

[Guide](#) » [Developer Guide](#) » [Rest api](#) »

Users

/users.:format

GET

Returns a list of users.

Example:

```
GET /users.xml
```

POST

Creates a user.

Parameters:

- **user (required):** a hash of the user attributes, including:
 - **login (required):** the user login
 - **password:** the user password
 - **firstname (required)**
 - **lastname (required)**
 - **mail (required)**
 - **auth_source_id:** authentication mode id

Example:

```
POST /users.xml

<?xml version="1.0" encoding="ISO-8859-1" ?>
<user>
  <login>jplang</login>
  <firstname>Jean-Philippe</firstname>
  <lastname>Lang</lastname>
  <password>secret</password>
  <mail>jp_lang@yahoo.fr</mail>
  <auth_source_id>2</auth_source_id>
</user>
```

Response:

- 201 Created: **user was created**
- 422 Unprocessable Entity: **user was not created due to validation failures** (response body contains the error messages)

/users/:id.:format

GET

Returns the user details. You can use `/users/current.:format` for retrieving the user whose credentials are used to access the API.

Parameters:

- **include (optional):** a coma separated list of associations to include in the response:
 - **memberships**
 - **groups (added in 2.1)**

Examples:

```
GET /users/current.xml
```

Returns the details about the current user.

```
GET /users/3.xml?include=memberships,groups
```

Returns the details about user ID 3, and additional detail about the user's project memberships.

Reponse:

```
<user>
  <id>3</id>
  <login>jplang</login>
  <firstname>Jean-Philippe</firstname>
  <lastname>Lang</lastname>
  <mail>jp_lang@yahoo.fr</mail>
  <created_on>2007-09-28T00:16:04+02:00</created_on>
  <last_login_on>2011-08-01T18:05:45+02:00</last_login_on>
  <custom_fields type="array" />
```

Users
/users.:format
GET
POST
/users/:id.:format
GET
PUT
DELETE
See also

```
<memberships type="array">
  <membership>
    <project name="Redmine" id="1"/>
    <roles type="array">
      <role name="Administrator" id="3"/>
      <role name="Contributor" id="4"/>
    </roles>
  </membership>
</memberships>
<groups type="array">
  <group id="20" name="Developers"/>
</groups>
</user>
```

PUT

Updates a user.

Example:

```
PUT /users/20.xml
```

Parameters:

- **user** (required): a hash of the user attributes (same as for user creation)

DELETE

Deletes a user.

Example:

```
DELETE /users/20.xml
```

Response:

- 200 OK: user was deleted

See also

- The [Memberships API](#) for adding or removing a user from a project.
- The [Groups API](#) for adding or removing a user from a group.