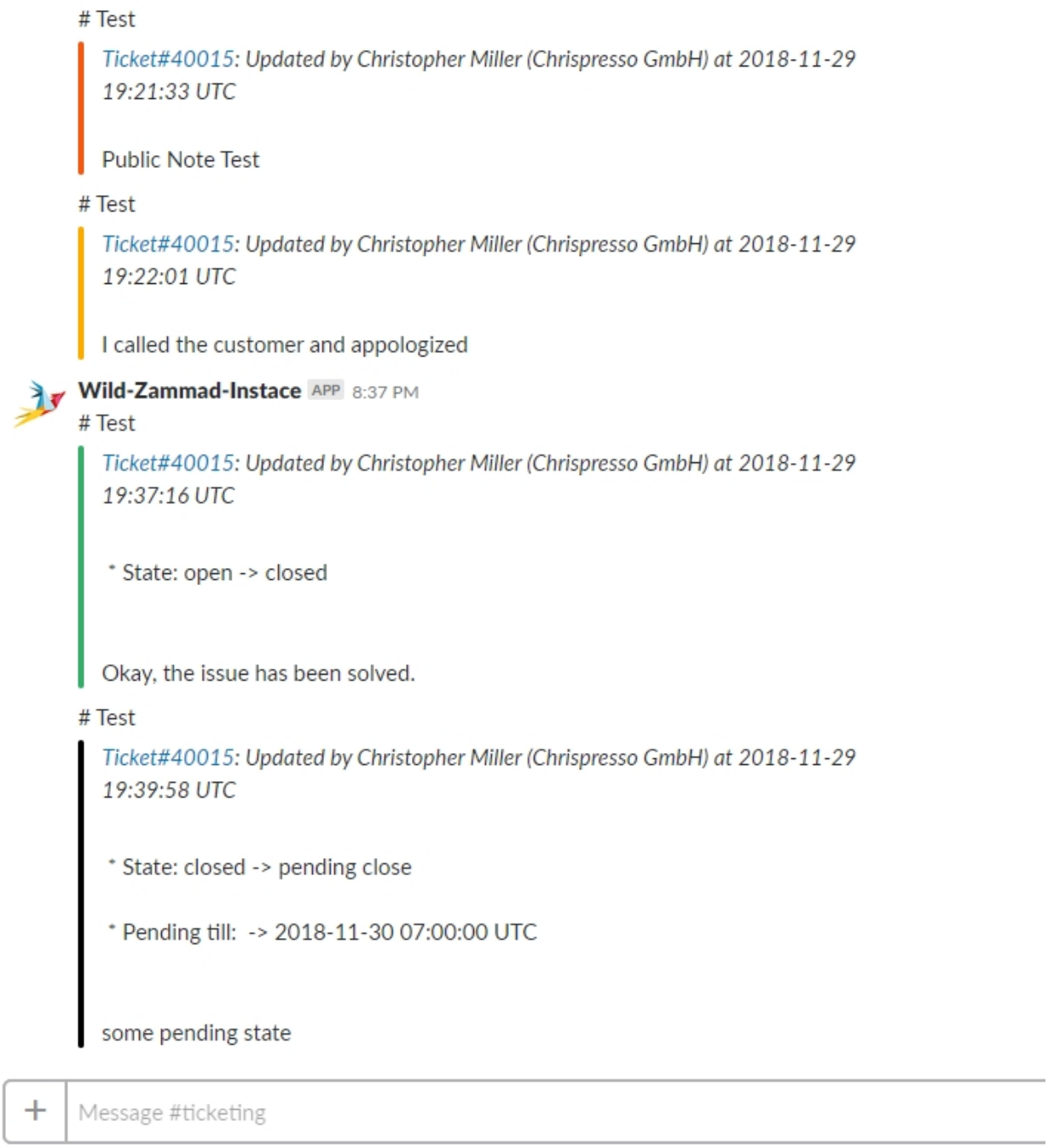
The following figure shows how it will look if you choose to receive updates on created and updated tickets. On every post Zammad sends to the Slack channel, you can create new threads to discuss about the new article.



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!



29.4.2 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

Abb. 8: 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).

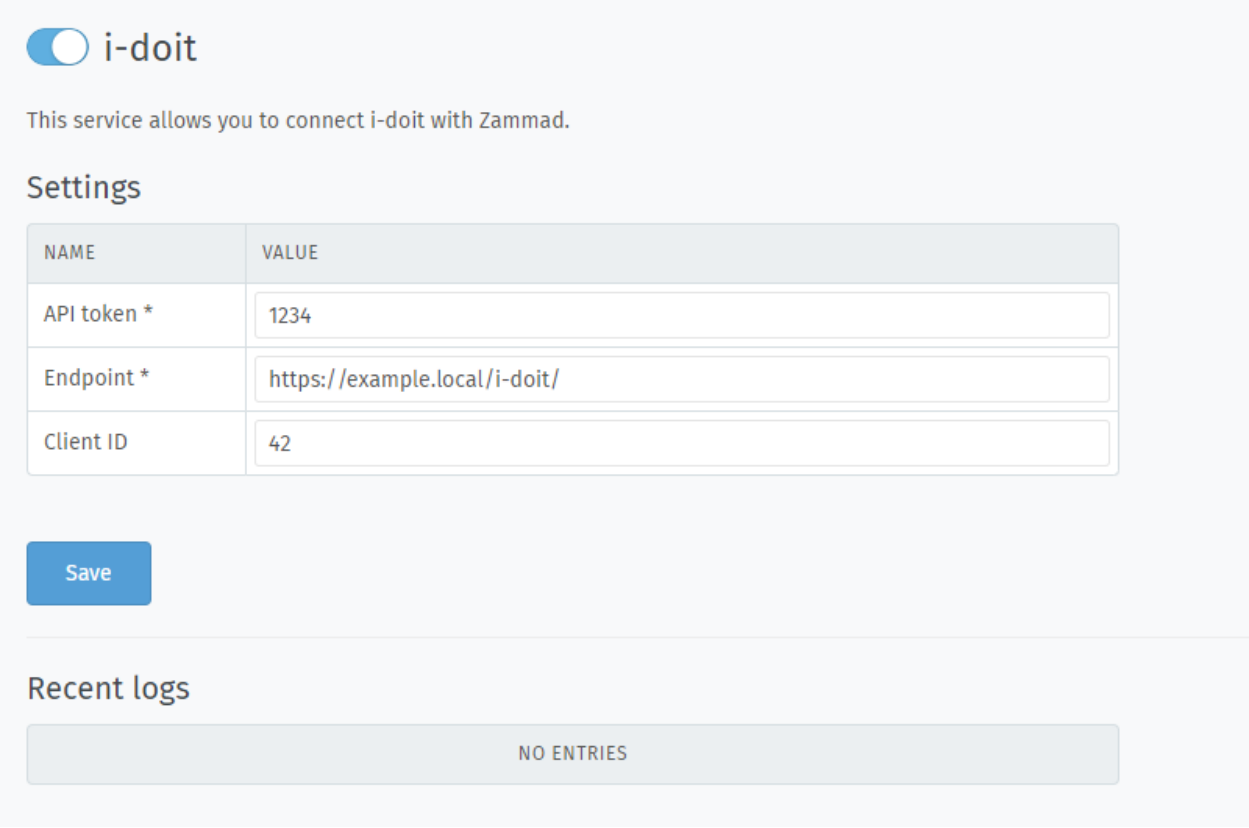
How to set it up

Bemerkung: **Requires i-doit's** API Add-on. Use the following settings:

Active Yes

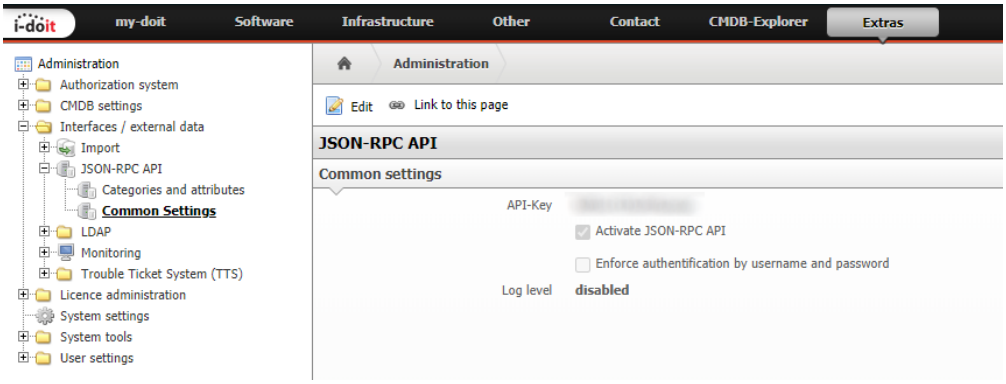
Enforce autentication by username and password No

To set it up, enable the integration in the Zammad admin panel under **System > Integrations > i-doit**:



API token* Found in the i-doit admin panel under **Interfaces / external data > JSON-RPC API > Common Settings**.

Endpoint* The root URL of your i-doit installation.



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!)

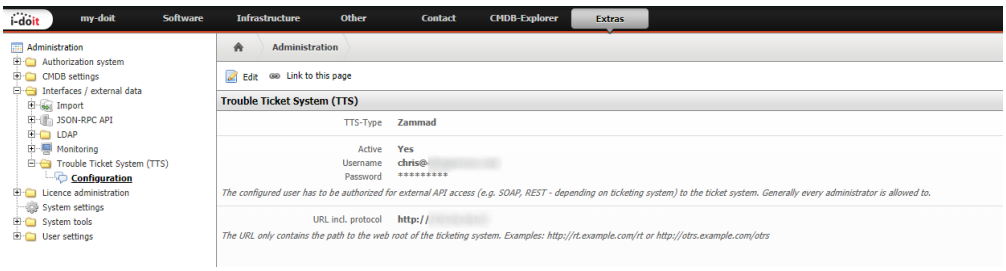
2. List / Create Zammad Tickets in i-doit

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).

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.

> **Bemerkung:** 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.)

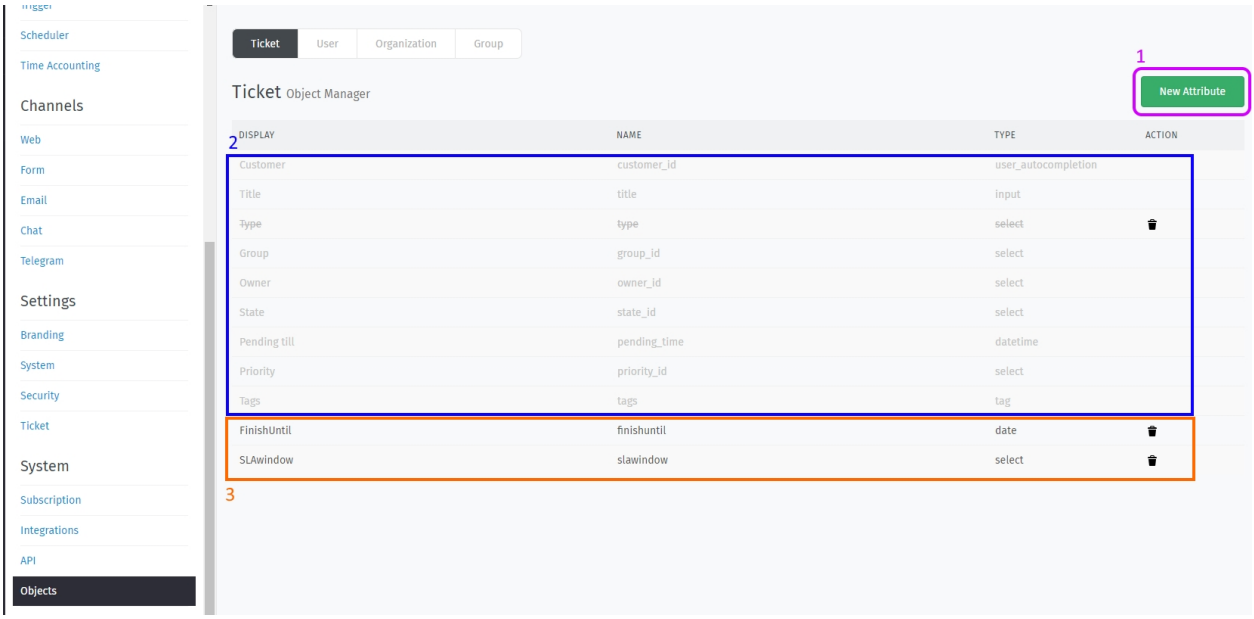
URL incl. protocol `https://your.zammad.domain`

KAPITEL 30

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).

Bemerkung: Try to avoid deleting objects (rather disable them) as Zammad might run into unexpected conditions if they where referenced somewhere.



Here's an overview of the objects. On the upper right you can add new Attributes (no 1). By default, there will be no custom fields - standard objects (no 2) will be grayed out, you can't delete or change those. Custom objects (no 3) will be displayed in black font and have a trash bin on the right site to delete not needed objects. By click on custom objects, you can edit them so they can suite your needs.

30.1 Object types

Hinweis: Starting with Zammad 3.2 you can create dynamic URL-Fields for select and text attributes. See more below!

When adding a new object, you can choose between the following object types:

- **Boolean**
  - true or false, you can adjust the display for those keys and mark a default one.
- **Date**
  - enables you to use the date picker of Zammad.
  - you can allow / forbid the date chosen to be in the future and in past.
  - set the standard time difference in hours, which is entered as the default value if no value is defined.
- **Date-time**
  - enables you to use the date picker plus time selection of Zammad.
  - you can allow / forbid the date chosen to be in the future and in past.
  - set the default time difference in hours.
- **Integer**
  - you can set the default value of this field.
  - you can configure the minimum and maximum value that can be used.
- **Select**
  - you can add as many selections you need, you'll see the field as drop-down menu.
  - they key and display name can differ (display name can be translated if needed).
  - You can select a default value for this field, if you want to.
  - Allows you to define a URL if needed
- **Text**
  - you can enter a default value.
  - you can choose between the following types: email, phone, text, or URL.
  - you can pick the maximum length of the field.
  - Allows you to define a URL if needed
- **Tree Select**
  - this object enables you to use up to 6 sub keys.
  - you'll see this object as a kind of drop down menu.

Bemerkung: You cannot change the object format / type as soon as you applied it. If you don't further need an object, you can disable it.

30.2 URL fields (Link-Template)

Bemerkung: This option is available within Zammad 3.2 and later. This function is restricted to Text and Select objects only.

Link-Templates are an amazing way to dynamically generate URLs. This allows you to integrate other systems better, without having the need to manually copy an URL! As soon as field is set and updated, a URL-Icon will appear, allowing you to open the link in a new tab.

Hinweis: Even though the Link-Template field shows up within the object edition, it is optional and not used if you don't fill in anything.

30.2.1 How does this work. . . ?!

The Link-Template can consist of static and dynamic parts. This means that you can, for exmaple, create a URL for a google search by defining the following:

```
https://www.google.com/search?q=#{ticket.amazingobject}
```

The above will create a google search link which will contain the search phrase (which is the objects value). You can use any variable that's available within Zammad! This gives your great options within your structures!

Learn more about *Variables*.

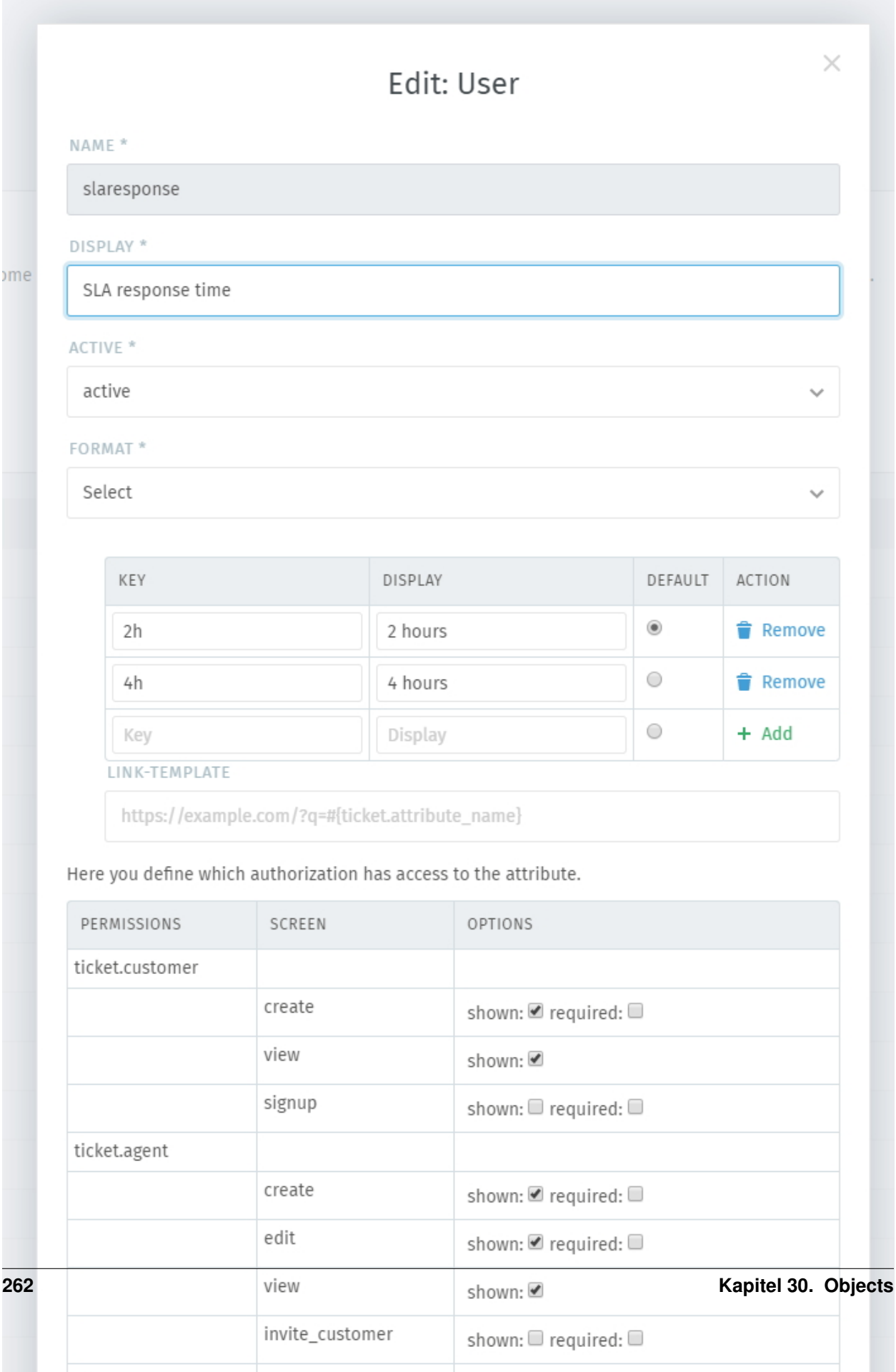
Abb. 1: The above screencast shows how the link template will perform after object creation.

30.3 Object permissions

When ever needed you can restrict access to objects for permission roles (agents and customer) and even set a object to be required.

- **During Ticket Creation**
  - show / hide the field.
  - make the field required (if needed).
- **During Updating a Ticket**
  - show / hide the field.
  - make the field required (if needed).

Hinweis: You can't change these settings for pre defined objects (as you can't edit them via UI). Please note that this currently works as designed.


Edit: User
NAME *
slaresponse
DISPLAY *
SLA response time
ACTIVE *
active
FORMAT *
Select
KEY
DISPLAY
DEFAULT
ACTION
2h
2 hours
Remove
4h
4 hours
Remove
Key
Display
Add
LINK-TEMPLATE
https://example.com/?q=#{ticket.attribute_name}
Here you define which authorization has access to the attribute.
PERMISSIONS
SCREEN
OPTIONS
ticket.customer
create
shown: required:
view
shown:
signup
shown: required:
ticket.agent
create
shown: required:
edit
shown: required:
view
shown:
invite_customer
shown: required:

30.4 Updating database after adding or editing objects

Hinweis: Zammad doesn't restart automatically and prompts you to restart? You may want to check configure enviroment variables to solve this.

Bemerkung: If you're on a self hosted system, you might need console access to your Zammad-Server to restart the service.

When adding or changing objects, Zammad will not apply the changes isntantly, 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".

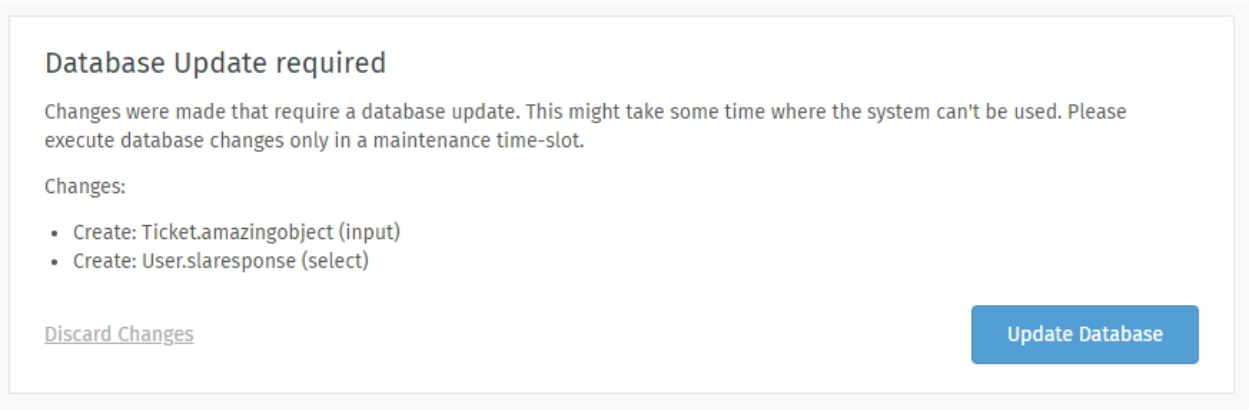


Abb. 3: Changes on objects require you to update the database to apply these changes.

KAPITEL 31

Variables

Bemerkung: 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.

Hinweis: 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!).

31.1 Variable Categories

31.1.1 Config

Bemerkung: 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.

Tab. 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

31.1.2 Current User

Bemerkung: 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.

Bemerkung: 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.

Tab. 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

Fortsetzung auf der nächsten Seite

Tab. 2 – Fortsetzung der vorherigen Seite

name	variable	example
Current User > Updated by > Login	`#{user.updated_by.login}`	`jdoe`
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`

Fortsetzung auf der nächsten Seite

Tab. 2 – Fortsetzung der vorherigen Seite

name	variable	example
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
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

31.1.3 Articles

Bemerkung: 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.

Tab. 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`

Fortsetzung auf der nächsten Seite

Tab. 3 – Fortsetzung der vorherigen Seite

name	variable	example
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
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

Fortsetzung auf der nächsten Seite

Tab. 3 – Fortsetzung der vorherigen Seite

name	variable	example
Ticket > Article#	`#{ticket.article_count}`	`1` number of ticket articles

31.1.4 Ticket

Bemerkung: 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.

Tab. 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

Fortsetzung auf der nächsten Seite

Tab. 4 – Fortsetzung der vorherigen Seite

name	variable	example
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)
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`

Fortsetzung auf der nächsten Seite

Tab. 4 – Fortsetzung der vorherigen Seite

name	variable	example
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, ...`
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

Fortsetzung auf der nächsten Seite

Tab. 4 – Fortsetzung der vorherigen Seite

name	variable	example
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
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

Fortsetzung auf der nächsten Seite

Tab. 4 – Fortsetzung der vorherigen Seite

name	variable	example
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

31.2 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}`.

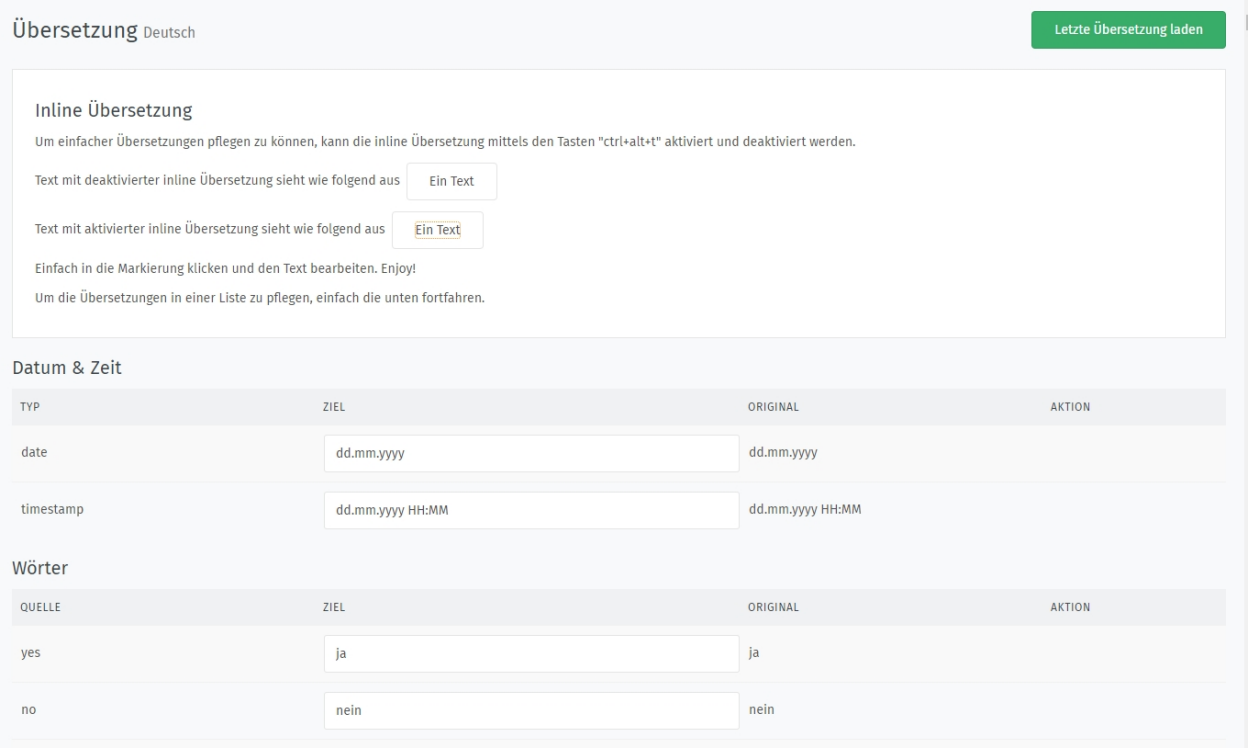
Now, in some situations it's possible that you'll need to use a third part. The best example for such a situation would be a `select` or `tree-select` field which will by default return the key value, not it's display name. For this, just extend your variable with `.value`. This will result in `#{ticket.select.value}`.

KAPITEL 32

Translations

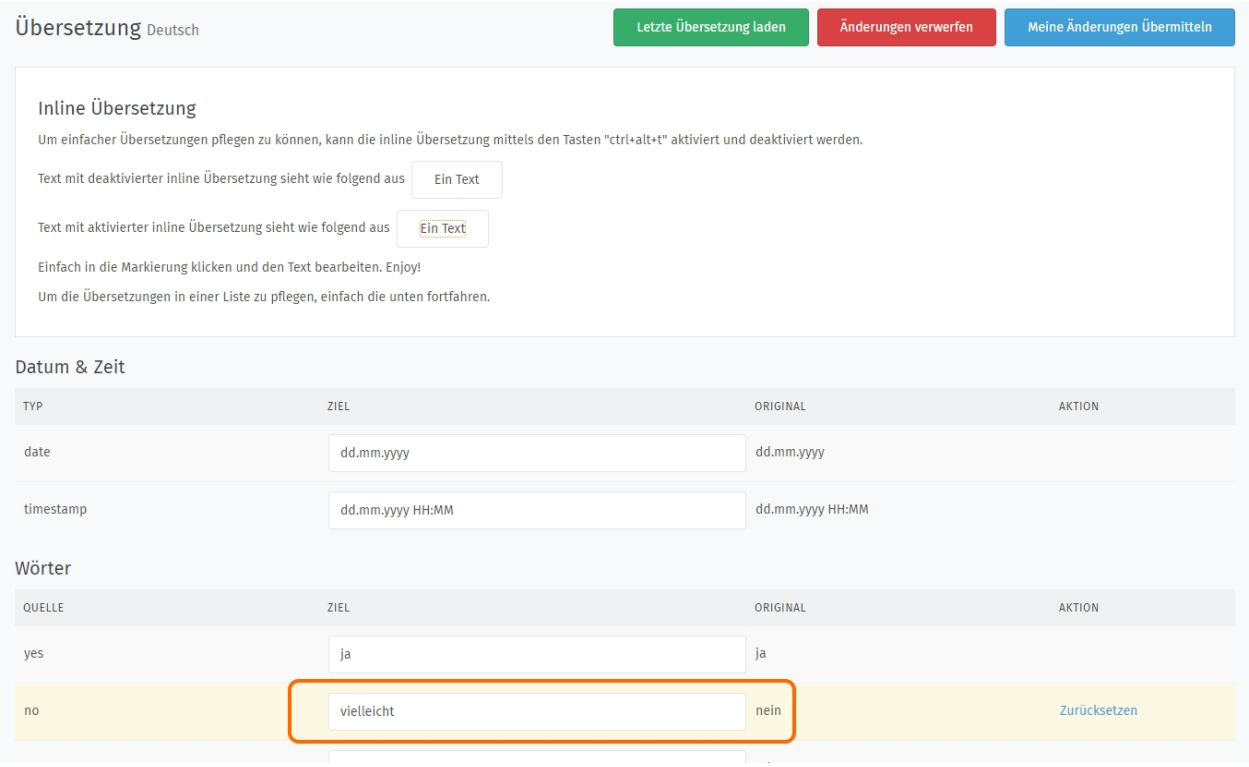
In Zammad (as admin) you can translate strings having typos or even not being translated. You can always „Get last translations“ to ensure that you’re up to date before doing that.

Note: The language for translation is always the language you chose within your profile. As Zammad’s main language is English, you can’t translate into English.



Translated strings are highlighted within Zammad. If you’ve translated strings, you have two options: You can send your changes to our central system, this will share your translation with other users. This is a great contribution.

If you don't like your changes or want to start over, you can discard your changes at any time.



KAPITEL 33

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**.

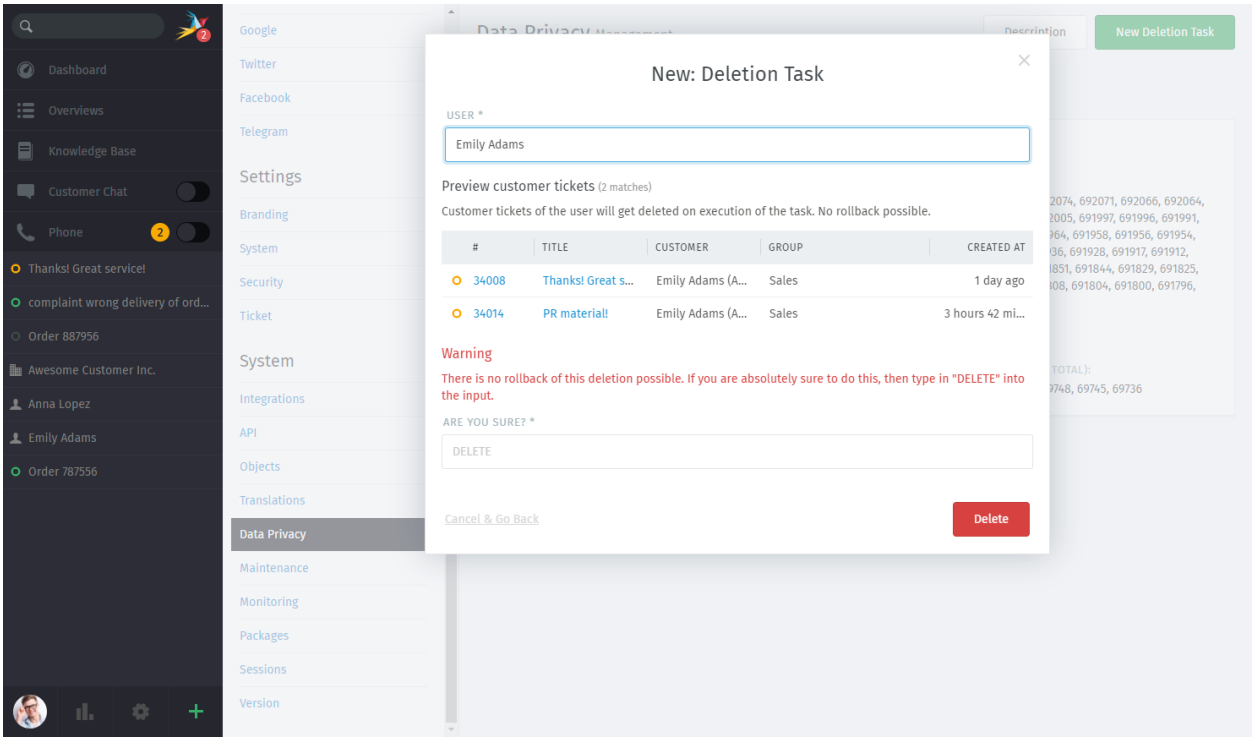


Abb. 1: The user deletion dialog lists some of the tickets that will be removed from the system along with the user.

Bemerkung: 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.

33.1 Deleting Users

Warnung: 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.

Bemerkung: The following records cannot be deleted:

- your own account
- the system's last remaining administrator account

33.1.1 Step 1: Find a user / confirm deletion

There are three ways to access the user deletion dialog:

from the user's profile

Abb. 2: Click **Action > Delete**.

in the "Manage > Users" Admin Panel

Abb. 3: Use the **Actions** menu for the target user.

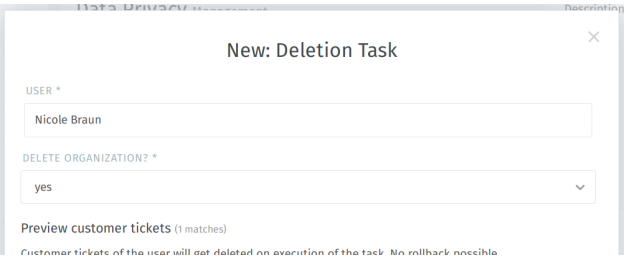
in the "System > Data Privacy" Admin Panel

Abb. 4: Use the **New Deletion Task** button. Search for users by name or email address.

Hinweis: 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:

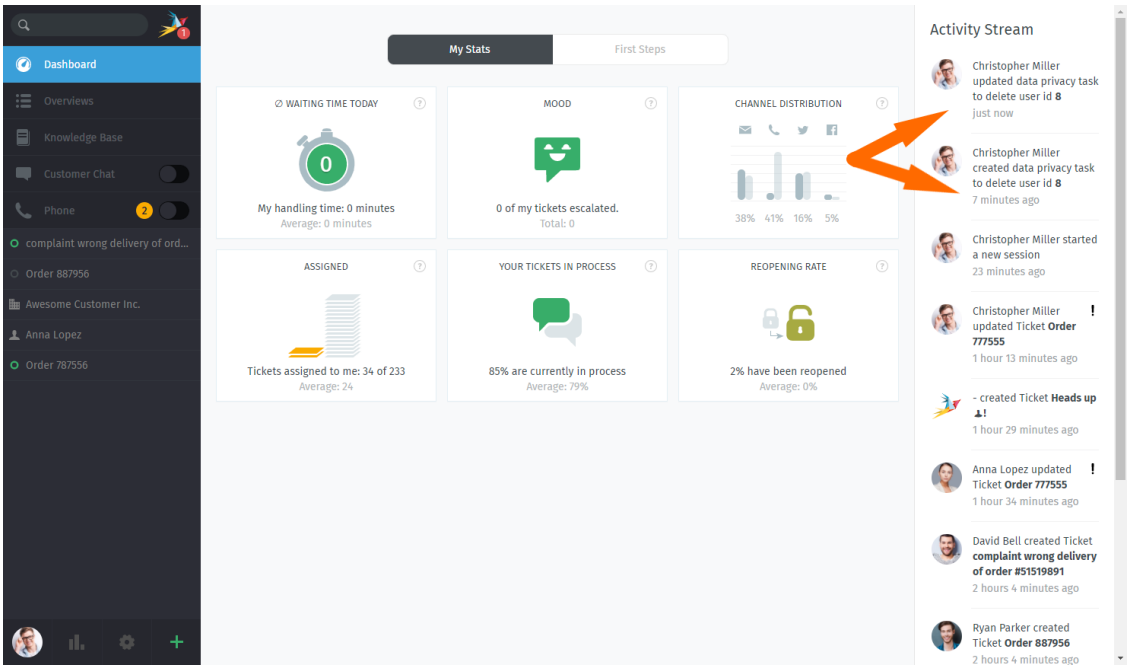
(If this option does not appear, make sure there are no pending deletion tasks for other customers from this organization.)



33.1.2 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



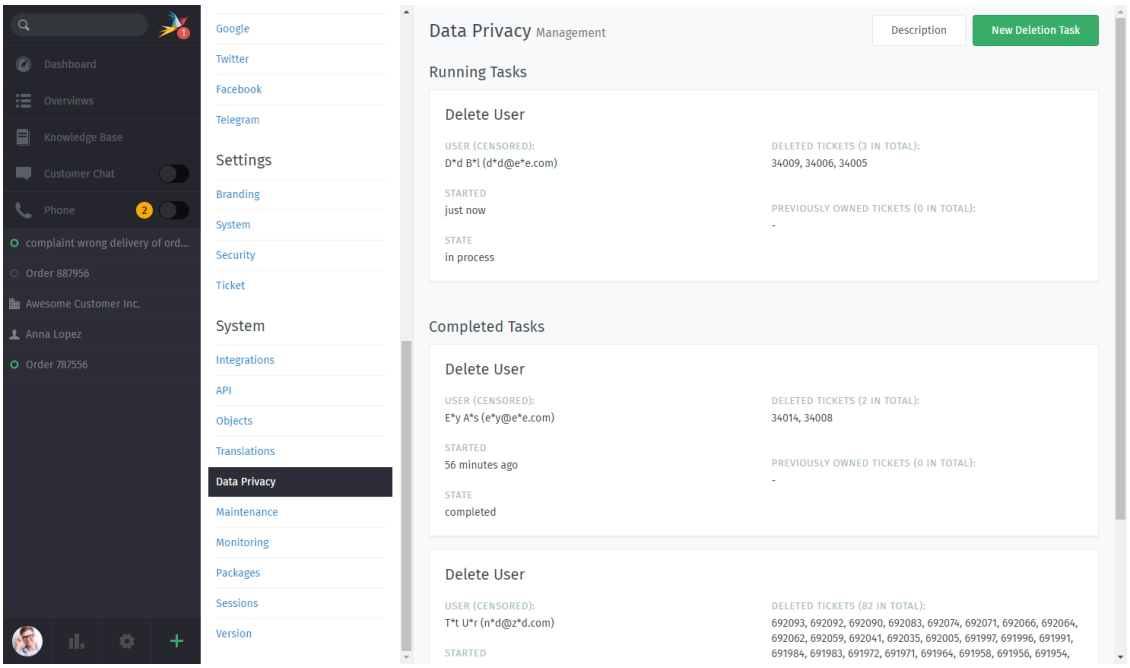
For each deleted user, the Activity Stream will be updated twice—once when the task is created, and once when it's complete.

Hinweis: These notifications are only visible to users with `admin.data_privacy` permissions.

in the "System > Data Privacy" Admin Panel

33.2 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.

KAPITEL 34

Maintenance

If maintenance work on the system is necessary, a number of preparations must be made. You can take care of them here.

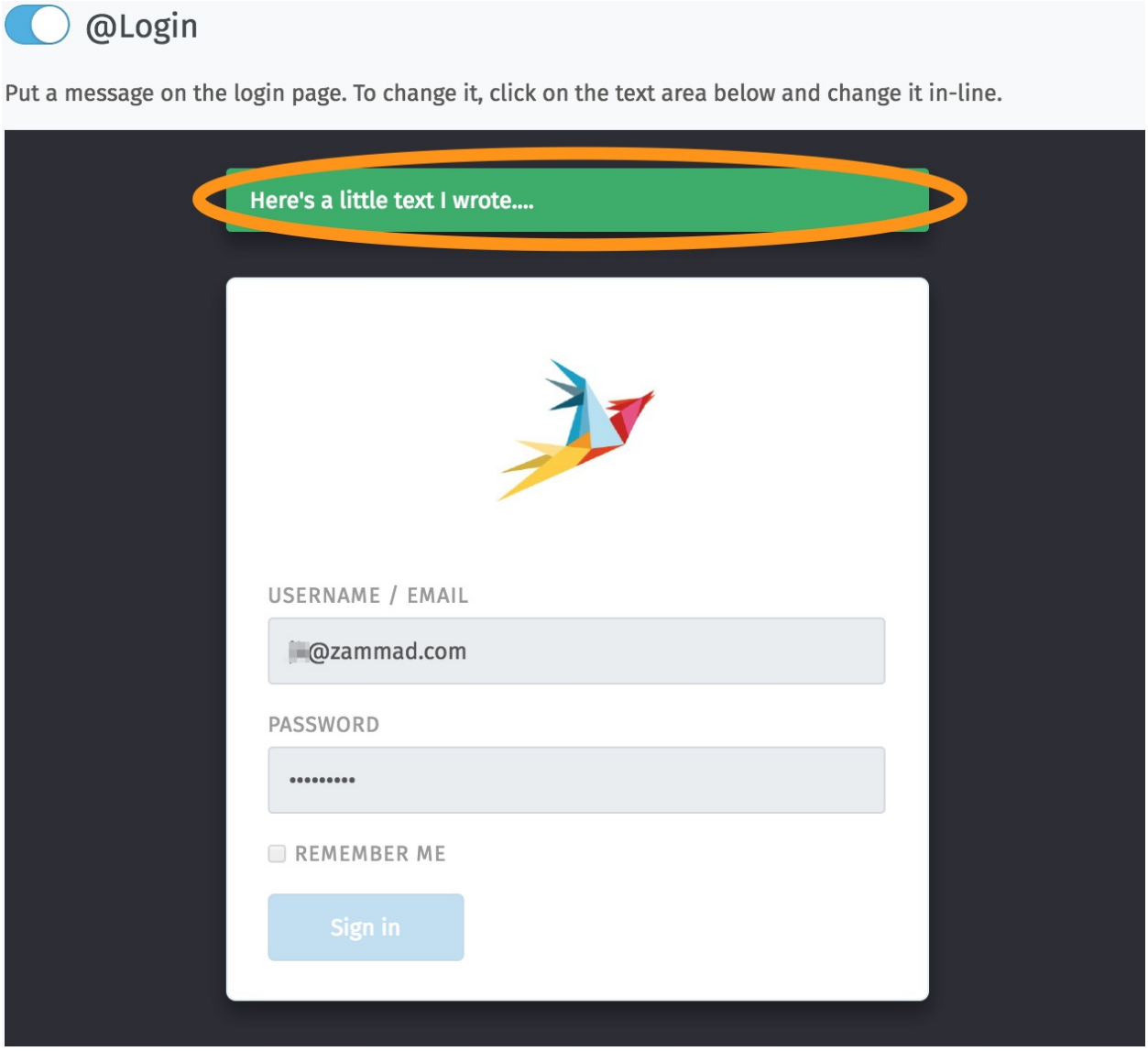
34.1 Mode

Enable or disable the maintenance mode of Zammad. If enabled, all non-administrators get logged out and only administrators can start a new session.

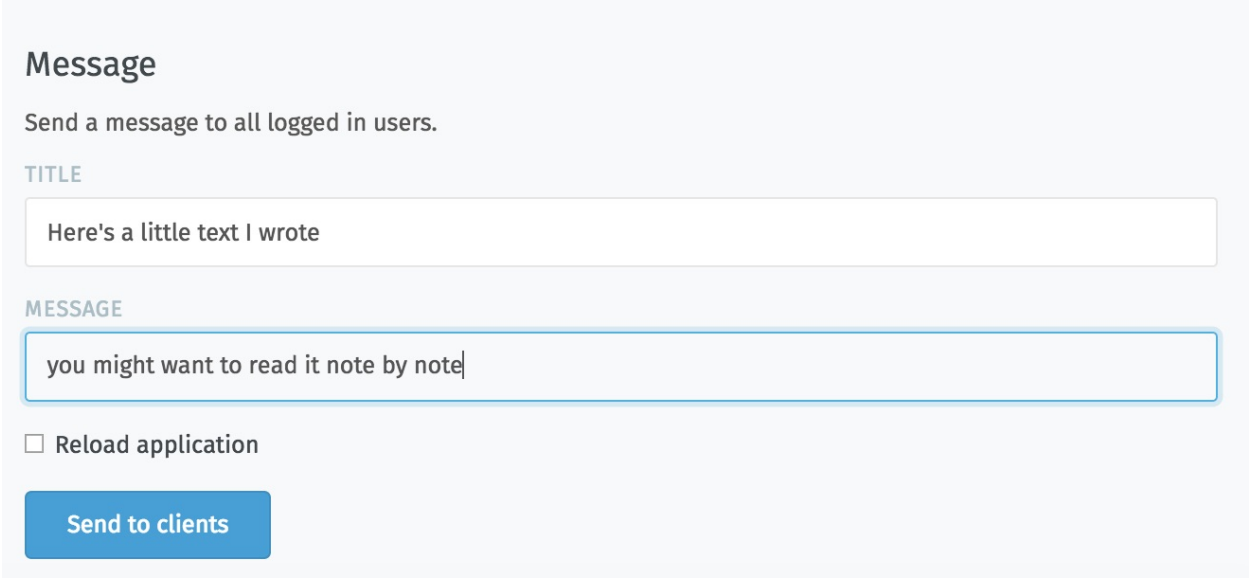
Before the maintenance mode is activated, you can not only inform about it on the login page…

34.2 @Login

You can place a short message for all users on the login page. This will then be displayed as follows:



But you can also inform all agents with a message about something:


Message
Send a message to all logged in users.
TITLE
Here's a little text I wrote
MESSAGE
you might want to read it note by note
Reload application
Send to clients

KAPITEL 35

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\
↪"=\XXXXXXXXXXXXXXXXX, \"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

KAPITEL 36

Packages

That's the package management-area.

Individual add-ons for Zammad can be installed and managed here.

KAPITEL 37

Sessions

Shows who logged on to the Zammad instance as a user, when and from where.

KAPITEL 38

Version

Shows which version is currently being used on your Zammad-instance.

KAPITEL 39

Composer Settings

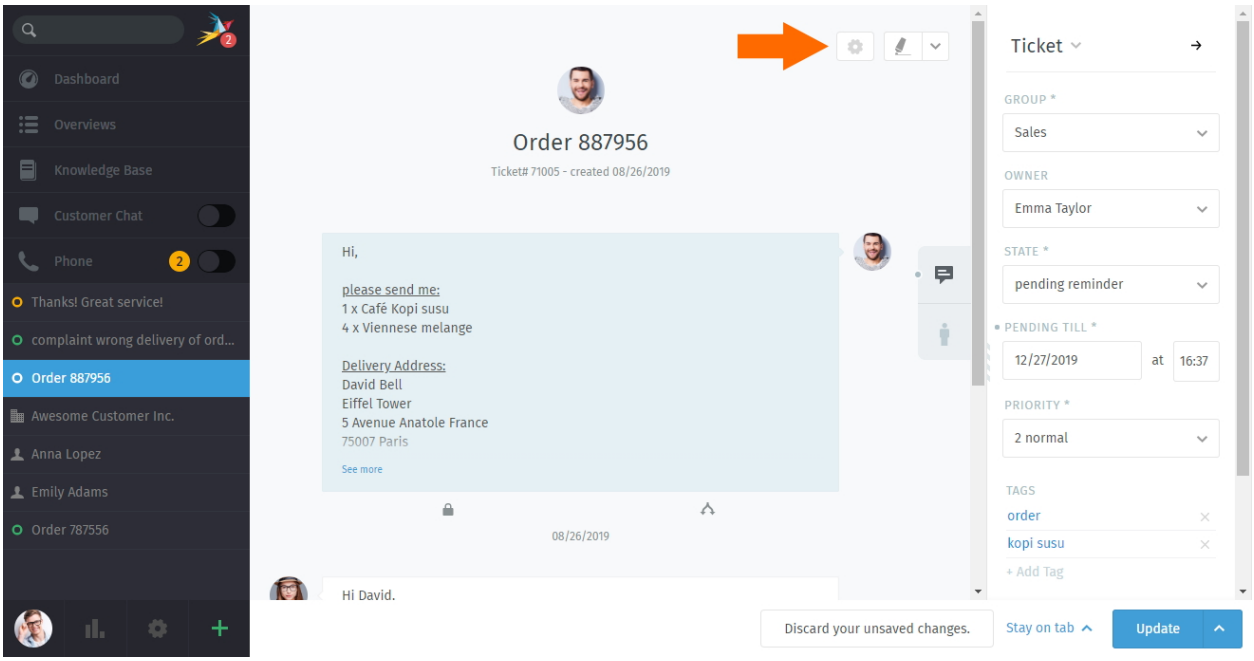
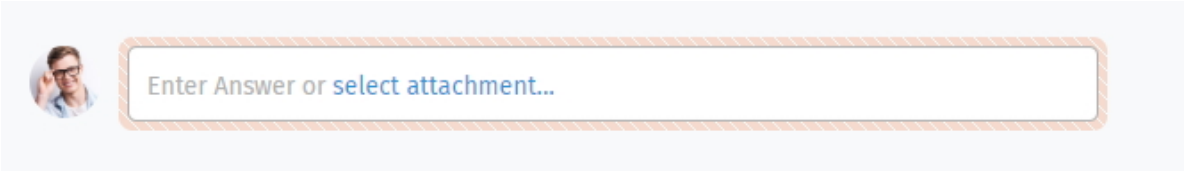


Abb. 1: Use the Composer Settings to change the behavior of the new message editor.

Bemerkung: 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.



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.

> **Warnung:** 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.

> **Bemerkung:** 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 lastname) to the bottom of every tweet answer you create. This only affects tickets that come from the Twitter Channel.