
groundwork-users Documentation

Release 0.1.2

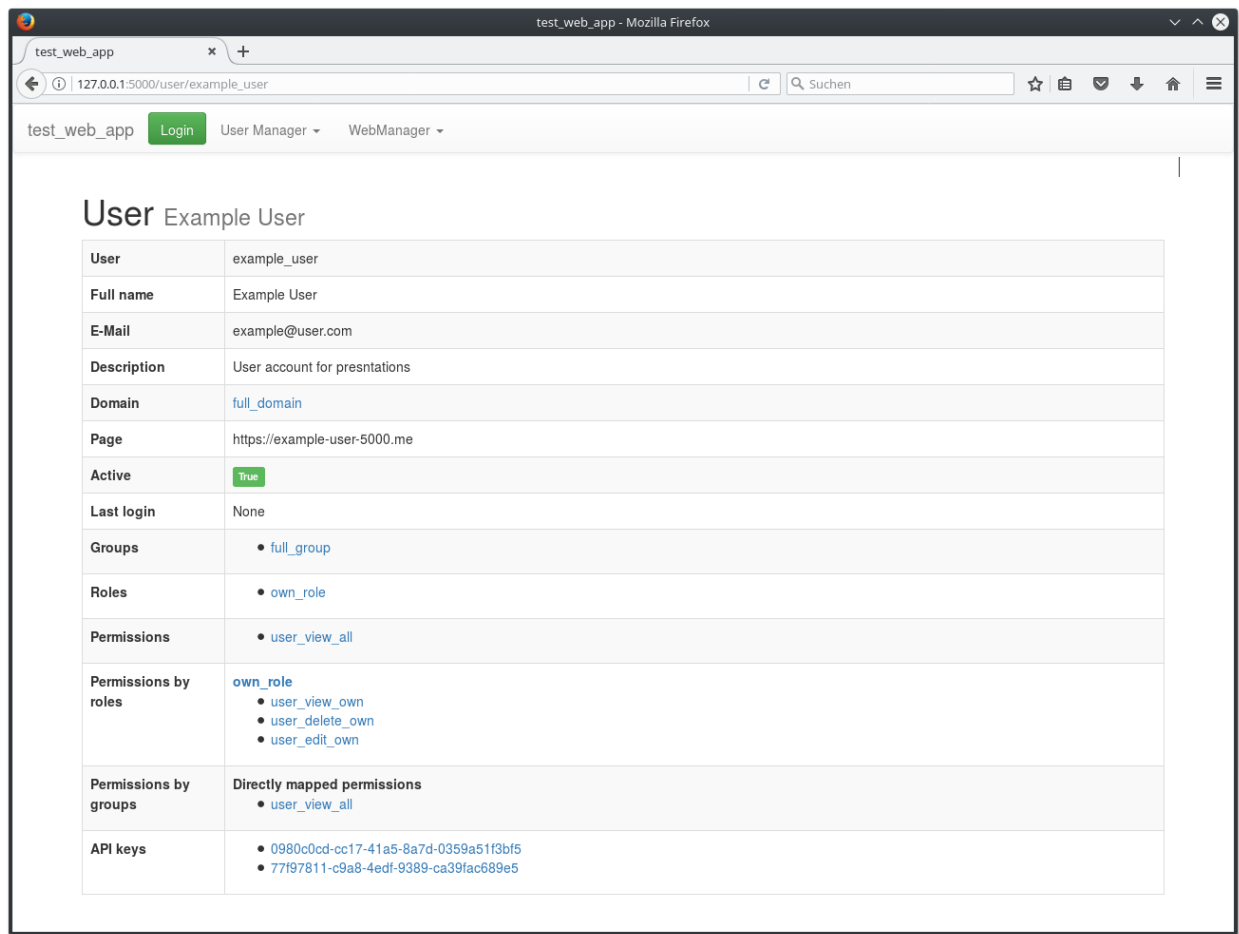
team useblocks

Jun 08, 2017

Contents

1	Functions	3
2	Package content	5
2.1	Screenshots	5
2.2	JINJA Templates	21
2.3	Configuration	22
2.4	API	22
	Python Module Index	23

This is a [groundwork](#) extension package for managing users and related functions.



More images are shown on the [Screenshots](#) page.

groundwork framework

[groundwork](#) is a plugin based Python application framework, which can be used to create various types of applications: console scripts, desktop apps, dynamic websites and more.

Visit groundwork.useblocks.com or read the [technical documentation](#) for more information.

CHAPTER 1

Functions

All of the following functions are available as API via *GwUsersPattern* or as ready-to-use web views via the plugin *GwUsersWebManager*.

Users and Groups

- Create, edit and delete users and their data
- Create, edit and delete groups
- Bundle users inside groups
- Activate/Deactivate user accounts

Permissions and Roles

- Create, edit and delete permissions
- Create, edit and delete roles
- Bundle permissions inside roles
- Assign roles to users/groups
- Assign single permission to users/groups
- Secure functions/views with needed permissions

Authentication

- Authenticate users by basicAuth, token, session or api_key
- Secure functions with specific auth methods like “api_key only for API calls”.

API keys

- Create, edit and delete API keys
- Assign multiple API keys to users
- Activate/Deactivate api keys

Domains

- Create, edit and delete domains
- Bundle users to domains for multi-client-support ([Multitenancy](#))

CHAPTER 2

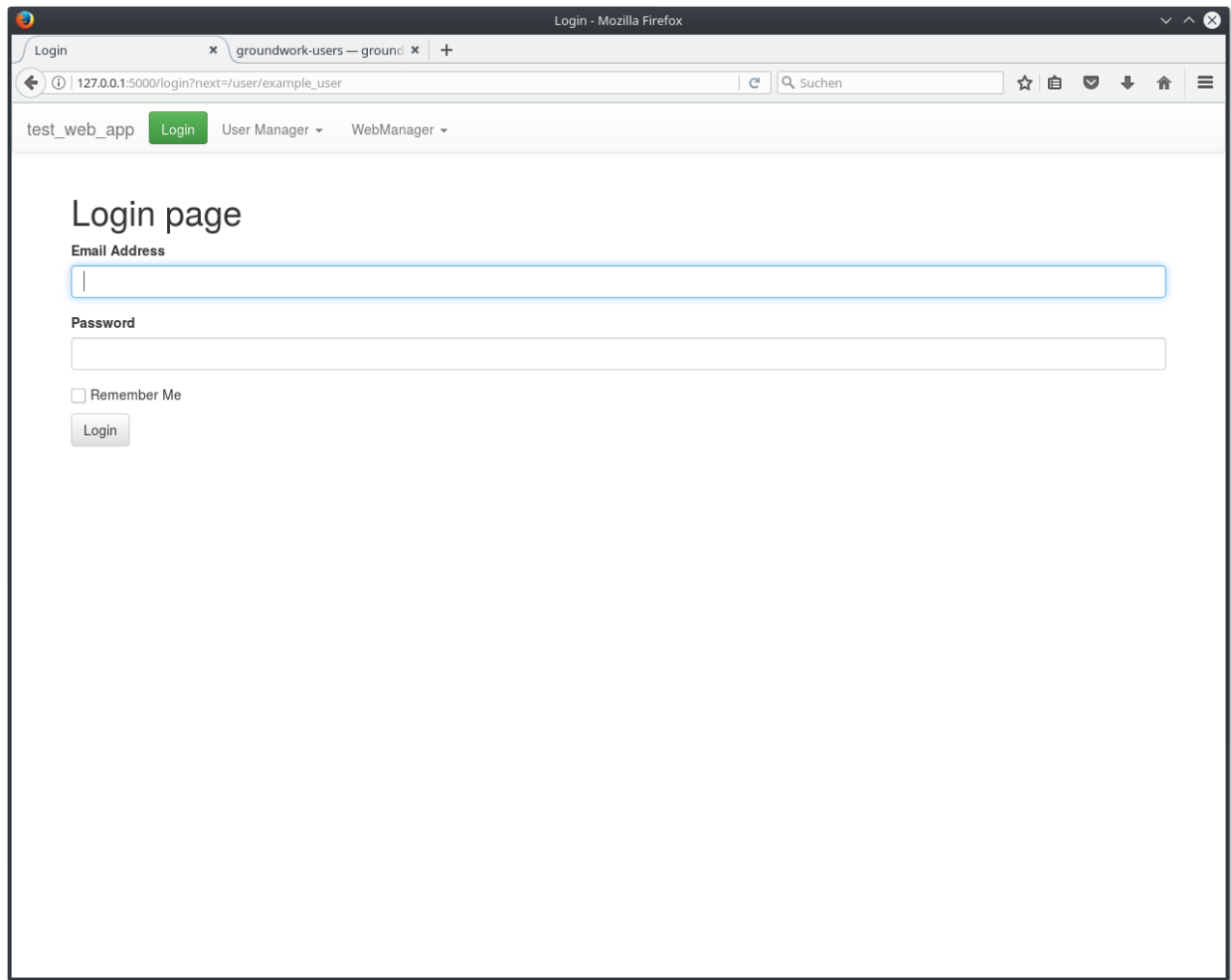
Package content

- Plugins
- GwUsersCliManager
- GwUsersWebManager
- Patterns
- GwUsersPattern

Screenshots

The following screenshots are all taken from the plugin *GwUsersWebManager*.

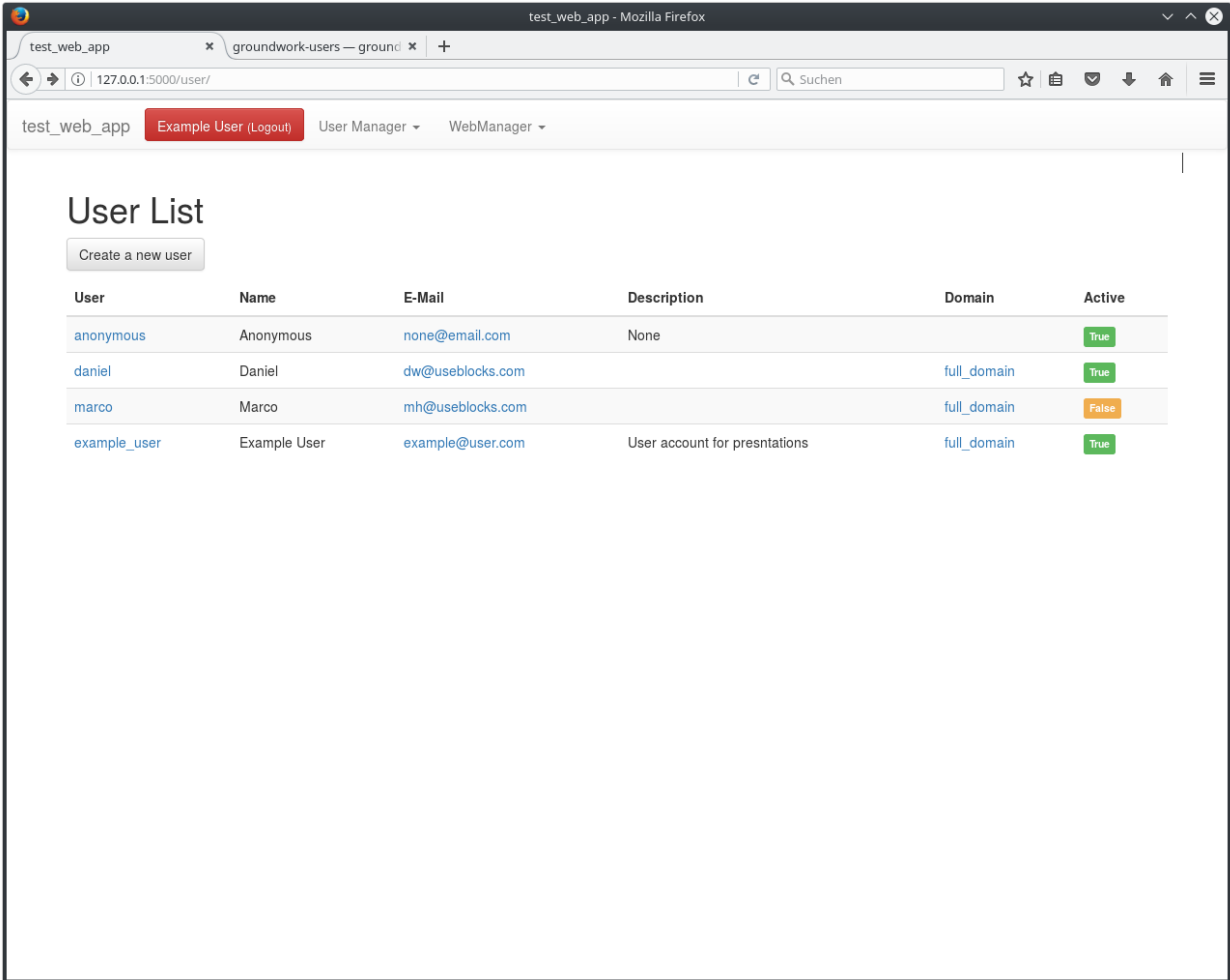
Login



The screenshot shows a web browser window titled "Login - Mozilla Firefox". The address bar displays the URL "127.0.0.1:5000/login?next=/user/example_user". The browser's navigation bar includes a search bar with the text "Suchen" and several icons (star, document, checkmark, download, home, and menu). The page header shows "test_web_app" followed by a green "Login" button and two dropdown menus labeled "User Manager" and "WebManager". The main content area is titled "Login page" and contains the following elements:

- Email Address**: A text input field with a blue border.
- Password**: A text input field with a grey border.
- ☐ Remember Me: A checkbox with the label "Remember Me".
- Login: A grey button labeled "Login".

Users

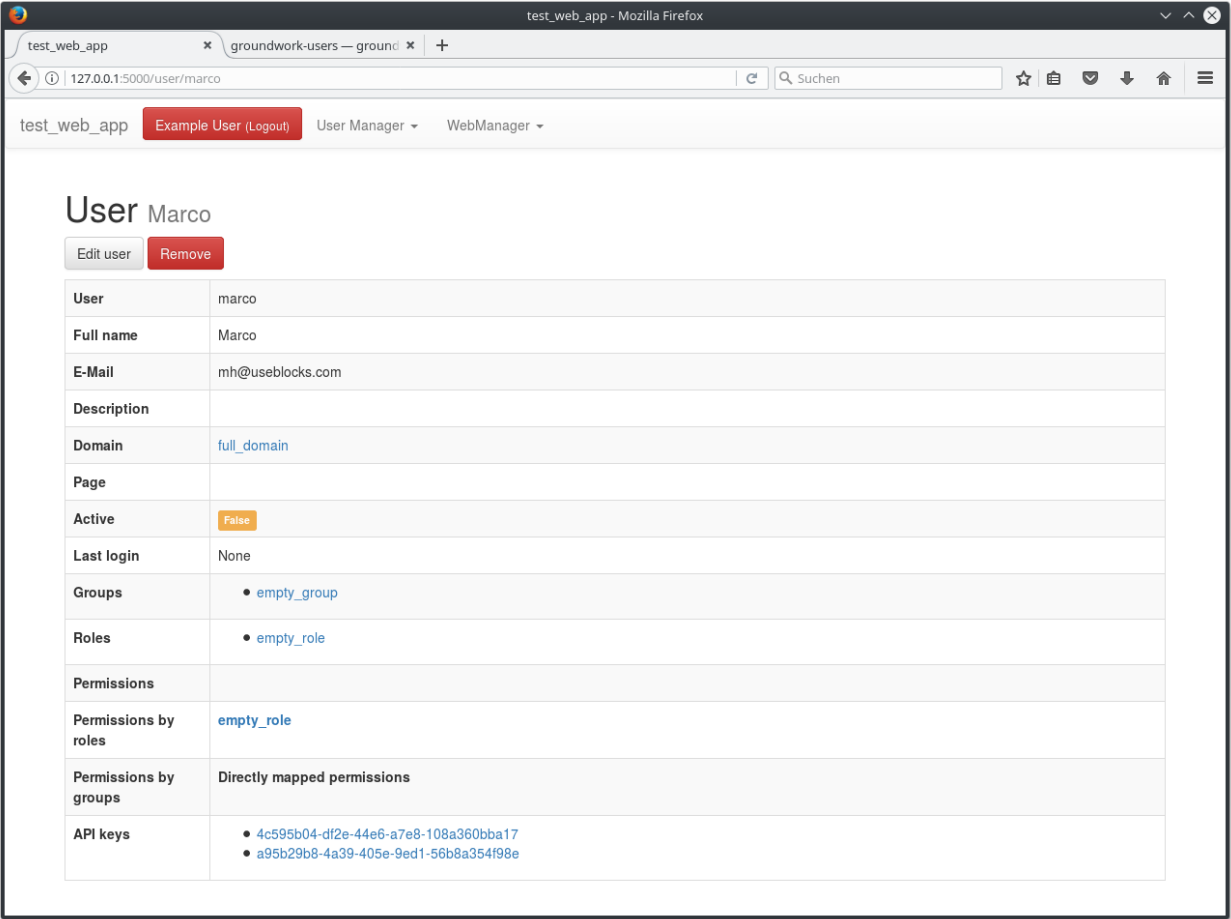


The screenshot shows a web browser window titled "test_web_app - Mozilla Firefox" with the address bar displaying "127.0.0.1:5000/user/". The page header includes the text "test_web_app" and a red "Example User (Logout)" button, followed by "User Manager" and "WebManager" dropdown menus. The main content area is titled "User List" and features a "Create a new user" button. Below this is a table with the following columns: User, Name, E-Mail, Description, Domain, and Active. The table contains four rows of user data.

User	Name	E-Mail	Description	Domain	Active
anonymous	Anonymous	none@email.com	None		True
daniel	Daniel	dw@useblocks.com		full_domain	True
marco	Marco	mh@useblocks.com		full_domain	False
example_user	Example User	example@user.com	User account for presentations	full_domain	True

The screenshot shows a web browser window with the URL `127.0.0.1:5000/user/add`. The page title is "Usermanager new". The form contains the following fields and sections:

- Email Address**: A text input field.
- User name**: A text input field.
- Full name**: A text input field.
- Domain**: A dropdown menu showing "empty domain". Below it is the text "Select domain".
- Active**: A checkbox. Below it is the text "Is user active and can be used for logins?".
- Select groups**: A list box showing "full_group" and "empty_group". Below it is the text "Select groups".
- Select roles**: A list box showing "Anonymous", "empty_role", "admin_role", and "own_role". Below it is the text "Select permission roles".
- Select permissions**: A list box showing "user_view_all" and "user_edit_all".



test_web_app | Example User (Logout) | User Manager | WebManager

Usermanager

Email Address
mh@useblocks.com

User name
marco

Full name
Marco

Domain
full domain
Select domain

Active
☐
Is user active and can be used for logins?

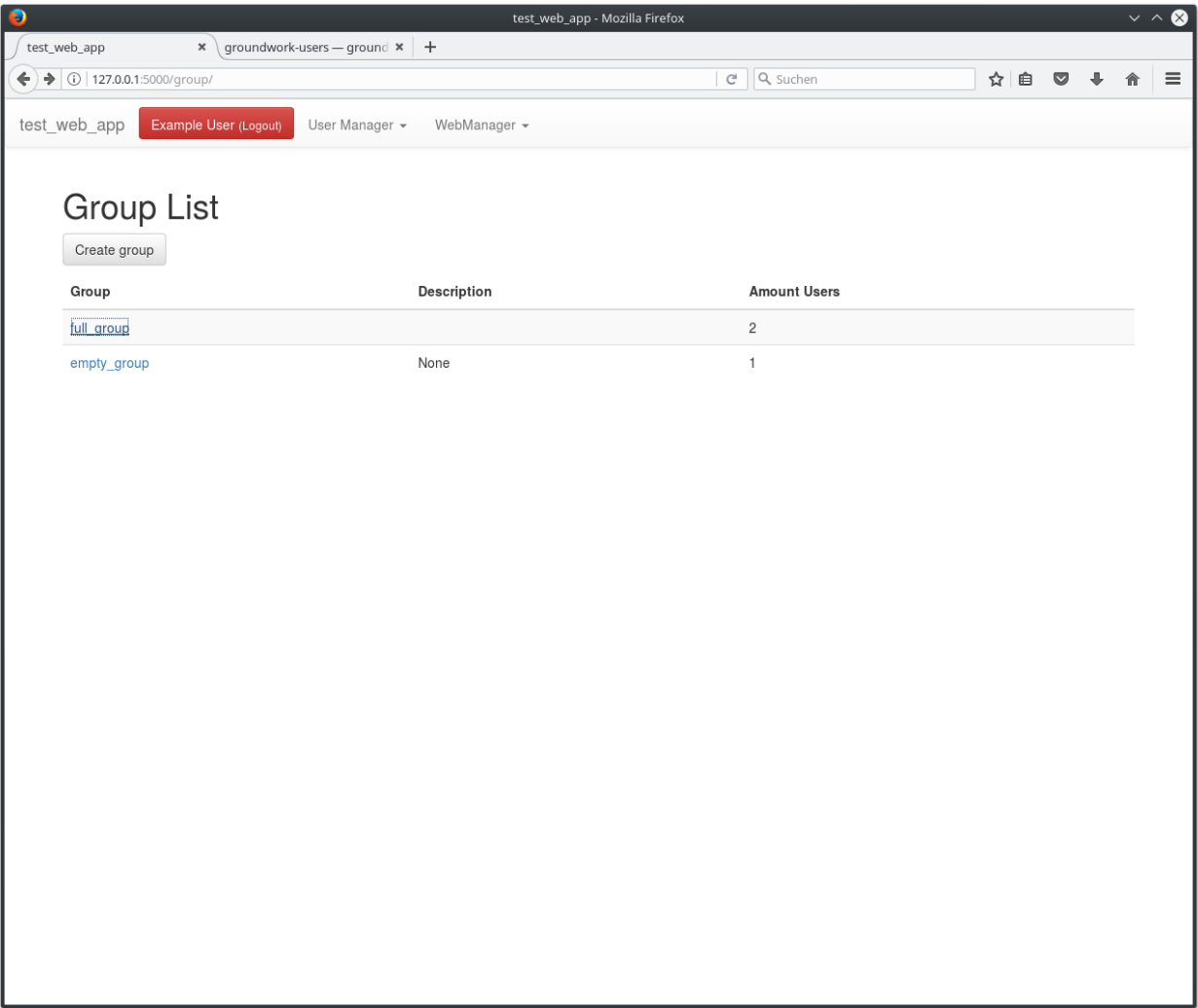
Select groups
full_group
empty_group
Select groups

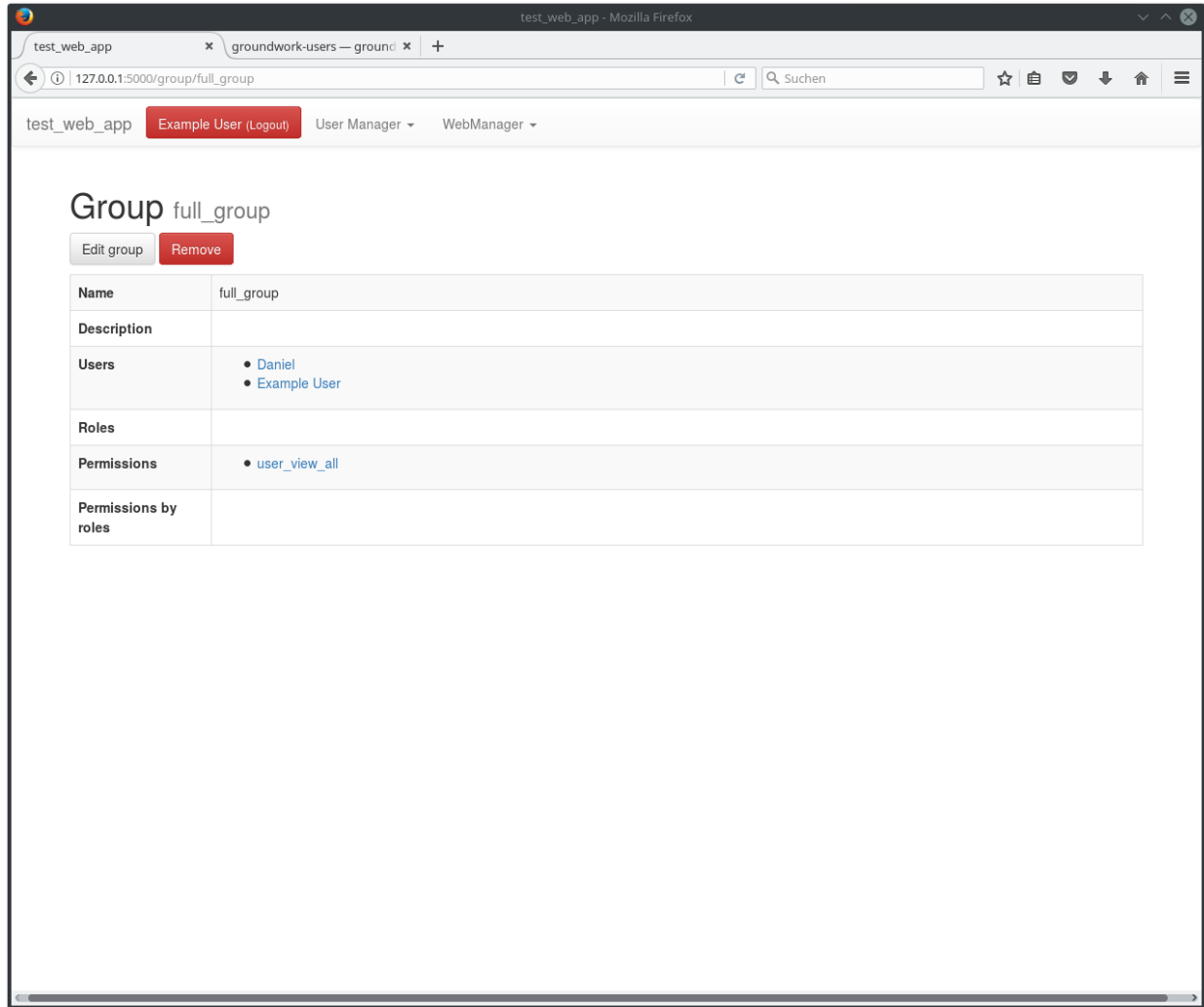
Select roles
Anonymous
empty_role
admin_role
own_role
Select permission roles

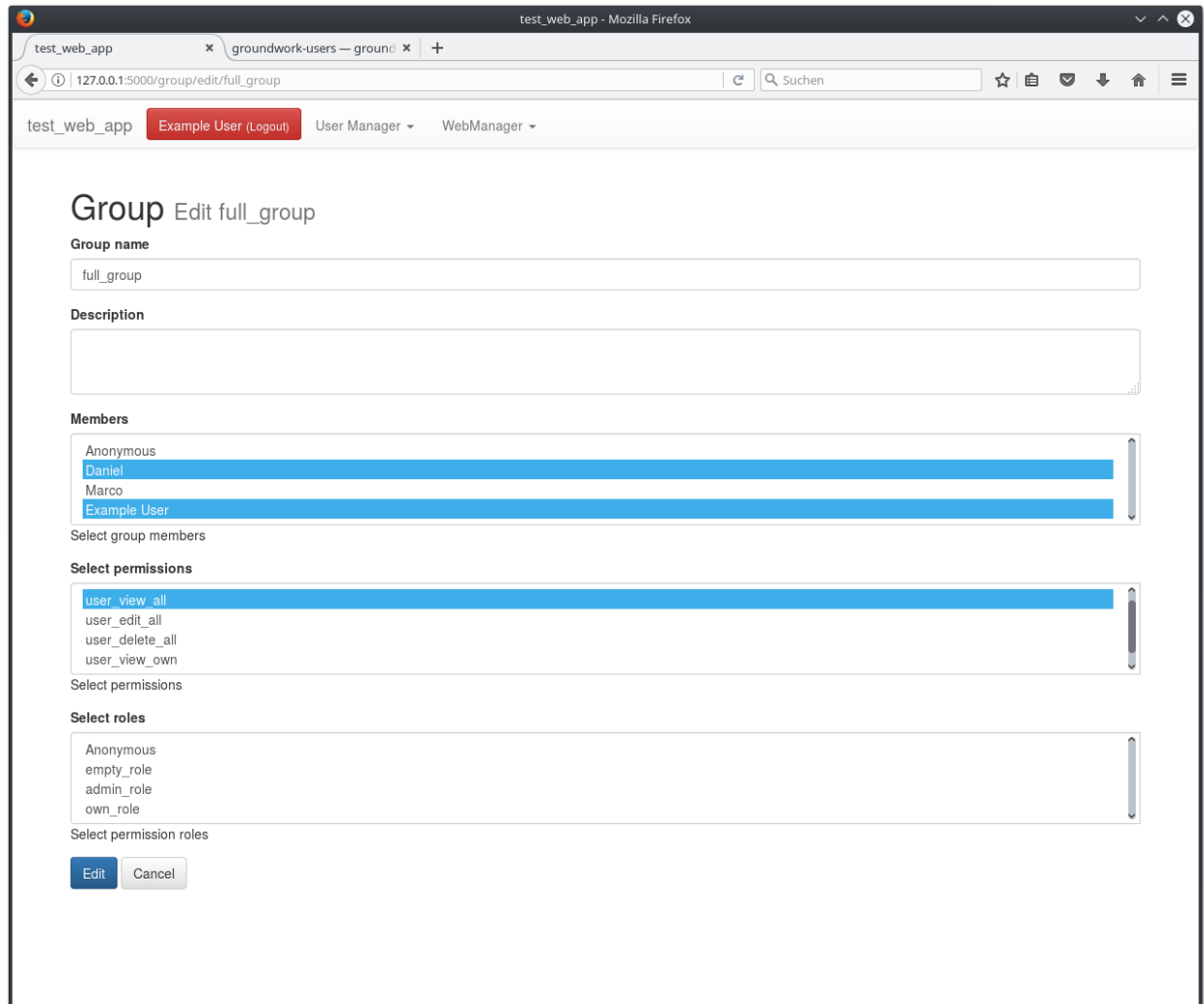
Select permissions
user_view_all
user_edit_all
user_delete_all
user_view_own
Select permissions

Description

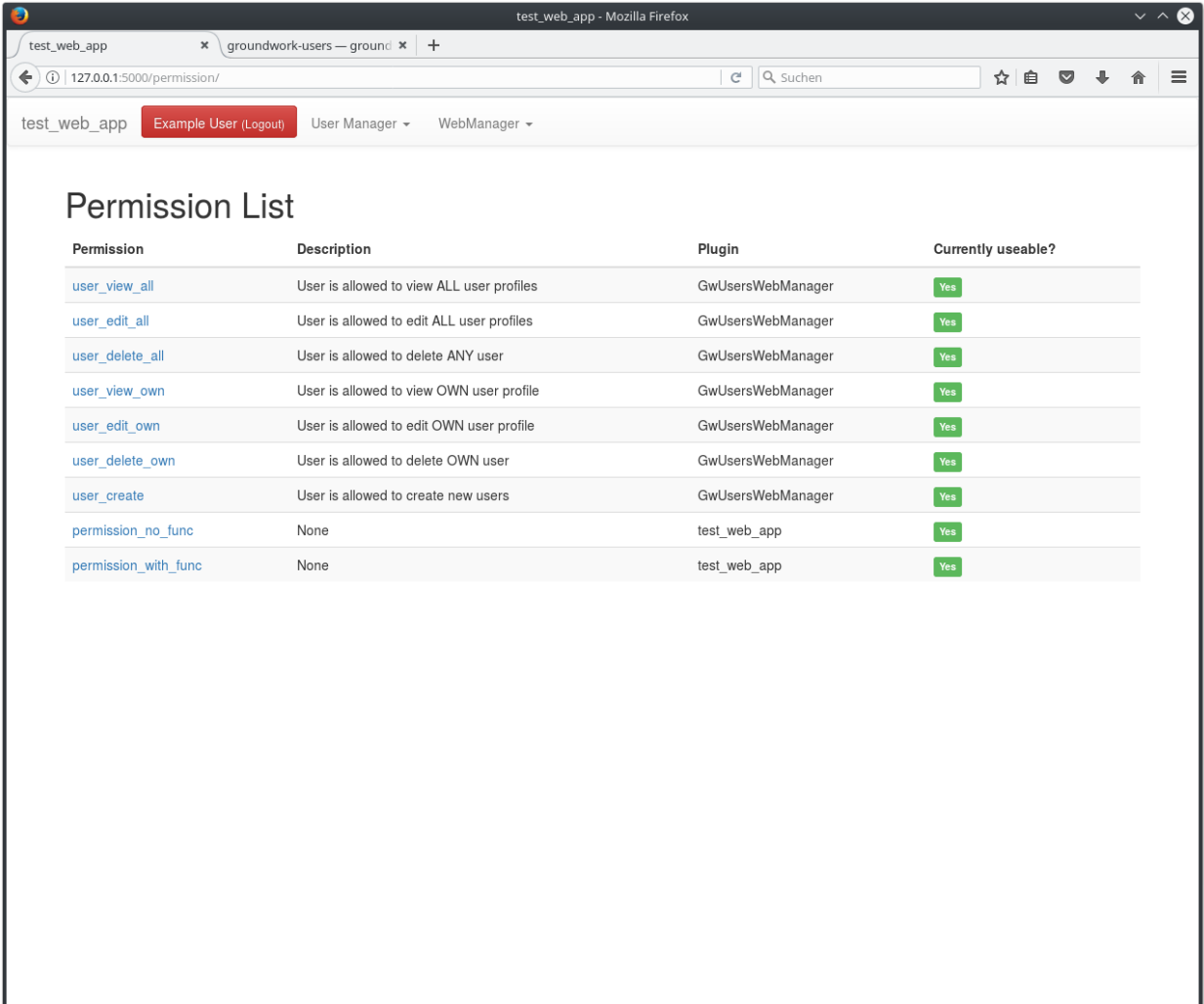
Groups





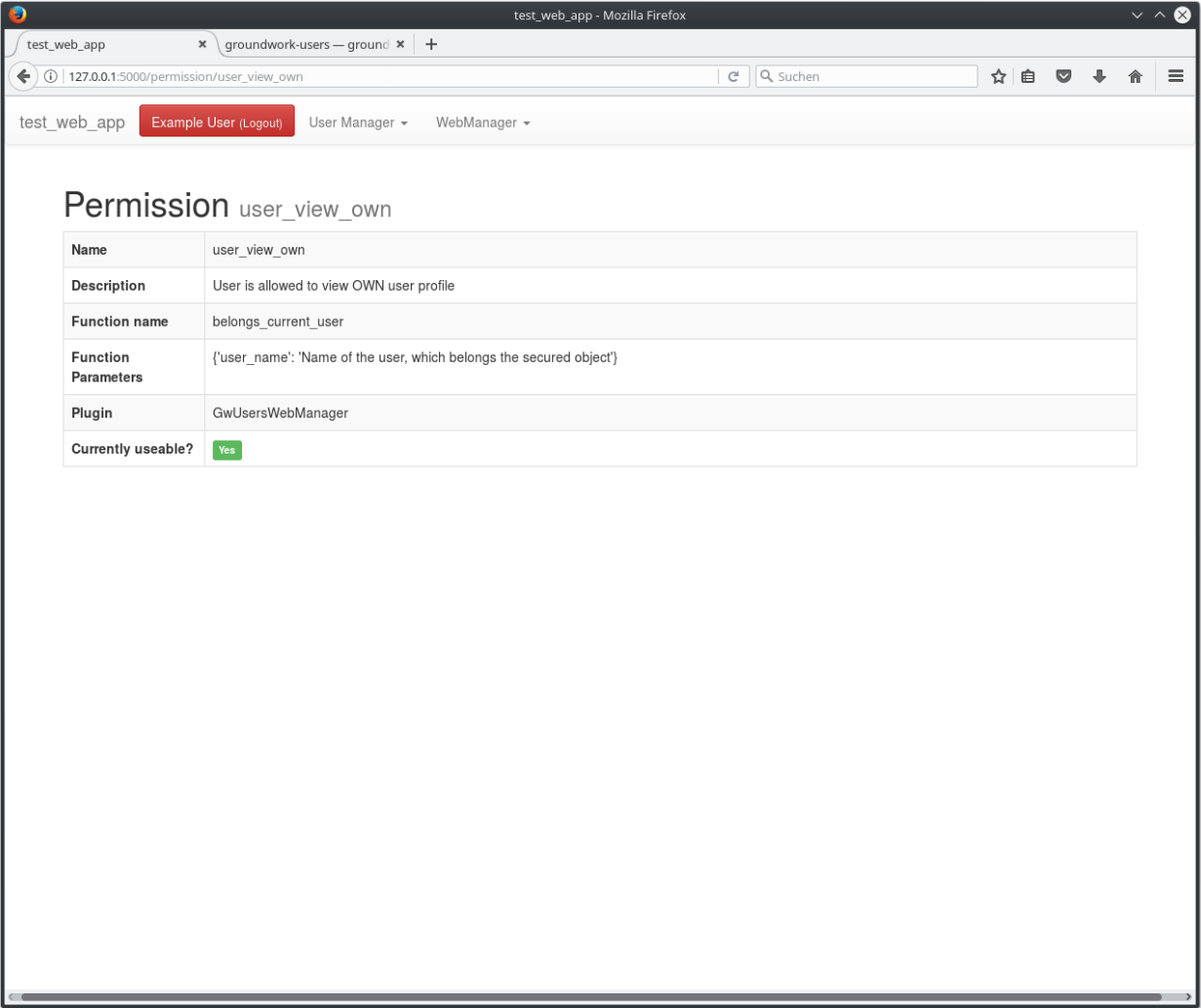


Permissions

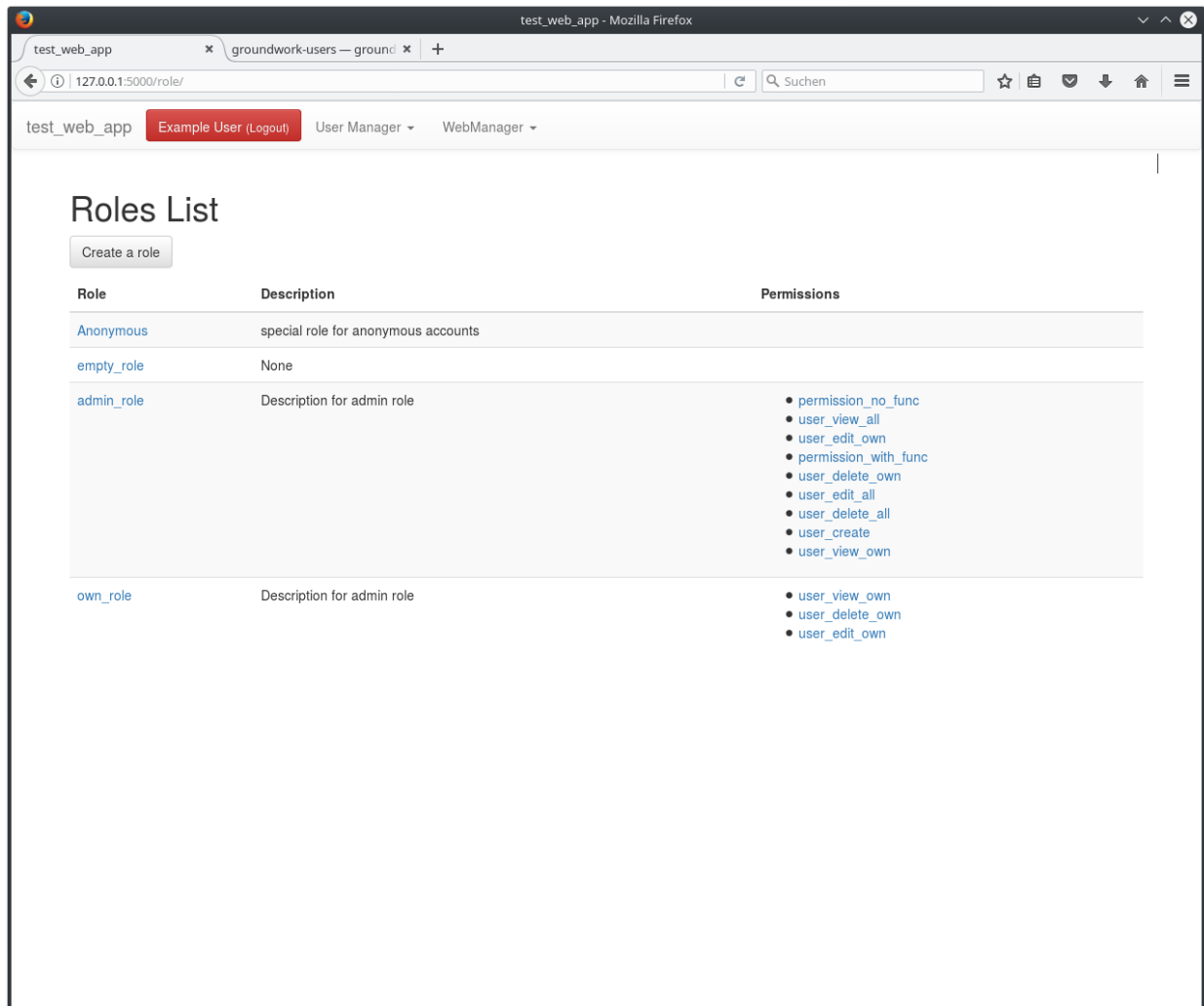


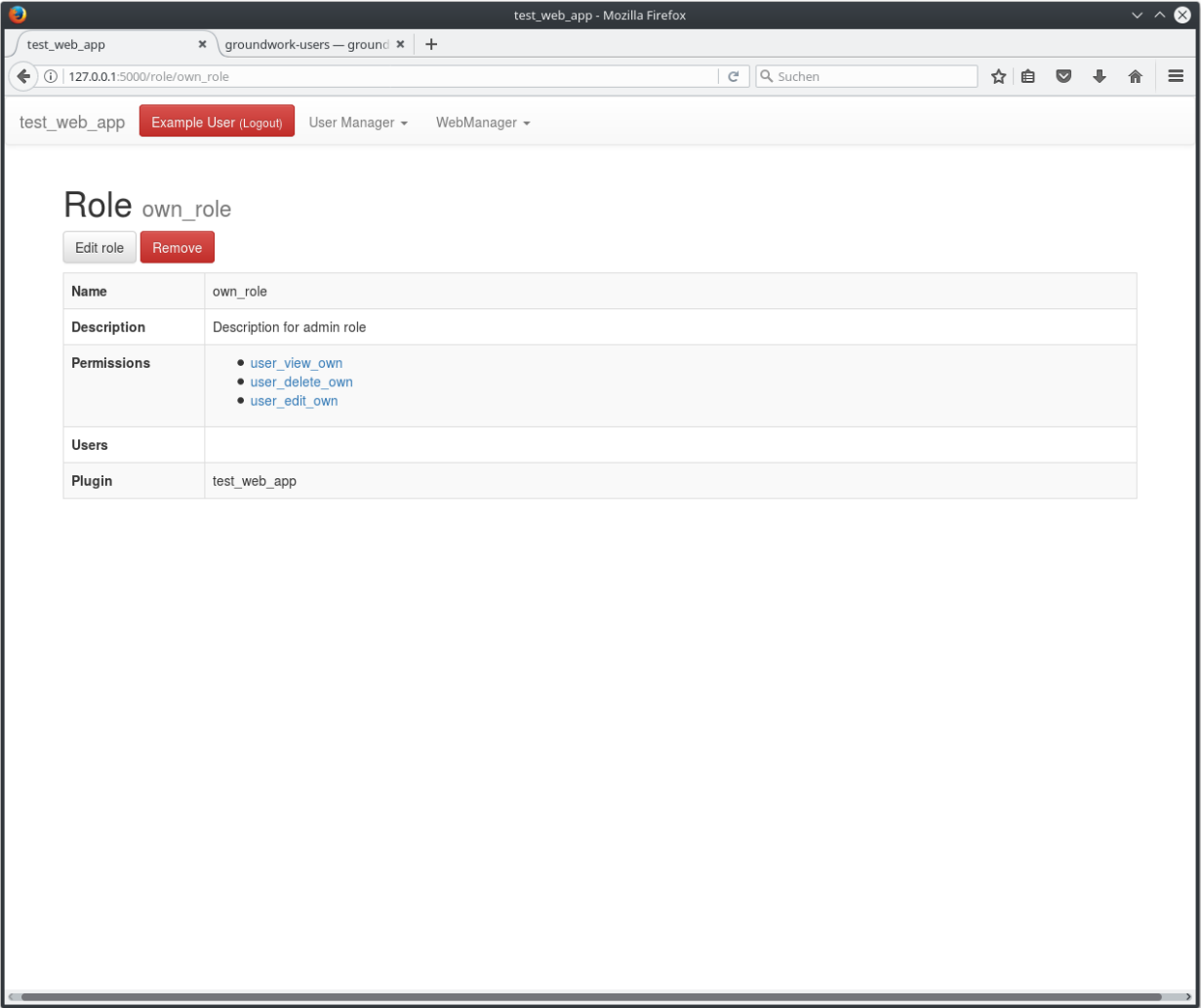
The screenshot shows a web browser window titled "test_web_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/permission/". The page header includes "test_web_app", a red "Example User (Logout)" button, and dropdown menus for "User Manager" and "WebManager". The main content area is titled "Permission List" and contains a table with the following data:

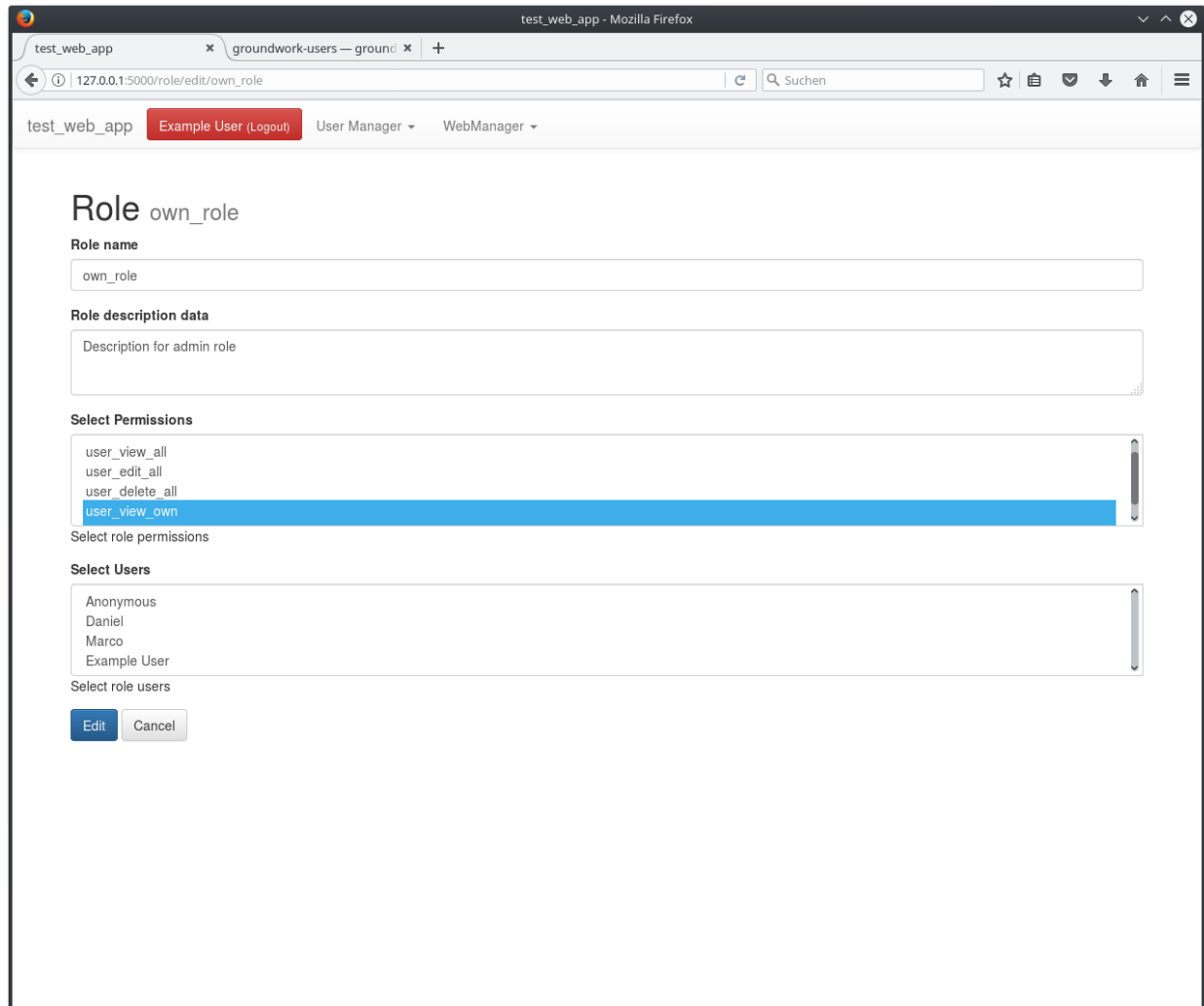
Permission	Description	Plugin	Currently useable?
user_view_all	User is allowed to view ALL user profiles	GwUsersWebManager	Yes
user_edit_all	User is allowed to edit ALL user profiles	GwUsersWebManager	Yes
user_delete_all	User is allowed to delete ANY user	GwUsersWebManager	Yes
user_view_own	User is allowed to view OWN user profile	GwUsersWebManager	Yes
user_edit_own	User is allowed to edit OWN user profile	GwUsersWebManager	Yes
user_delete_own	User is allowed to delete OWN user	GwUsersWebManager	Yes
user_create	User is allowed to create new users	GwUsersWebManager	Yes
permission_no_func	None	test_web_app	Yes
permission_with_func	None	test_web_app	Yes



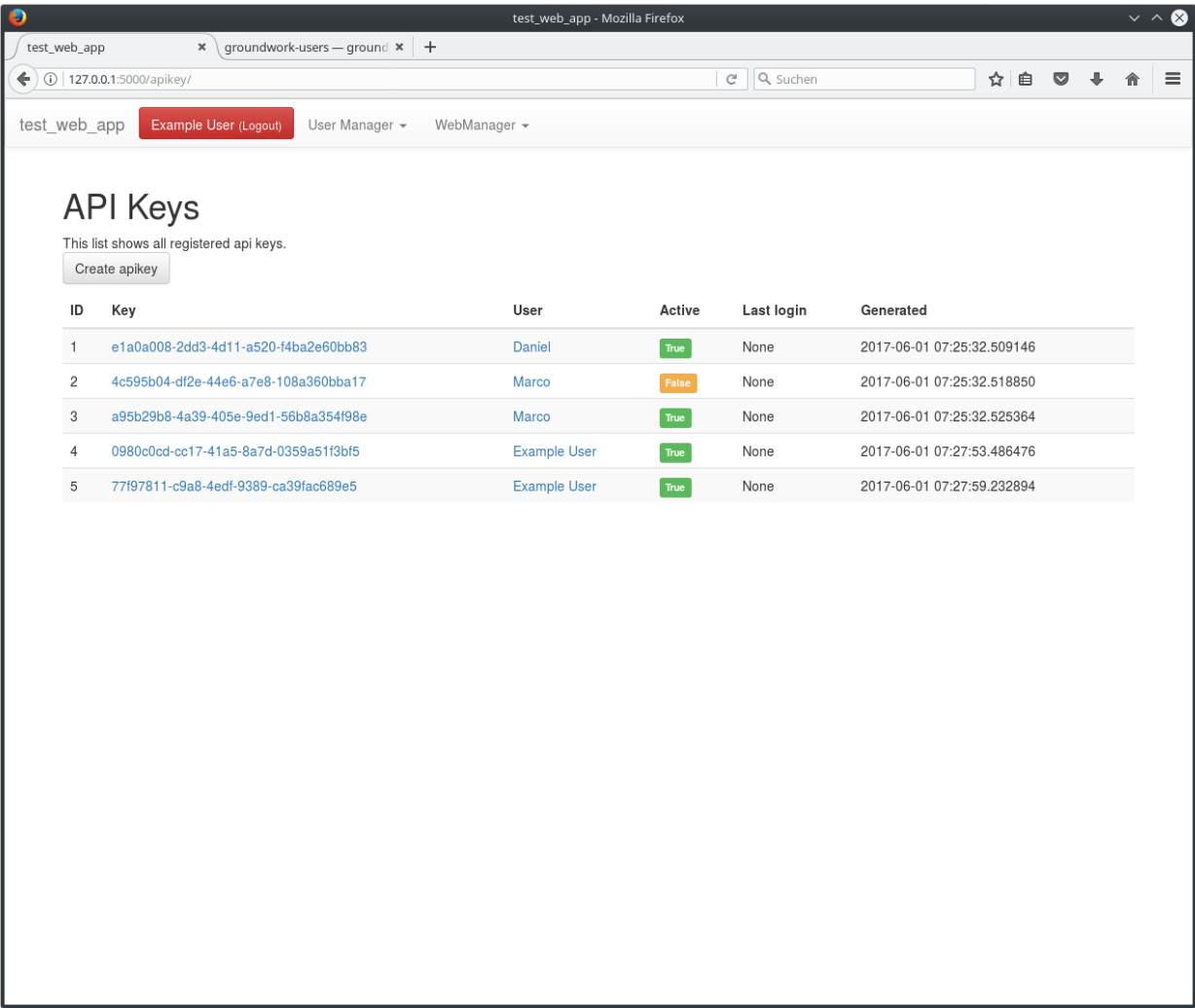
Roles

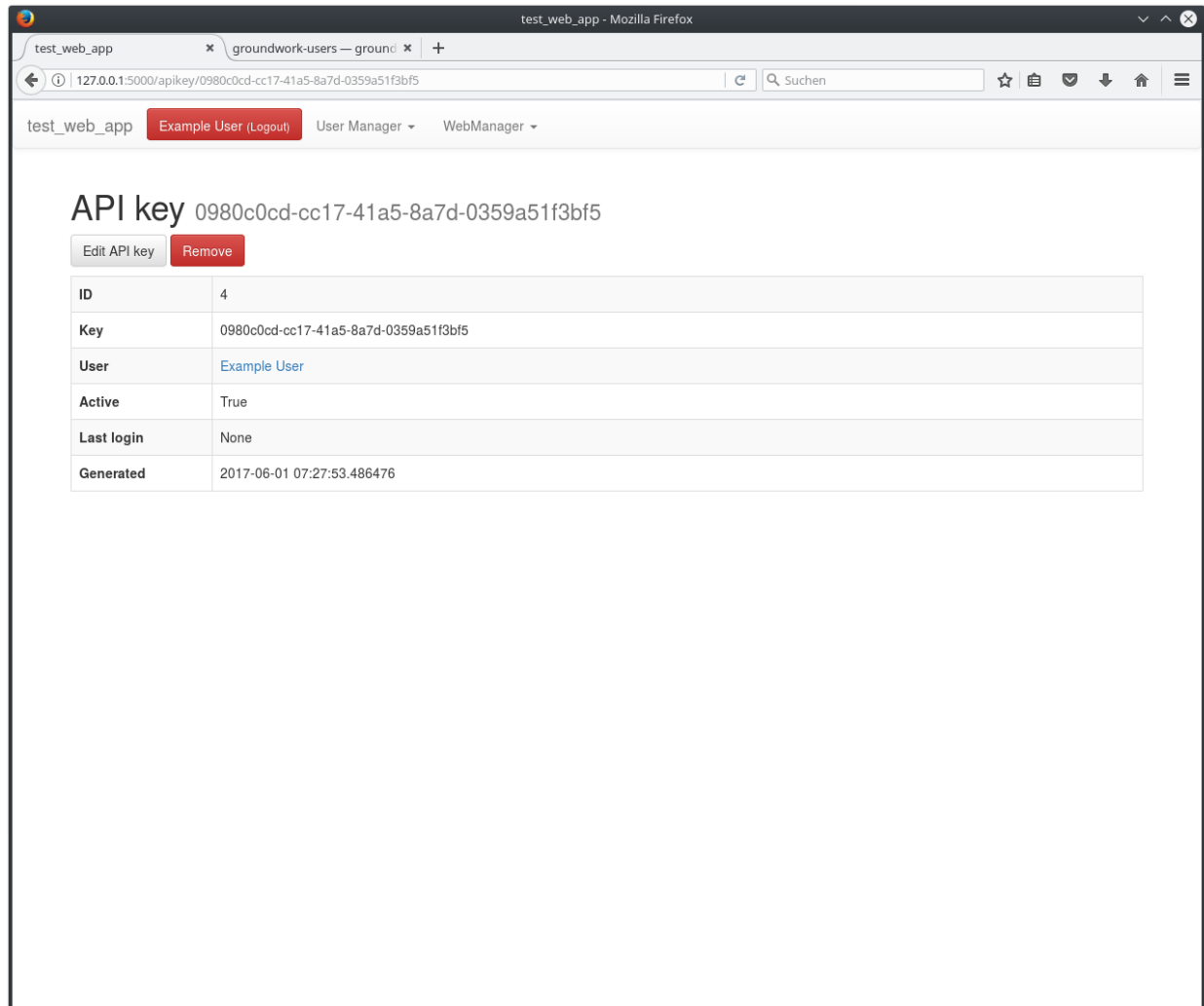


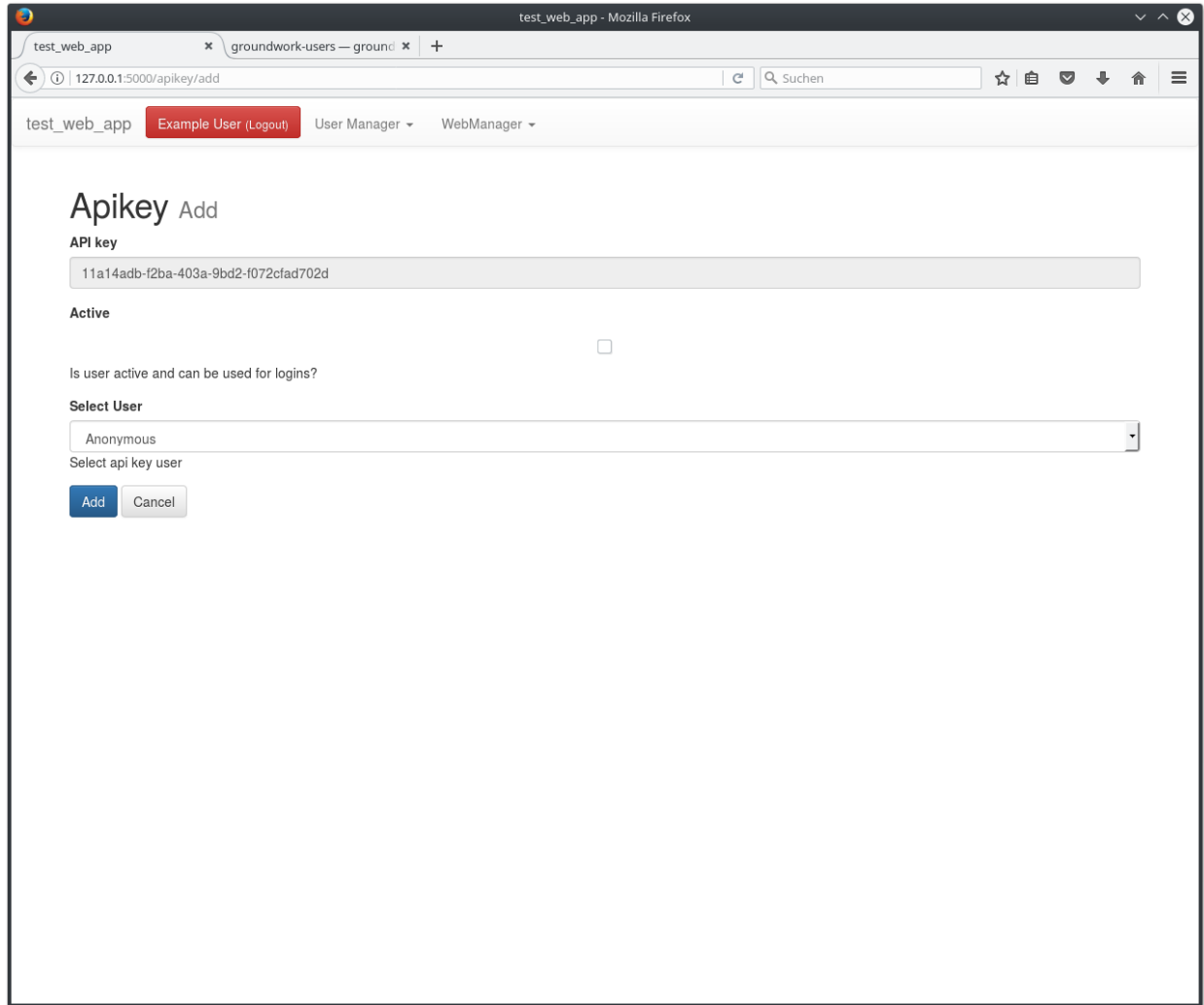




API keys







JINJA Templates

Check permission

You can easily check the permission status of the current user:

```
{% if current_user.has_permission("user_edit_all") %}
<a class="btn btn-default" href="{{url_for('.add')}}">
  {{_("Create a new user")}}</a>
{% endif %}
```

The above code will show a button “Create a new user” only, if the current logged in user has the permission “user_create”.

Configuration

Application configuration parameters

USERS_DB_URL

Database connection string / url, which defines the database tu use for storing user related data.:

```
USERS_DB_URL = "sqlite://{0}/my_users.db".format(APP_PATH)
```

API

Plugins

GwUsersWebManager

```
class groundwork_users.plugins.GwUsersWebManager (app, *args, **kwargs)
```

Patterns

GwUsersPattern

```
class groundwork_users.patterns.GwUsersPattern (app, *args, **kwargs)
```

users = None

Instance of UsersPlugin. Provides functions to register and manage users

g

groundwork_users, [22](#)

G

groundwork_users (module), [22](#)

GwUsersPattern (class in groundwork_users.patterns), [22](#)

GwUsersWebManager (class in groundwork_users.plugins), [22](#)

U

users (groundwork_users.patterns.GwUsersPattern attribute), [22](#)