

---

# **groundwork-users Documentation**

***Release 0.1.2***

**team useblocks**

**Jun 08, 2017**



---

## Contents

---

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



This is a `groundwork` extension package for managing users and related functions.

User	example_user
<b>Full name</b>	Example User
<b>E-Mail</b>	example@user.com
<b>Description</b>	User account for presentations
<b>Domain</b>	full_domain
<b>Page</b>	<a href="https://example-user-5000.me">https://example-user-5000.me</a>
<b>Active</b>	True
<b>Last login</b>	None
<b>Groups</b>	• full_group
<b>Roles</b>	• own_role
<b>Permissions</b>	• user_view_all
<b>Permissions by roles</b>	own_role • user_view_own • user_delete_own • user_edit_own
<b>Permissions by groups</b>	Directly mapped permissions • user_view_all
<b>API keys</b>	• 0980c0cd-cc17-41a5-8a7d-0359a51f3bf5 • 77f97811-c9a8-4edf-9389-ca39fac689e5

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](http://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

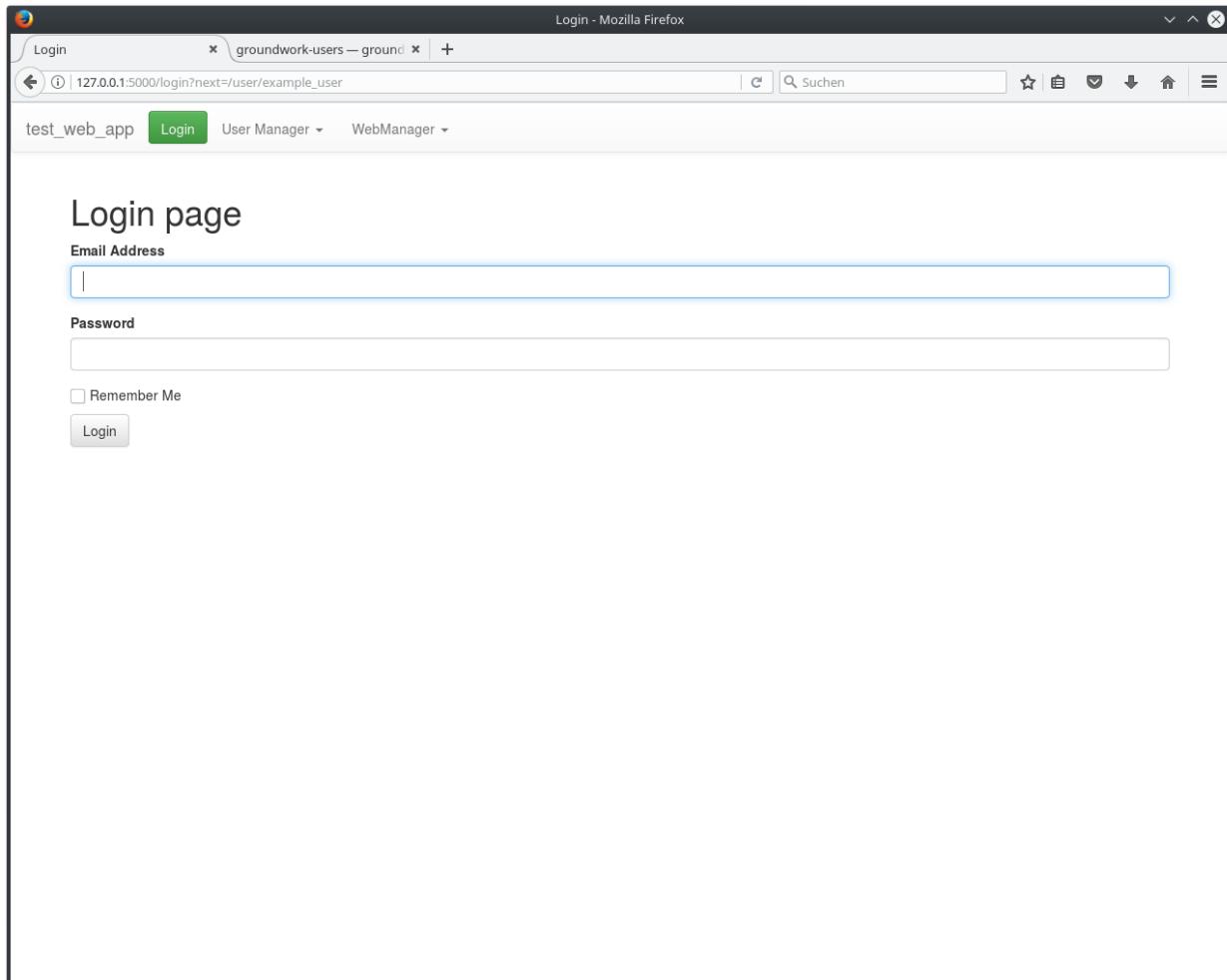
---

- Plugins
- GwUsersCliManager
- GwUsersWebManager
- Patterns
- GwUsersPattern

## Screenshots

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

## Login



## Users

The screenshot shows a Mozilla Firefox browser window titled "test\_web\_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/user/". The page content is titled "User List" and contains a table of users:

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 Mozilla Firefox browser window with the title "test\_web\_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/user/add". The page content is titled "Usermanager new". The form fields are as follows:

- Email Address: A text input field.
- User name: A text input field.
- Full name: A text input field.
- Domain: A dropdown menu with options "empty\_domain" and "Select domain".
- Active: A checkbox labeled "Is user active and can be used for logins?".
- Select groups: A dropdown menu with options "full\_group" and "empty\_group".
- Select roles: A dropdown menu with options "Anonymous", "empty\_role", "admin\_role", and "own\_role".
- Select permission roles: A dropdown menu with options "user\_view\_all" and "user\_edit\_all".

The screenshot shows a Mozilla Firefox browser window with the title "test\_web\_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/user/marco". The page content is titled "User Marco" and contains a table with user information. The table has two columns: "User" and "Value". The rows are:

User	marco
Full name	Marco
E-Mail	mh@useblocks.com
Description	
Domain	full_domain
Page	
Active	False
Last login	None
Groups	• empty_group
Roles	• empty_role
Permissions	
Permissions by roles	empty_role
Permissions by groups	Directly mapped permissions
API keys	• 4c595b04-df2e-44e6-a7e8-108a360bba17 • a95b29b8-4a39-405e-9ed1-56b8a354f98e

The screenshot shows a Mozilla Firefox browser window titled "test\_web\_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/user/edit/marco". The page content is titled "Usermanager". It contains the following fields:

- Email Address: mh@useblocks.com
- User name: marco
- Full name: Marco
- Domain: full domain (selected)
- Active: A checkbox is unchecked.
- Select groups: A dropdown menu lists "full\_group" and "empty\_group", with "empty\_group" currently selected.
- Select roles: A dropdown menu lists "Anonymous", "empty\_role", "admin\_role", and "own\_role", with "empty\_role" currently selected.
- Select permission roles: A dropdown menu lists "user\_view\_all", "user\_edit\_all", "user\_delete\_all", and "user\_view\_own".
- Description: An empty text input field.

## Groups

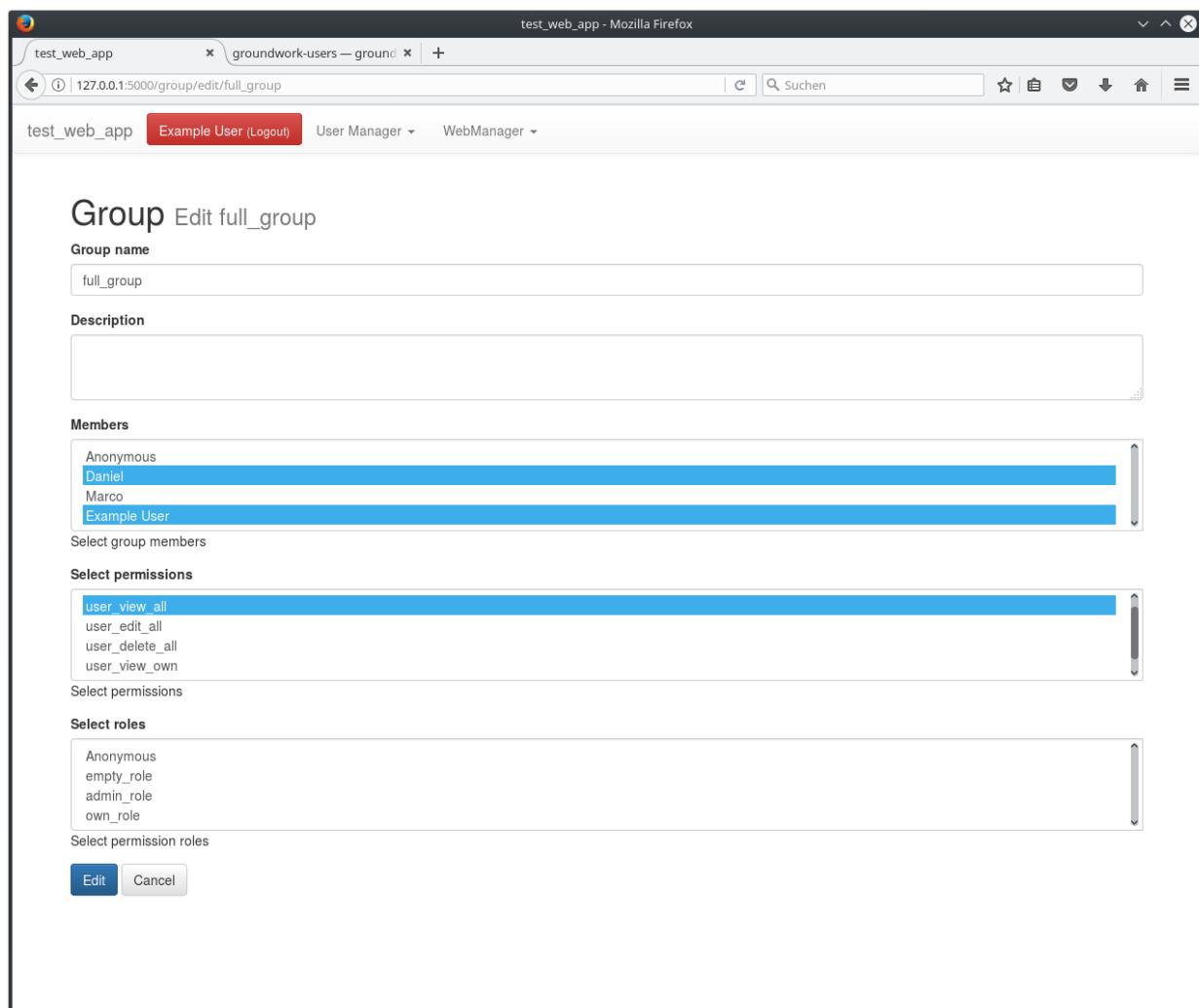
The screenshot shows a Mozilla Firefox browser window with the title bar "test\_web\_app - Mozilla Firefox". The address bar displays "test\_web\_app" and "groundwork-users — ground" with the URL "127.0.0.1:5000/group/". The main content area is titled "Group List" and contains a table with two rows. The table has columns: "Group", "Description", and "Amount Users". The first row has a "Group" value of "full\_group", a "Description" value of "None", and an "Amount Users" value of "2". The second row has a "Group" value of "empty\_group", a "Description" value of "None", and an "Amount Users" value of "1". A "Create group" button is located at the top left of the table.

Group	Description	Amount Users
full_group	None	2
empty_group	None	1

The screenshot shows a Mozilla Firefox browser window with the title "test\_web\_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/group/full\_group". The main content area is titled "Group full\_group". It contains the following information:

Name	full_group
Description	
Users	• Daniel • Example User
Roles	
Permissions	• user_view_all
Permissions by roles	

At the top of the page, there are buttons for "Edit group" and "Remove". Above the table, there are navigation links: "test\_web\_app", "Example User (Logout)", "User Manager", and "WebManager".



## Permissions

The screenshot shows a Mozilla Firefox browser window with the title "test\_web\_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/permission/". 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

The screenshot shows a Mozilla Firefox browser window with the title bar "test\_web\_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/permission/user\_view\_own". The main content area is titled "Permission user\_view\_own" and contains a table with the following data:

Name	user_view_own
Description	User is allowed to view OWN user profile
Function name	belongs_current_user
Function Parameters	{'user_name': 'Name of the user, which belongs the secured object'}
Plugin	GwUsersWebManager
Currently useable?	Yes

## Roles

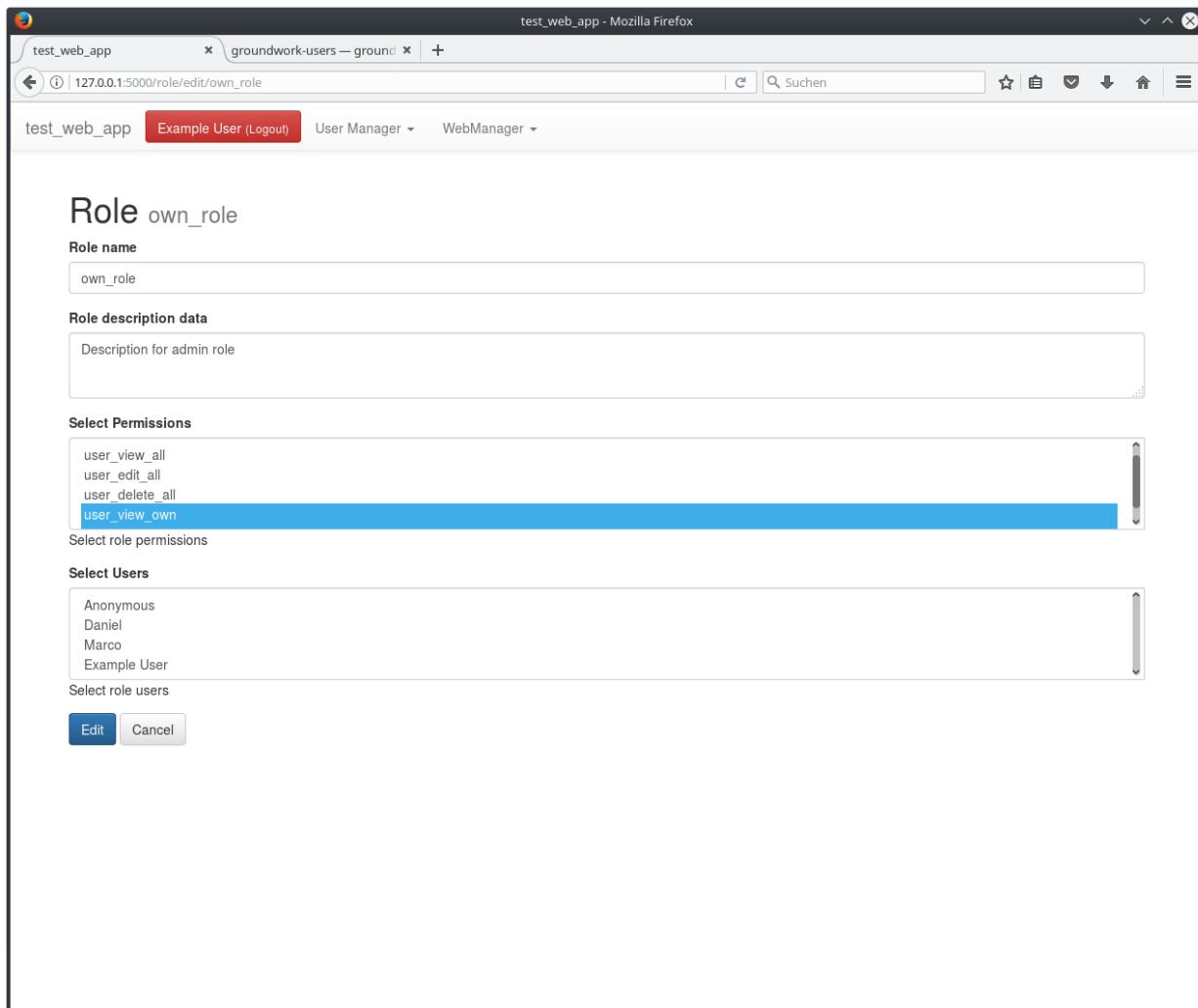
The screenshot shows a Mozilla Firefox browser window with the title "test\_web\_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/role/". The page content is titled "Roles List" and contains a table with the following data:

Role	Description	Permissions
Anonymous	special role for anonymous accounts	
empty_role	None	
admin_role	Description for admin role	<ul style="list-style-type: none"><li>● permission_no_func</li><li>● user_view_all</li><li>● user_edit_own</li><li>● permission_with_func</li><li>● user_delete_own</li><li>● user_edit_all</li><li>● user_delete_all</li><li>● user_create</li><li>● user_view_own</li></ul>
own_role	Description for admin role	<ul style="list-style-type: none"><li>● user_view_own</li><li>● user_delete_own</li><li>● user_edit_own</li></ul>

The screenshot shows a Mozilla Firefox browser window with the title bar "test\_web\_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/role/own\_role". The main content area is titled "Role own\_role". It contains a table with the following data:

Name	own_role
Description	Description for admin role
Permissions	<ul style="list-style-type: none"><li>• user_view_own</li><li>• user_delete_own</li><li>• user_edit_own</li></ul>
Users	(empty)
Plugin	test_web_app

At the top of the page, there are two buttons: "Edit role" (gray) and "Remove" (red). At the bottom left, there are navigation links: "test\_web\_app", "Example User (Logout)", "User Manager", and "WebManager".



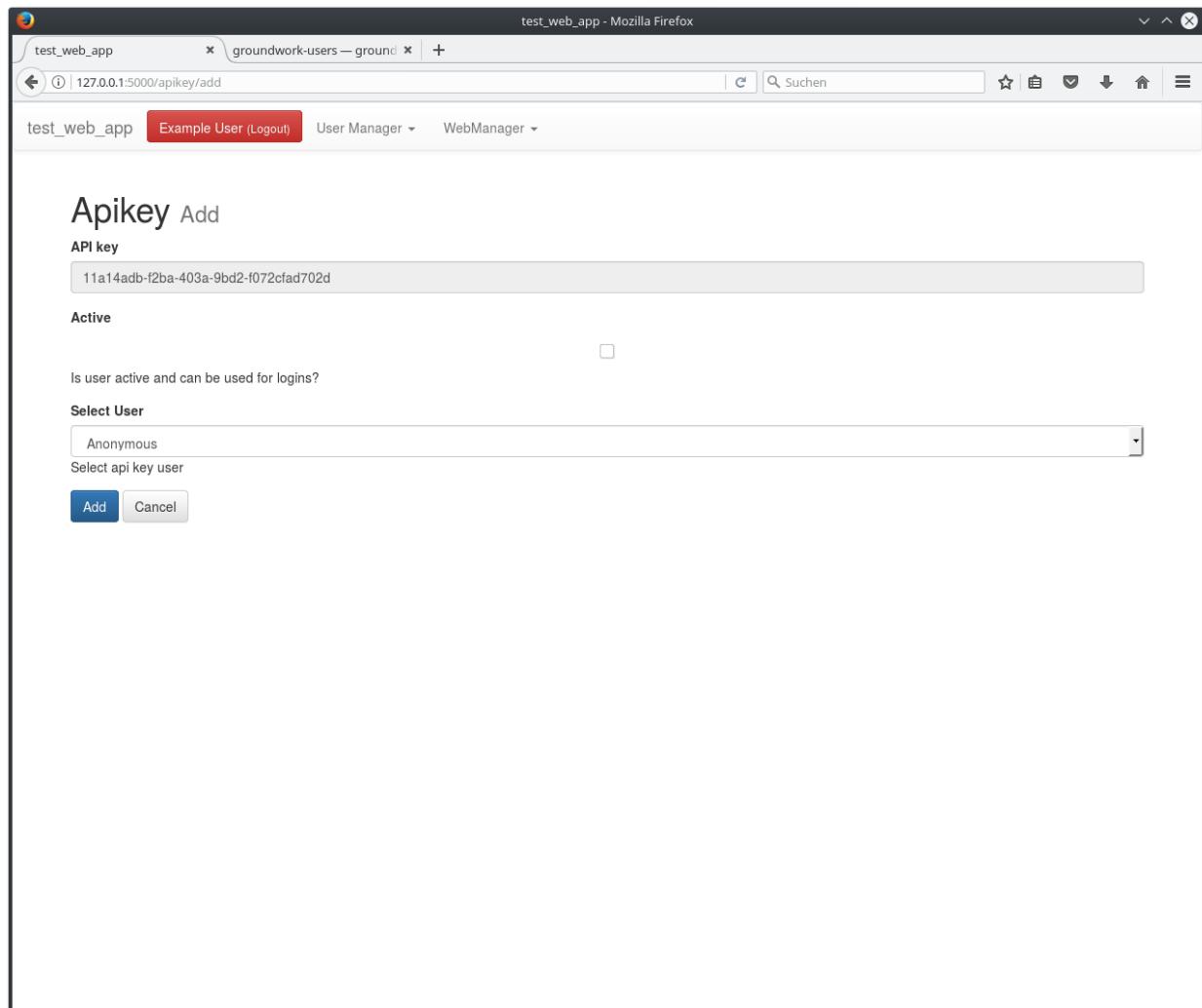
## API keys

The screenshot shows a Mozilla Firefox browser window titled "test\_web\_app - Mozilla Firefox". The address bar displays "127.0.0.1:5000/apikey/". The page content is titled "API Keys" and contains a table listing registered API keys. A button labeled "Create apikey" is visible above the table. The table has columns for ID, Key, User, Active, Last login, and Generated.

ID	Key	User	Active	Last login	Generated
1	e1a0a008-2dd3-4d11-a520-f4ba2e60bb83	Daniel	True	None	2017-06-01 07:25:32.509146
2	4c595b04-df2e-44e6-a7e8-108a360bba17	Marco	False	None	2017-06-01 07:25:32.518850
3	a95b29b8-4a39-405e-9ed1-56b8a354f98e	Marco	True	None	2017-06-01 07:25:32.525364
4	0980c0cd-cc17-41a5-8a7d-0359a51f3bf5	Example User	True	None	2017-06-01 07:27:53.486476
5	77f97811-c9a8-4edf-9389-ca39fac689e5	Example User	True	None	2017-06-01 07:27:59.232894

A screenshot of a Mozilla Firefox browser window titled "test\_web\_app - Mozilla Firefox". The address bar shows the URL "127.0.0.1:5000/apikey/0980c0cd-cc17-41a5-8a7d-0359a51f3bf5". The page content displays an API key detail page for "Example User". The page title is "API key 0980c0cd-cc17-41a5-8a7d-0359a51f3bf5". Below the title are two buttons: "Edit API key" (gray) and "Remove" (red). A table lists the following information:

ID	4
Key	0980c0cd-cc17-41a5-8a7d-0359a51f3bf5
User	<a href="#">Example User</a>
Active	True
Last login	None
Generated	2017-06-01 07:27:53.486476



## 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 to 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

---

## Python Module Index

---

**g**

`groundwork_users`, 22



---

## Index

---

### 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](#)