

Administration - CLI

- [Deaktivierte Benutzer in Zammad über Skript löschen](#)
- [Zammad Status umbenennen](#)
- [Zammad Ticket-Status Optionen anzeigen](#)
- [Zammad E-Mails an Kunden versenden ohne SMTP-Konto](#)
- [Zammad Ticket-Status hinzufügen](#)
- [Zammad Ticket-StatusTypen anzeigen](#)
- [Zammad Übersetzung löschen](#)

Deaktivierte Benutzer in Zammad über Skript löschen

Einleitung

In diesem Beitrag erkläre ich kurz, wie wir in **Zammad** automatisiert alle **deaktivierten Benutzer** löschen können. Dafür verwenden wir ein Skript, welches die Benutzer dann löscht.

Achtung: Bei diesem Vorgang können auch aktive Benutzer gelöscht werden! Ich übernehme keine Gewähr oder Haftung für Verluste!

Benutzer löschen

Um die Benutzer zu löschen, melden wir uns zuerst über SSH auf unserem **Zammad** Server an. Dann öffnen wir mit dem folgenden Befehl die **Rails Shell**.

```
# Package Installation
zammad run rails c

# Source Installation
rails c
```

Und dort geben wir den folgenden Befehl ein:

```
User.where(active: false).each {|user|
  next if user.id == 1
  user.destroy
}
```

Zammad Status umbenennen

Einleitung

In dieser kleinen Anleitung beschreibe ich kurz, wie in der "*rails Shell*" einen Ticket-Status von *Zammad* umbenennen können. Das benötigen wir z.B. wenn wir einen Status angelegt haben, aber der Name sich verändert hat oder sich ändern soll.

Namen ändern

Um den Namen zu ändern, müssen wir im ersten Schritt die "*rails Shell*" öffnen. Dies können wir mit dem nachstehenden Befehl durchführen:

```
zammad run rails c
```

Info: Falls wir uns jetzt die vorhandenen Ticket-Status Optionen anzeigen lassen wollen, setzen wir den folgenden Befehl ab: `Ticket::State.pluck(:id, :name)`

Wenn wir jetzt den Namen ändern möchten, benötigen wir den Original-Namen des Ticket-Status. Diesen erhalten wir mit dem oben angegebenen Befehl.

Jetzt verwenden wir den unten angegebenen Befehl, um den Namen zu ändern. Dabei geben wir im ersten String den aktuellen Namen an, und im zweiten den Namen an, wie der Status in Zukunft heißen soll.

```
Ticket::State.find_by(name: 'alter_name').update(name: 'Neuer Name')
```

Wenn wir jetzt die Seite neu laden, sollte der Status-Name aktualisiert sein.

Zammad Ticket-Status Optionen anzeigen

Einleitung

In diesem kleinen Artikel beschreibe ich kurz, wie wir in der "Rails Shell" von Zammad uns die verschiedenen Ticket-Status Optionen anzeigen lassen können. Diese benötigen wir, um z.B. weitere Modifikationen vorzunehmen.

Ticket-Status Optionen anzeigen

Um die verschiedenen Optionen anzeigen zu lassen, müssen wir im ersten Schritt die "*Rails Shell*" öffnen. Dazu geben wir den folgenden Befehl ein:

```
zammad run rails c
```

Danach geben wir den nächsten Befehl ein. Dieser gibt uns den original Namen und die ID unserer Ticket-Status Optionen an. Mit diesen Informationen könnten wir dann weiter arbeiten, falls wir z.B. Übersetzungen, etc. hinzufügen möchten.

```
Ticket::State.pluck(:id, :name)
```

Wir erhalten dann eine Ausgabe, die wie folgt aussehen kann:

```
=>
[[1, "new"],
 [2, "open"],
 [3, "pending reminder"],
 [4, "closed"],
 [5, "merged"],
 [6, "removed"],
 [7, "pending close"]]
```

Zammad E-Mails an Kunden versenden ohne SMTP-Konto

Einleitung

Bei der Konfiguration des *Zammad Servers* bin ich letztens auf den Wunsch gestoßen, E-Mails zu versenden, ohne zwangsweise ein **SMTP-Konto** zu erstellen.

Konto hinzufügen

Um jetzt das **Mail-Konto** hinzuzufügen, melden wir uns mit der Konsole bspw. über *SSH* auf unserem **Zammad-Server** an. Dort öffnen wir die **Rails-Shell** oder geben den folgenden Befehl im Ganzen ein.

```
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)"
```

Jetzt muss man sich einmal neu anmelden, dann können wir über die `hinzufügen` Schaltfläche neue E-Mail-Adressen hinzufügen. Die E-Mails werden dann über `sendmail` versendet.

Zammad Ticket-Status hinzufügen

Einleitung

In diesem Artikel beschreibe ich kurz, wie wir in *Zammad* einen neuen Status hinzufügen können. Dies können wir verwenden, falls uns die vordefinierten Status-Optionen nicht ausreichen. Diese Änderung können wir aber nur in der "Rails Shell" erledigen. Es gibt hier aktuell noch keine Möglichkeit, einen Status über die Web-Oberfläche hinzuzufügen.

Durchführung

Neuen Status anlegen

Im ersten Schritt müssen wir natürlich die "*Rails Shell*" öffnen. Dazu geben wir den folgenden Befehl ein:

```
zammad run rails c
```

Jetzt können wir den unten stehenden Befehl verwenden, um den Ticket-Status in der Datenbank anzulegen. Dabei ändern wir Parameter wie z.B. den Namen, auf den wir gerne hätten (vorzugsweise auf Englisch). Den Status ändern wir auch auf die Option, die wir benötigen.

```
Ticket::State.create_or_update(  
  name: 'in progress',  
  state_type: Ticket::StateType.find_by(name: 'open' ),  
  ignore_escalation: false,  
  created_by_id: 1,  
  updated_by_id: 1,  
)
```

Status in der Web-Oberfläche sichtbar machen

Um jetzt den Status in der Web-Oberfläche sichtbar zu machen, geben wir den folgenden Befehl ein:

```
attribute = ObjectManager::Attribute.get(  
  object: 'Ticket',  
  name: 'state_id',
```

```

)
attribute.data_option[:filter] = Ticket::State.by_category(:viewable).pluck(:id)
attribute.screens[:create_middle]['ticket.agent'][:filter] =
Ticket::State.by_category(:viewable_agent_new).pluck(:id)
attribute.screens[:create_middle]['ticket.customer'][:filter] =
Ticket::State.by_category(:viewable_customer_new).pluck(:id)
attribute.screens[:edit]['ticket.agent'][:filter] =
Ticket::State.by_category(:viewable_agent_edit).pluck(:id)
attribute.screens[:edit]['ticket.customer'][:filter] =
Ticket::State.by_category(:viewable_customer_edit).pluck(:id)
attribute.save!

```

Wenn wir jetzt die Web-Oberfläche neu öffnen oder laden, sollten die neuen Status-Optionen verfügbar sein.

Übersetzung hinzufügen

Wenn wir jetzt, wie *Zammad* vorschlägt, dem Ticket-Status einen englischen Namen gegeben haben, können wir eine Übersetzung hinzufügen, damit in der Web-Oberfläche der Status auch in Deutsch angezeigt werden kann. Dazu verwenden wir den folgenden Befehl und verändern nur die "*Namen-Strings*".

```

Translation.create_if_not_exists(:locale => 'de-de', :source => "in progress", :target => "in
Arbeit" , created_by_id: 1, updated_by_id: 1)

```

Zammad Ticket-StatusTypen anzeigen

Einleitung

In diesem Artikel beschreibe ich kurz, wie wir in *Zammad* die *Ticket-StatusTypen* uns anzeigen lassen können.

Durchführung

Im ersten Schritt müssen wir auch hier die "*Rails Shell*" öffnen. Dafür verwenden wir den unten stehenden Befehl:

```
zammad run rails c
```

Jetzt können wir den nachstehenden Befehl eingeben und wir erhalten eine Ausgabe mit den verfügbaren *Status-Typen*.

```
Ticket::StateType.pluck(:id, :name)
```

Wir erhalten dann eine Ausgabe, die wie folgt aussehen kann:

```
=>
[[1, "new"],
 [2, "open"],
 [3, "pending reminder"],
 [4, "pending action"],
 [5, "closed"],
 [6, "merged"],
 [7, "removed"]]
```


Zammad Übersetzung löschen

Einleitung

In diesem kleinen Artikel beschreibe ich kurz, wie wir in der *"Rails Shell"* von *Zammad* eine Übersetzung löschen können. Dies muss gemacht werden, falls eine Übersetzung in unserem Datenbestand fehlerhaft ist. Leider können wir diesen Vorgang zum jetzigen Zeitpunkt nur über die *"Rails Shell"* realisieren.

Durchführung

Im ersten Schritt müssen wir uns auch hier über den folgenden Befehl in der *"Zammad-CLI"* anmelden.

```
zammad run rails c
```

Wenn wir hier angekommen sind, können wir dann mit dem folgenden Befehl die Übersetzung löschen:

```
Translation.find_by(id: <id>).delete
```

ID Herausfinden

Falls wir die ID unserer Übersetzung nicht haben, können wir alle Übersetzungen mit dem folgenden Befehl abfragen. Dann sehen wir alle Übersetzungen und können uns die entsprechende Übersetzung heraussuchen.

```
Translation.pluck(:id, :source, :target)
```