

GitLab SUPSI User manual

IT Service

Summary

1.	Introduction	3
2.	Purpose of the document	3
3.	Accessing GitLab instances	4
3.1	Edit personal profile	5
3.2	Two-Factor Authentication (2FA)	5
4.	Users, Groups and Projects	7
4.1	Users	7
4.2	Groups	7
4.3	Projects	8
4.4	Accessibility levels of Groups and Projects	9
5.	Nomenclatures	9
6.	Administrative procedures	10
6.1	Enable SUPSI user collaborator	10
6.2	External user creation	10
6.3	Subgroup creation	11
6.4	Add members to a subgroup	13
6.5	Project creation	14
6.6	Enabling a student on a project by a faculty member	16
7.	Support	17

1. Introduction

The growing needs of SUPSI in managing different types of software projects, according to the latest standards, led to the introduction of the GitLab platform. In order to meet the different needs of users, two different installations have been realized:

- **Educational** <https://gitlab-edu.supsi.ch/> - For educational projects
- **Core** <https://gitlab-core.supsi.ch/> - For business projects.

2. Purpose of the document

This document introduces users to the use of GitLab by showing how to access the platform and increase the security of their profile. It also provides a general overview of: the main elements, nomenclatures and administrative modes.

3. Accessing GitLab instances

As mentioned in the introduction, there are two instances of GitLab to address different project needs:

- **Educational** <https://gitlab-edu.supsi.ch/>
- **Core** <https://gitlab-core.supsi.ch/>

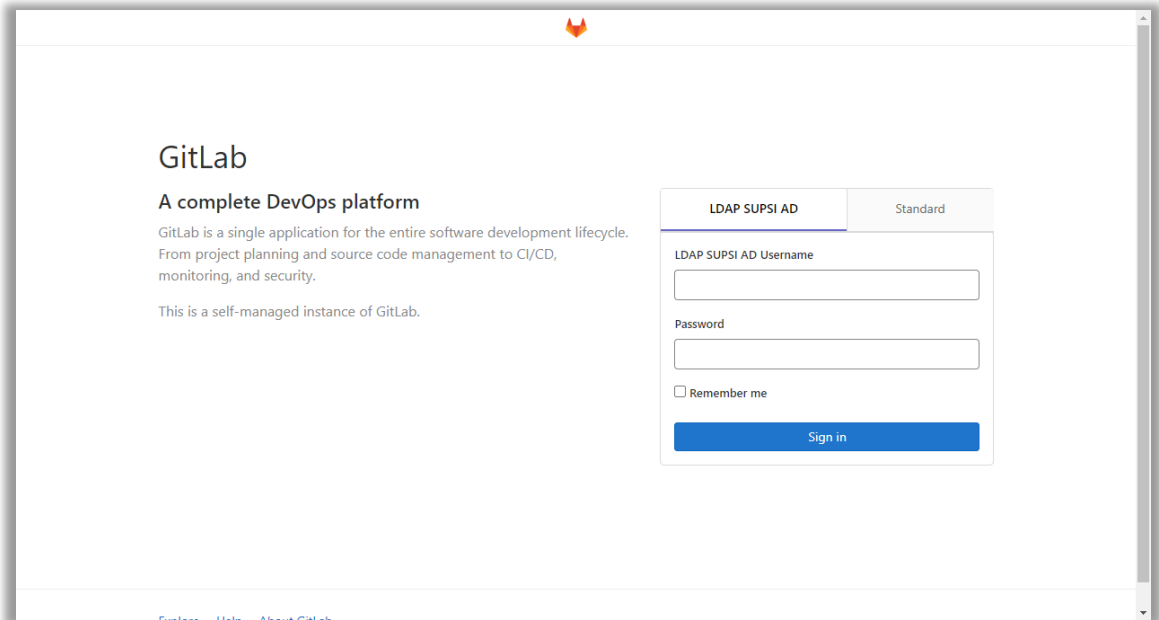


Figure 1 - Logging in to GitLab

SUPSI collaborators and students can access by selecting the *LDAP SUPSI AD* section and entering their academic credentials name.surname@supsi.ch and relative password (those of the Windows account used to access the Collaborators Portal, Microsoft Teams, etc.).

Standard authentication is available to give access to external users.

3.1 Edit personal profile

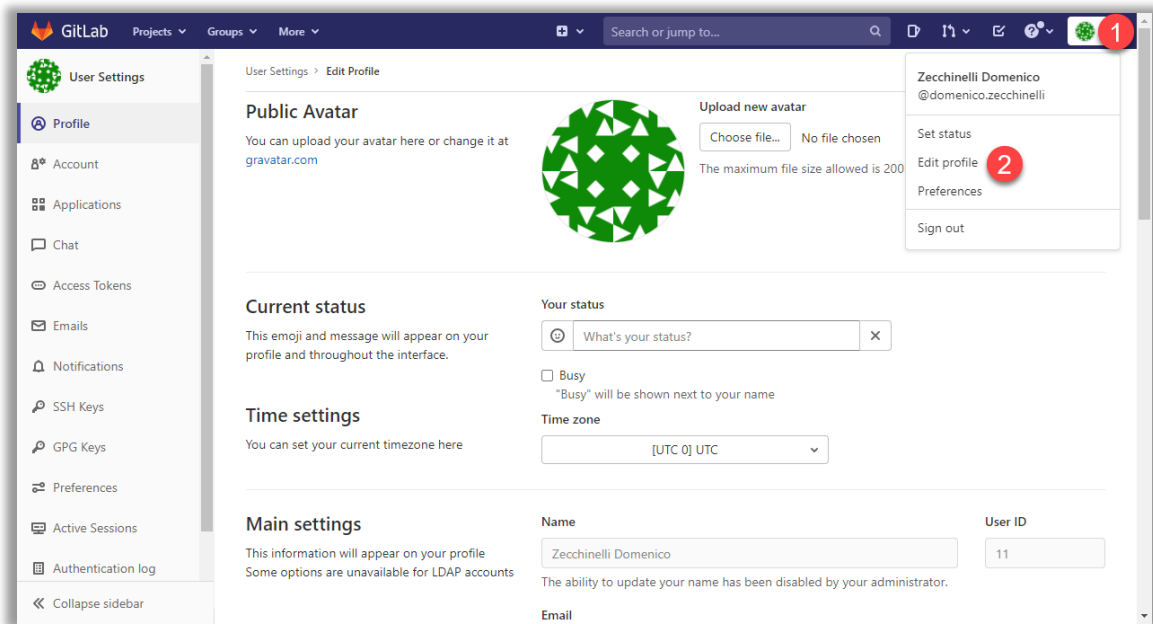


Figure 2 - Edit user profile

Once you have logged in to the platform you can edit your profile by clicking the 1 and 2 buttons (Figure 2 - Edit user profile).

For more details about user profile management refer to the official documentation:

<https://docs.gitlab.com/ee/user/profile/>.

3.2 Two-Factor Authentication (2FA)

Two-Factor Authentication is a process that allows to identify the user with two passwords: the first is the static password associated with the user's account; the second credential is temporary, produced by a third-party device or software. The most popular method of obtaining the second temporary login credential is through an "authenticator app," for example: Google Authenticator, Authy, Duo Mobile and Lastpass.

To activate the two-factor authentication procedure on GitLab, you first need an authenticator app. Then you need to access the settings of your profile and, as described in the chapter [Edit personal profile](#), click on the buttons 1 and 2 as shown in the image below (Figure 3 - 2FA activation (1)).

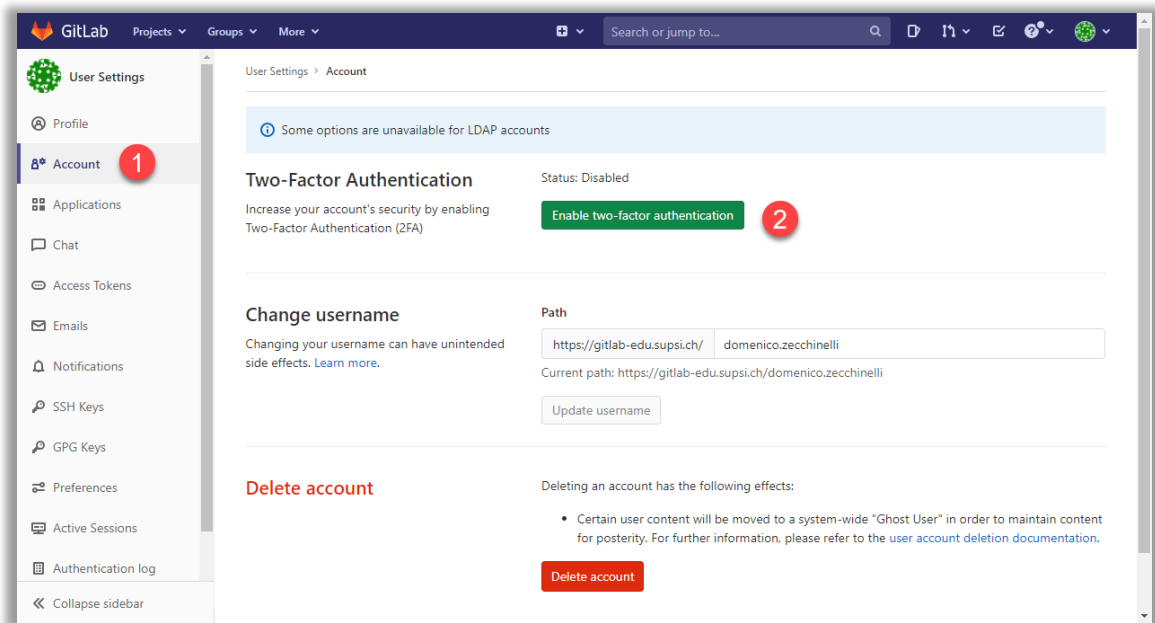


Figure 3 - 2FA activation (1)

Then, refer to the image below (Figure 4 - 2FA activation (2)) and using the authenticator app:

1. Read the QR-Code with the camera
2. Enter the code generated by the app
3. Complete the procedure.

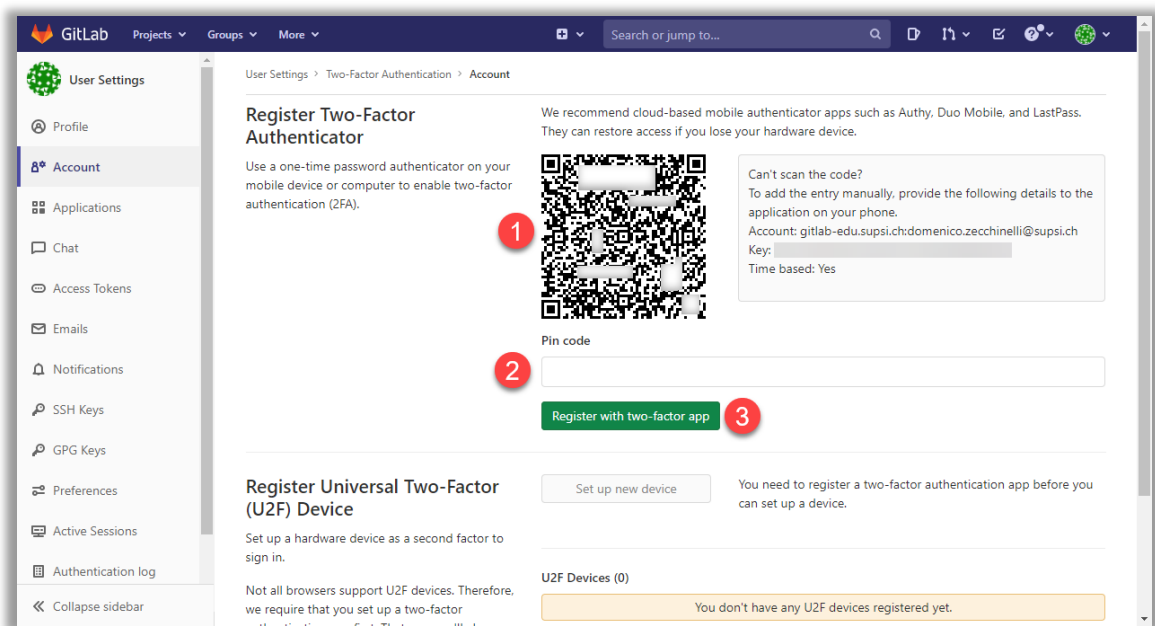


Figure 4 - 2FA activation (2)

Note: at the end of the procedure, the system displays codes that can be downloaded or copied and that are used to access your account in case you can no longer use the authenticator app to generate the temporary password. **Please save and store these codes carefully.**

4. Users, Groups and Projects

In this chapter are briefly described the main elements present in GitLab reporting, for each of them, the pages of the official guide where you can find detailed and always updated information.

4.1 Users

Each GitLab user has a personal profile in which some personal information, customizations and activities performed on the platform are saved.

See <https://docs.gitlab.com/ee/user/> for more details about users.

4.2 Groups

Groups are elements that allow to aggregate and manage users, projects and related elements, such as: security, requirements, issues, merge requests and more. They also let you view different types of statistics and create wikis (simplified web pages available in *Educational* version) to collect the most important information and improve collaboration between users.

An absolutely interesting feature of groups is the possibility of creating up to 20 levels of subgroups, thus creating hierarchies with which to manage user permissions to access resources.

Note 1: Subgroups inherit the characteristics of the groups from which they are derived.

If the parent group is *Private* (visible only to its members), the subgroup can be neither *Public* (visible to all), nor *Internal* (visible to logged in users), but only *Private*.

Note 2: When a user is added to the members of a group, they have access to the contents of that group and the contents of all subgroups.

Summary example

Suppose we have users A and B and the following hierarchy of groups exists with, in parentheses, the level of visibility and member users:

- **Group 1** (*Public*)
 - **Group 2** (*Private with user A in the members*)
 - **Group 4** (*Private*)
 - **Group 5** (*Private with user B in the members*)
 - **Group 3** (*Private with user B in the members*)
 - **Group 6** (*Private*)

User **A** is able to see:

- **Group 1** as *Public*
- **Group 2** as he has been granted access in an explicit manner
- **Group 4** and **Group 5** as *Private* as Group 2 that contains them and which is accessible by the user.

User **B** is able to see:

- **Group 1** as *Public*
- **Group 2** as part of the path to get to view Group 5. Projects and other groups within this group, such as Group 4, are not visible
- **Group 5** as he has been granted access in an explicit manner
- **Group 3** as he has been granted access in an explicit manner
- **Group 6** as *Private* as Group 3 that contains it and which is accessible by the user.

See <https://docs.gitlab.com/ee/user/group/> for more details about groups.

<https://docs.gitlab.com/ee/user/permissions.html#group-members-permissions> for information on managing permissions in groups.

4.3 Projects

Projects are items where to store software source code. Users enabled to work in a project can collaborate in work planning, development, issue tracking, testing and release activities via CI/CD (Continuous Integration and Continuous Delivery).

Note: when you want to grant access to a user to allow them to work on a project, especially if a teacher wants to authorize a student:

- The project must be *Private*
- The user should be added to the project and not to the group that contains the project
- Access should be granted individually to each user who is to work on the project and not to a group of users, otherwise all members of the group will be able to work on the project
- Assuming you decide to add users to the members of a group, they should only be SUPSI teachers, assistants or collaborators, not students.

Responsibility for what happens within groups and projects created by a user/teacher, is the responsibility of the user/teacher.

See <https://docs.gitlab.com/ee/user/project/> for more details about projects.

<https://docs.gitlab.com/ee/user/permissions.html#project-members-permissions> for information on managing permissions in projects.

4.4 Accessibility levels of Groups and Projects

Groups and Projects have three different levels of accessibility that allow them to be:

- **Public** Visible to everyone, even unauthenticated users, at */public*.
Sample URL of public projects for the *Educational* version:
<https://gitlab-edu.supsi.ch/public>
- **Internal** Visible to all authenticated users who are not *External*
- **Private** Visible only to users who are members.

5. Nomenclatures

For the naming of Users, Groups and Projects use only:

- Unaccented characters
- Numbers
- ".", "-" and "_".

Important: Do not use the sequence "__-" because it causes problems with the docker registry.

Note: In group names, in order to have the URL as similar as possible to the name, replace spaces with the "_" character.

Example: Group_name_example

6. Administrative procedures

This chapter describes the procedures involved in handling some specific requirements.

6.1 Enable SUPSI user collaborator

To enable user and configure access to a SUPSI collaborator:

1. Ask the collaborator to log in with his institutional credentials to the instance of interest. The access links are:
 - o **Core** <https://gitlab-core.supsi.ch/>
 - o **Educational** <https://gitlab-edu.supsi.ch/>
2. Send an e-mail to IT Service to configure access:
 - a. Addressee: helpit@supsi.ch
 - b. Subject: "GITLAB **User activation** request"
 - c. Body:
 - Reference GitLab instance (*Core* or *Educational*)
 - Name
 - Surname
 - E-mail address
 - Role in SUPSI
 - Department
 - Institute.

6.2 External user creation

To request the creation of a user account for an external user send an e-mail to the IT Service:

- a. Addressee: helpit@supsi.ch
- b. Subject: "GITLAB **User creation** request"
- c. Body:
 - Reference GitLab instance (*Core* or *Educational*)
 - Name
 - Surname
 - E-mail address
 - Department
 - Institute
 - Specify whether the user can see and request access to projects and groups where he is not part of the membership.

6.3 Subgroup creation

The creation of subgroups is allowed only for users who are members of a group with a *Maintainer* role or higher.

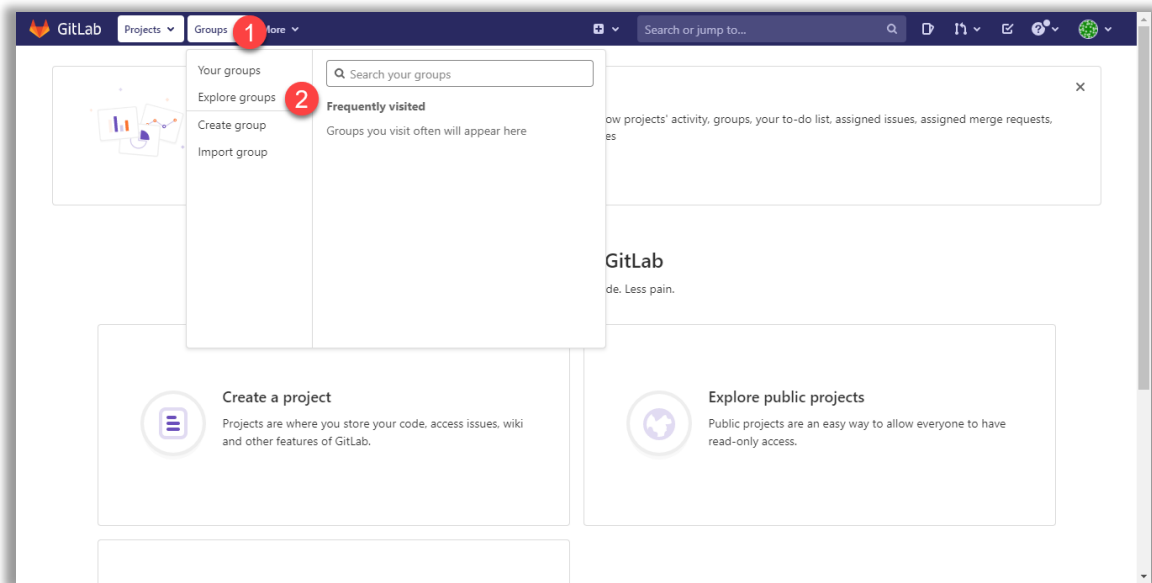


Figure 5 - Explore groups

Referring to the image above (*Figure 5 - Explore groups*), click *Groups* (1) and *Explore groups* (2) to view the existing *Public*, *Internal* and *Private* groups. Click on the groups you are interested in to position yourself within the group where you want to create your subgroup.

If the parent group does not exist, you can ask the service contact person at your institute to create it. Even if base group of the institute does not exist, the one in the `DIPARTMENT-ISTITUTE` format, send an e-mail to the IT Service:

- a. Addressee: helpit@supsi.ch
- b. Subject: "GITLAB **Group creation** request"
- c. Body:
 - Reference GitLab instance (*Core* or *Educational*)
 - Group name
 - Department
 - Institute
 - First and last name of the user to be set as *Maintainer* or *Owner* (usually the applicant).

If the parent group exists, with reference to the image below (*Figure 6 - Subgroup creation (1)*), click *New subgroup (1)*.

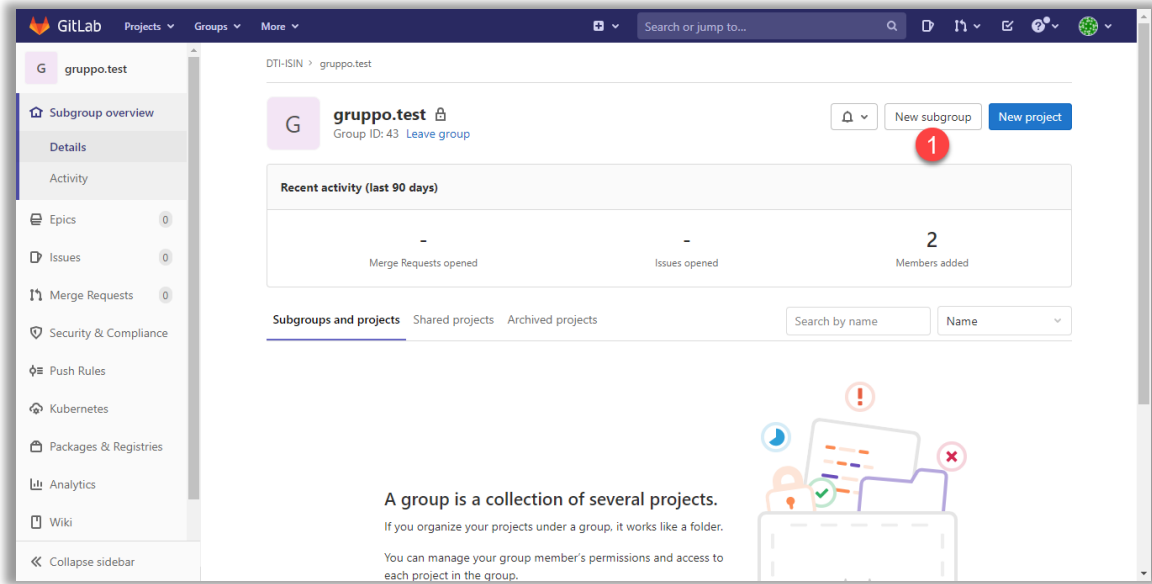


Figure 6 - Subgroup creation (1)

As shown in the next image (*Figure 7 - Subgroup creation (2)*):

1. Enter the name of the group (**Note:** Enter the name in the *Group name* field, not in *Group URL*);
2. Create the group.

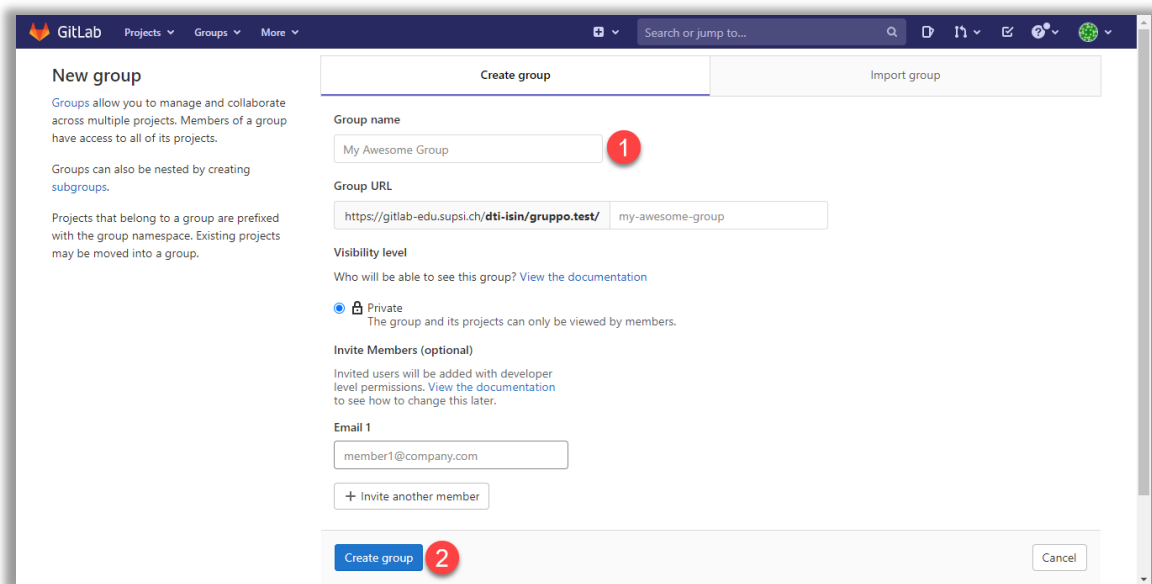


Figure 7 - Subgroup creation (2)

Note: As indicated in the *Nomenclatures* chapter, you can only use:

- Unaccented characters
- Numbers
- “.”, “-” and “_”.

Spaces must be replaced with “_” to have the URL match the name as closely as possible.

Important: Do not use the sequence “_ _” because it causes problems with the docker registry.

6.4 Add members to a subgroup

In order to add members to a subgroup, the user must be the *Owner* of the subgroup. If this is not the case, consider creating a more specific subgroup or send an e-mail to the IT Service:

- a. Addressee: helpit@supsi.ch
- b. Subject: "GITLAB Add member to the group"
- c. Body:
 - Group URL
 - Name and surname of the user to add
 - Role to be assigned to the user (*Guest, Reporter, Developer, Maintainer* or *Owner*)
 - Possible end date of access privileges.

If you are the *Owner* of the subgroup, with reference to the image below (*Figure 8 - Adding members to a group*), to add members you just need to position yourself inside it and:

1. Select the page for managing members
2. Type the first name, last name or e-mail address of the user you want to add
3. Select the role
4. Enter any date for the end of the granting of access
5. Confirm that the user has been added to the group members.

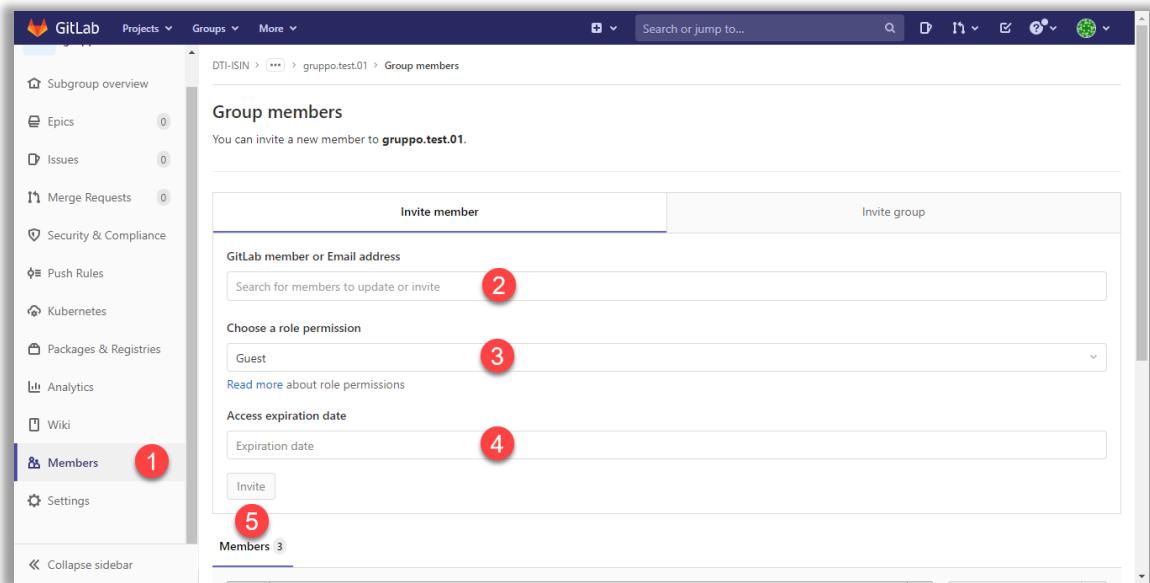


Figure 8 - Adding members to a group

6.5 Project creation

To create a project it is first necessary that the correct subgroup exists and that the user is a member with a *Developer* role or higher. If the subgroup does not exist refer to the chapter [Subgroup creation](#).

With reference to the following image (*Figure 9 - Project creation (1)*): position yourself in the subgroup where you want to create the project and click the *New project (1)* button.

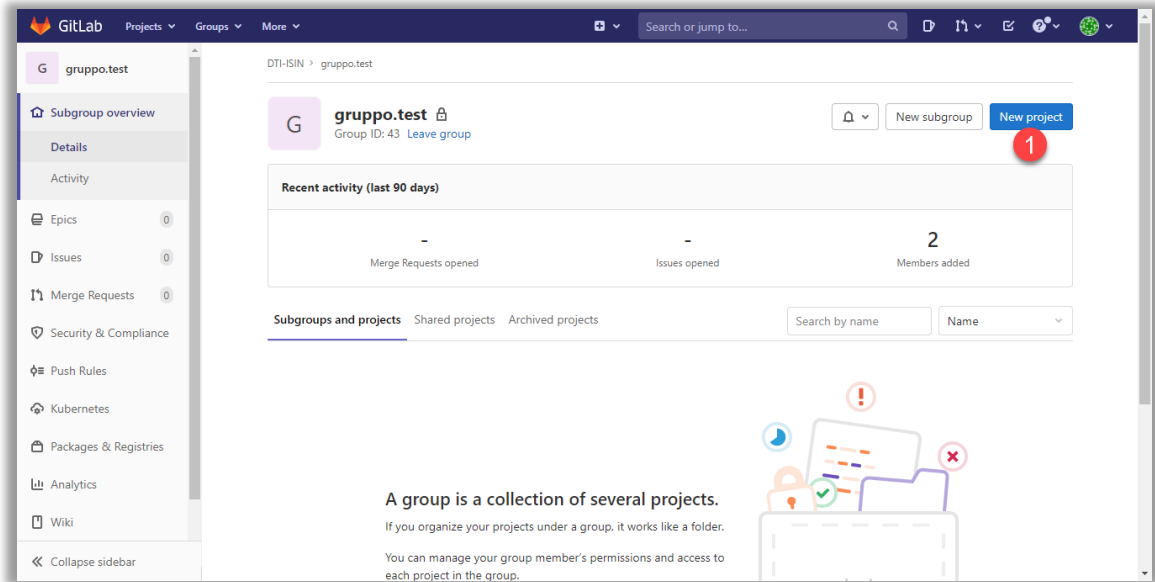


Figure 9 - Project creation (1)

Then, as shown in the next image (*Figure 10 - Project creation (2)*), click *Create blank project (1)*.

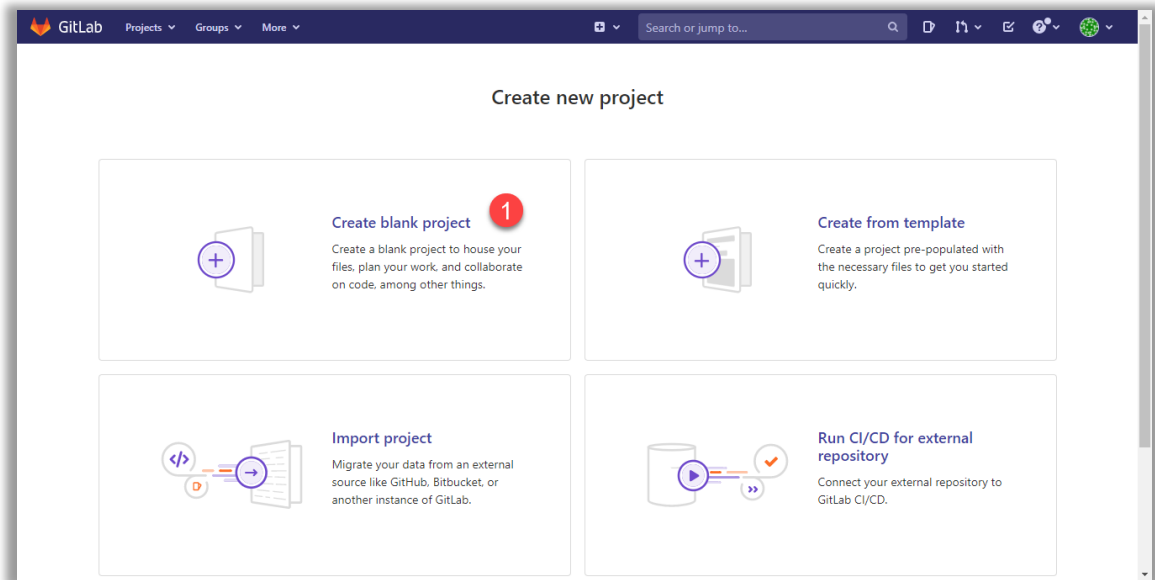


Figure 10 - Project creation (2)

Refer to the image below (*Figure 11 - Project creation (3)*):

1. Enter the name of the project using only:
 - a. Unaccented characters
 - b. Numbers
 - c. ".", "-", and "_".

Important: Do not use the sequence "___" because it causes problems with the docker registry.

2. Create the project.

The screenshot shows the GitLab interface for creating a new project. The form is titled "New project · Create blank project". It includes the following fields and options:

- Project name:** A text input field containing "My awesome project", marked with a red circle "1".
- Project URL:** A dropdown menu showing "https://gitlab-edu.supsi.ch/dti-isin/gruppo.test".
- Project slug:** A text input field containing "my-awesome-project".
- Project description (optional):** A text area for entering a description.
- Visibility Level:** A radio button selected for "Private".
- Initialize repository with a README:** An unchecked checkbox.
- Buttons:** A green "Create project" button (marked with a red circle "2") and a "Cancel" button.

Figure 11 - Project creation (3)

Finally, if you want to configure access privileges for other users, proceed as shown in the image below (*Figure 12 - Project creation (4)*):

1. Select the page for managing members
2. Type the first name, last name or email address of the user you want to add
3. Select the role
4. Enter any date for the end of the granting of access
5. Confirm that the user has been added to the project members.

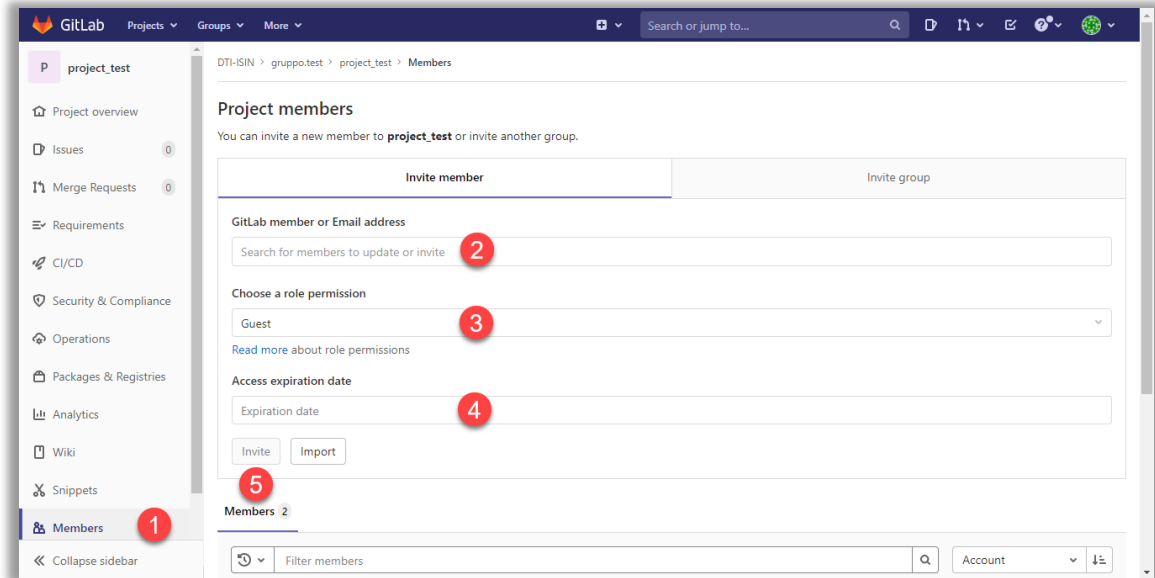


Figure 12 - Project creation (4)

6.6 Enabling a student on a project by a faculty member

Each teacher has a personal group, corresponding to his or her first and last name, in which he or she can create subgroups and projects needed for teaching activities.

To enable a student to work on a project, it is necessary:

1. Create the subgroup in which to create the project (see chapter [Subgroup creation](#));
2. Create the project (see chapter [Project creation](#));
3. Add student to project members with *Developer* role.

Example:

Assuming that the teacher Mario Rossi of the ISIN institute (of the DTI) has to allow the student Luca Bianchi to work on the graduation project "Test project", the final structure of the groups will be:

- . **DTI-ISIN** (created by administrator)
- .. **mario.rossi** (teacher's personal group created by administrator)
- ... **Degree_projects** (group to manage degree projects created by the teacher)
- **Test_project** (project created by the teacher in which the student is member)

With this type of structure, when the student logs into the system, the student Luca Bianchi is enabled to work only on the *Test_project*.

Note: students can only access the *Educational* instance.

7. Support

For all support requests and clarifications needed to use the platform, please send an e-mail to the IT Service:

- a. Addressee: helpit@supsi.ch
- b. Subject: "GITLAB – ***Scope of request***"
- c. Body: Useful details to better explain the problem.