

# Git Service

/rdlab - rdlab@cs.upc.edu

October 5, 2020

## Contents

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

# 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: <https://rdlab.cs.upc.edu/contact-us/>

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/<repository>.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":

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

## 6 Save changes to local disk

The "git commit" command creates a new local revision of the project. It is mandatory to include a commit message. If we do not use the "m" flag, git will open our default text editor to let us write the message.

```
git commit -m "added HelloWorld.java"
```

The system will show the following output:

```
[506ac89] Added HelloWorld.java
Committer: user <user@lsi.upc.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate. You can suppress this message by setting them
explicitly:
```

```
git config --global user.name "Your Name"
git config --global user.email you@example.com
If the identity used for this commit is wrong, you can fix
it with:
git commit --amend --author='Your Name <you@example.com>'
```

```
1 files changed, 1 insertions(+), 0 deletions(-)
   create mode 100644 HelloWorld.java
```

Note: Contrary to the svn behavior, when modifying a project file it is not automatically included in the next commit: it is necessary to execute the "git add" command (see chapter 5) even in this situation. An alternative is to use the "git commit -a" command, which is equivalent to used "git add ." followed by "git commit".

This command commits modified files, not newly added ones. To know which changes will be added at the next commit the "git status" command may be executed, which will show which files will be included in the next revision and which ones will be excluded.

## 7 Upload changes to git-rdlab server

The "git commit" command only creates a local revision. In order to upload changes to the git-rdlab server, issue the "git push" command:

```
git push
```

Note: only the changes saved with the last "git commit" command will be uploaded to the server .The system will show the following output:

```
Fetching remote heads...
  refs/
  refs/heads/
  refs/tags/
updating 'refs/heads/master'
  from b6a55fa54e77b5c9aa56eb1ad9ea60d4cad2b1ee
  to   506ac89f2e9a66178e4b3d4471c9e3d97770cb97
sending 3 objects
done
Updating remote server info To
http://git-rdlab.lsi.upc.edu/git/<repository_name>
b6a55fa..506ac89 master -> master
```

If the system shows the following error:

```
error: The requested URL returned error: 401
while accessing
  http://git-rdlab.lsi.upc.edu/git/<repository>/info/refs
```

it is necessary to check that we have introduced our credentials (user and password) correctly , even at the command line or at the “.netrc” file, as shown in section 4.

## 8 External links

1. Git, oficial Page:  
<http://git-scm.com/>
2. Online book "Pro Git professional version control":  
<http://progit.org/book/>