

OTRS und kivitendo – Tickets und Stammdaten takten synchron



OTRS Open Technology
Real Services



OTRS Stammdaten



Kivitendo Kunden und Ansprechpartner

==> als Datenquelle im OTRS

Anwender-Sicht OTRS



Kundenname | Kundennummer

A screenshot of the OTRS user interface. The main window is titled "Create New Email Ticket" and contains several input fields: "From queue:", "To customer user:", "Cc:", "Bcc:", "CustomerID:", "Owner:", "Subject:", and "Options:". A search dropdown is open, showing a list of customer names and their corresponding IDs. The search term "Medi" is entered in the dropdown's input field. The list includes: "Media Saturn Holding GmbH 1000387", "MediMax Zentrale Electronic GmbH 1000515", "Media Markt TV-HiFi-Elektro GmbH Wiesbaden-Hasengarten 1000853", "Media Markt TV-HiFi-Elektro GmbH Hamburg-Nedderfeld, Store M 1000894", "Media-Saturn Deutschland GmbH 1001198", "Media Markt Enschede B.V. Store N012 1001256", "Media-Saturn E-Business Concepts & Service GmbH 1001359", "UK Mobile Media 1001430", "Media Market S.p.A. Italien 1001800", "Media Markt Zell am See TV-Hifi-Elektro GmbH, Store A021 1008500", "Media Markt TV-HiFi-Elektro GmbH Oberwart, A024 1010100", and "Media-Saturn IT Service GmbH 1014500". The background shows the "Create New Email Ticket" form with a red banner at the top that says "SI Create new Agents and work with these accounts instead." and a search bar with "Admin" and a magnifying glass icon.

Logische Sicht OTRS



Alles ist ein Ticket/E-Mail

Ticket-Verursacher haben **E-Mail-Adresse**

E-Mail-Adressen gehören **Personen**

Personen arbeiten für **Firmen**

Nicht alle Personen einer Firma dürfen Ticket-Verursacher sein

Datenangebot kivi



Kundennummer

Anrede

Kundenname

Abteilung

Straße

PLZ/Stadt

Land

GLN

Kontakt

Telefon

Fax

E-Mail

Ansprechpersonen

Geschlecht

Titel

Abteilung

Funktion/Position

Vorname

Nachname

E-Mail

+ Darf Tickets melden!

db-view in kivi



```
CREATE VIEW otrs_customer_contacts AS
SELECT
    contacts.cp_title
  , contacts.cp_givenname
  , contacts.cp_name
  , contacts.cp_email
  , contacts.cp_phone1
  , contacts.cp_mobile1
  , 'dummydummy91823791827391723981543218123'::text AS cp_password
  , 1 AS cp_valid_id
  , customer.customernumber
  , customer.name
  , customer.street
  , customer.zipcode
  , customer.city
  , customer.country
  , customer.phone
  , customer.fax
  , customer.homepage
  , customer.notes
  , CASE WHEN customer.obsolete IS NULL OR NOT customer.obsolete THEN 1 ELSE 2
END AS customer_valid_id
  , customer.itime AS create_time
  , customer.mtime AS change_time
FROM contacts
LEFT JOIN customer ON (contacts.cp_cv_id = customer.id)
WHERE (customer.id IS NOT NULL)
      AND (cp_email IS NOT NULL)
      AND (cp_email <> '');
```

Tickets in kivi

Alle geschloßenen Tickets

- in einer Periode
- eines Kunden
- erfasste Zeit
- Nummer/Beschreibung

Prototyp get_closed_ticket kivitendo

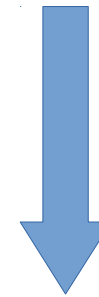
Alle erfolgreich geschloßenen Tickets

- des Vormonats
- eines Kunden
- erfasste Zeit
- Nummer/Beschreibung
- Eine Position pro Ticket
- Artikel in kivi vorbefüllt aus Auftrag (exp.)

Auftrag erfassen (exp.)



Preis	Rabatt		
		Artikel hinzufügen	Get closed tickets for previous month



D- 

Beschreibung: Systemtechnik Stundensatz

Nummer: 281653 Titel: Beamer Erfasste Zeit: 20 min

REST-Webservice OTRS



Yaml Definition in OTRS laden

```
---
Debugger:
  DebugThreshold: debug
  TestMode: '0'
Description: Ticket Connector REST Sample
FrameworkVersion: 3.4.x git
Provider:
  Operation:
    SessionCreate:|
      Description: Creates a Session
      MappingInbound: {}
      MappingOutbound: {}
      Type: Session::SessionCreate
    TicketCreate:
      Description: Creates a Ticket
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketCreate
    TicketGet:
      Description: Retrieves Ticket data
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketGet
    TicketSearch:
      Description: Search for Tickets
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketSearch
    TicketUpdate:
      Description: Updates a Ticket
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketUpdate
  Transport:
    Config:
      KeepAlive: ''
      MaxLength: '100000000'
      RouteOperationMapping:
        SessionCreate:
          RequestMethod:
            - POST
          Route: /Session
        TicketCreate:
          RequestMethod:
            - POST
          Route: /Ticket
```


Aufwand 20 h



OTRS

- verstehen
- Mapping yaml Definition und Client-Calls
- Uses CamelCaseForReadingSource
- Objekt-Modell OTRS (extendend)
- 12 Zustände eines Tickets

kivi

- otrs Zugangs-Daten konfigurierbar
- REST Calls modularisieren
- Ticket-Zustände konfigurierbar
- Ticket-Periode konfigurierbar
- optional wiederkehrende Rechnungen