

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

History

Project Memberships

/projects/:project_id/memberships.:format

GET

Returns a paginated list of the project memberships.
:project_id can be either the project numerical id or the project identifier.

Examples:

```
GET /projects/1/memberships.xml
GET /projects/redmine/memberships.xml
```

Project Memberships

/projects/:project_id/memberships.:format

GET

POST

/memberships/:id.:format

GET

PUT

DELETE

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<memberships type="array" limit="25" offset="0" total_count="3">
  <membership>
    <id>1</id>
    <project name="Redmine" id="1"/>
    <user name="David Robert" id="17"/>
    <roles type="array">
      <role name="Manager" id="1"/>
    </roles>
  </membership>
  <membership>
    <id>3</id>
    <project name="Redmine" id="1"/>
    <group name="Contributors" id="24"/>
    <roles type="array">
      <role name="Contributor" id="3"/>
    </roles>
  </membership>
</memberships>
```

Latest releases

[2.0.3 \(2012-06-18\)](#)
[1.4.4 \(2012-06-18\)](#)

Resources

[User's Guide](#)
[Changelog](#)
[FAQ, HowTo's](#)
[Plugins, Themes](#)

Wiki

[Start page](#)
[Index by title](#)
[Index by date](#)

```
</roles>
</membership>
<membership>
  <id>4</id>
  <project name="Redmine" id="1"/>
  <user name="John Smith" id="27"/>
  <roles type="array">
    <role name="Developer" id="2" />
    <role name="Contributor" id="3" inherited="true" />
  </roles>
</membership>
</memberships>
```

Notes:

- The membership owner can be either a user or a group ([Groups API](#) is added in Redmine 2.1)
- In the above example, the `inherited="true"` attribute on the last role means that this role was inherited from a group (eg. John Smith belongs to the Contributors group and this group was added as a project member). John Smith's membership can not be deleted without deleting the group membership first.
- The memberships of a given user can be retrieved from the [Users API](#).

POST

Adds a project member.

Parameters:

- membership (required): a hash of the membership attributes, including:
 - user_id (required): the numerical id of the user
 - role_ids (required): an array of roles numerical ids

Example:

```
POST /projects/redmine/memberships.xml
```

```
<membership>
  <user_id>27</user_id>
  <role_ids type="array">
    <role_id>2</role_id>
  </role_ids>
</membership>
```

Response:

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

/memberships/:id.:format

GET

Returns the membership of given :id.

Examples:

```
GET /memberships/1.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<membership>
  <id>1</id>
  <project name="Redmine" id="1"/>
  <user name="David Robert" id="17"/>
  <roles type="array">
    <role name="Developer" id="2"/>
    <role name="Manager" id="1"/>
  </roles>
</membership>
```

PUT

Updates the membership of given :id. Only the roles can be updated, the project and the user of a membership are read-only.

Parameters:

- membership (required): a hash of the membership attributes, including:
 - role_ids (required): an array of roles numerical ids

Example:

```
PUT /memberships/2.xml
```

```
<membership>
  <role_ids type="array">
    <role_id>3</role_id>
    <role_id>4</role_id>
  </role_ids>
</membership>
```

Response:

- 200 OK: membership was updated
- 422 Unprocessable Entity: membership was not updated due to validation failures (response body contains the error messages)

DELETE

Deletes a memberships.

Memberships inherited from a group membership can not be deleted. You must delete the group membership.

Parameters:

none

Example:

```
DELETE /memberships/2.xml
```

Response:

- 200 OK: membership was deleted
- 422 Unprocessable Entity: membership was not deleted