



CustomerMultitenancy

Release rel-11.0

Rother OSS GmbH

Dec 14, 2024

Contents

1	Description	3
2	System requirements	5
2.1	Framework	5
2.2	Packages	5
2.3	Third-party software	5
3	Setup	7
3.1	Backends	7
3.2	Frontend	8
4	Configuration Reference	9
4.1	Core::Permission	9
5	About	11
5.1	Contact	11
5.2	Version	11

(T U B)

Description

This package allows group based restrictions on customer and customer user information for agents. Per default all customers are visible for agents. After installation of this package customers can be assigned to an agent group. Those customers are then visible only for agents with access rights to the specific group. Also complete customer sources can be restricted.

Access rights on customer users will be inherited by the rights set on the customers, but can also individually be set.

This functionality works for the customer (user) tables of OTOBO, and read only for remote databases as well as LDAP.

System requirements

2.1 Framework

OTOBO 11.0.x

2.2 Packages

-

2.3 Third-party software

-

In the Config.pm a set of adjustments is mandatory. For both DB and LDAP it is necessary to change the CustomerCompany and/or CustomerUser blocks. (If you do not have them present, please copy them for Kernel/Config/Defaults.pm to Kernel/Config.pm and adjust them in the latter!) To enable setting of the agent group in AdminCustomer and AdminCustomerUser the variables have to be added to the maps.

3.1 Backends

3.1.1 Database

For the DB backend in both the CustomerCompany and CustomerUser definitions this can be added as follows.

```
$Self->{CustomerUser} = {  
  # ...  
  Map => [  
    # ...  
    [ 'UserGroupID',      Translatable('Only visible to agents in group'),  
      'group_id',        0, 0, 'var', '', 0, undef, undef ],  
    # ...  
  ],  
};
```

3.1.2 LDAP

For the LDAP backend in both the CustomerCompany and CustomerUser definitions this is similar.

```
$Self->{CustomerUser} = {  
  # ...
```

```
Map => [
  # ...
  [ 'UserGroupID',      Translatable('Only visible to agents in group'),
    'otoboGroup',      0, 0, 'var', '', 0, undef, undef ],
  # ...
],
};
```

'otoboGroup' here is just exemplary and has to be replaced with the attribute of the directory. Additionally in most cases a mapping between AD and OTOBO groups will be necessary. This is configured by the following.

```
$Self->{CustomerUser} = {
  # ...

  # Additional settings for the UserGroupID mapping.
  UserGroupIDSync => {
    # Match group names instead of group IDs.
    UseGroupNames => 1,
    # Remap group names or IDs.
    # The key is the group name/ID on the remote system (DB or LDAP-Attribute)
    # and the value is the group name/ID on the local system.
    RemoteGroupToLocalGroup => {
      'ad-admin-group' => 'admin',
      'ad-user-group'  => 'users',
    },
  },
};
```

3.1.3 Global

Independently of the backend, it can also be restricted completely.

```
$Self->{CustomerUser} = {
  # ...

  # Only allow access to the customer user source to members of this group:
  CustomerUserGroup => 'users',

  # ...
};
```

3.2 Frontend

In the customer and customer user administrations the groups can now be configured or seen for read only backends.

4.1 Core::Permission

4.1.1 Multitenancy

Enables the multitenancy function for customer and customer user.

4.1.2 Multitenancy::PermissionGroup

Specifies the group that can set multitenancy for customer. Multitenancy does not apply to members of this group.

5.1 Contact

Rother OSS GmbH
Email: hello@otobo.de
Web: <https://otobo.de>

5.2 Version

Author: Rother OSS GmbH / Version: rel-11.0 / Date of release: 2024-12-14