

# Git Service

---

November 2013

## Contenido

- 1. Introduction ..... 1
- 2. Service access..... 1
  - 2.1 https access ..... 2
- 3. Requirements..... 2
- 4. Initial project configuration ..... 2
- 5. Project modifications ..... 3
- 6. Save changes to local disk ..... 4
- 7. Upload changes to git-rdlab server..... 5
- 8. External links..... 5

## 1. Introduction

The following manual shows how to access the RDLab Git repository service, and how to perform key tasks in text mode (command line).

## 2. Service access

In order to get access to the service, a username and password is needed. A repository name can be requested too; if the repository name is not specified in the request, the default repository name will be the user's username. All these can be requested at <http://rdlab.lsi.upc.edu/index.php/es/servicios/peticionacceso.html>



Once received, the user will have a repository for his projects. This repository is initialized by RDlab with a file called “readme.rdlab”.

The repository is accessed through the Git client and the URL:

```
http://git-rdlab.lsi.upc.edu/git/<repository>.git
```

## 2.1 https access

Secure connection is also available through the URL:

```
https://git-rdlab.lsi.upc.edu/git/<repositori>.git
```

May you find the following error:

```
error: server certificate verification failed.  
CAfile: /etc/ssl/certs/ca-certificates.crt CRLfile:  
none while accessing https://git-  
rdlab.lsi.upc.edu/git/<repository>.git/info/refs  
  
fatal: HTTP request failed
```

it is necessary to run the following command:

```
git config --global http.sslverify "false"
```

## 3. Requirements

IMPORTANT: In order to use RDlab Git service it is **mandatory** to use a git client whose version is 1.6.6 or above.

To check the version of the git client installed, run “git --version”.

## 4. Initial project configuration

The first step is to set up the project at the user’s computer. A new folder must be created. This folder will be used to hold a copy of the project that will synchronize with the server “git-rdlab.” For our example purpose, let’s suppose that it is the folder “/home/user/myProject”. An initial import must be performed from this new folder using the command “git clone”:

```
$ git clone http(s)://<username>@git-  
rdlab.lsi.upc.edu/ git/<repository>.git
```

The system will prompt us for our password and then will show this output:

```
Initialized empty Git repository in  
/home/user/myProject/<repository>/.git
```

This command will have created a folder named <repository\_name> in which we will find a file "readme.rdlab" and a ".git" folder that contains the information that Git needs for version control.

Note: to avoid having to introduce the user and password on each "git commit" and "git push" execution (see section 7) it is possible to edit the ".netrc" file at the home directory (it should be created if it doesn't exist) and include the following lines:

```
Machine    git-rdlab.lsi.upc.edu  
login      <username>  
password   <password>
```

In this case, the command would be:

```
$ git clone http(s)://git-  
rdlab.lsi.upc.edu/git/<repository>.git
```

And the system will not prompt for a password.

## 5. Project modifications

Whenever a file of the project is modified or new files are added, it is necessary to indicate Git that these files must be added to the set of files under version control.

To do so, the "git add <file>" command is used. Use "git add ." to add all files in the folder to the version control. Let's suppose that a new "HelloWorld.java" file has been added to the project:

```
$ git add HelloWorld.java
```



