

TDDD04 – Git tips and tricks for IDE

Git will be used to connect to Gitlab, either over SSH or HTTPS with a login (username and password). SSH will be used for the integrated VCS (Version Control System), which manage the changes in your documents and code, in Eclipse and HTTPS will be used for the integrated VCS in Visual Studio. If you feel comfortable using `git` from the command line and don't want to use the integrated VCS, you can stop reading here.

By using Git with SSH you will not need to enter your username and password every time you want to connect to Gitlab (push, pull etc.). The drawback is that you will need to create a SSH key and add it to Gitlab.

The reason why you need to use HTTPS for Visual Studio is because Visual Studio doesn't support SSH.

You can see the difference between an SSH-address and HTTPS-address as follow:

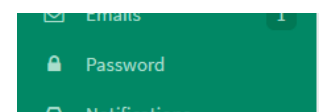
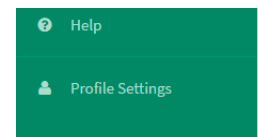
<git@gitlab.ida.liu.se:samle75/triangle.git> <--- SSH

<https://gitlab.ida.liu.se/samle75/triangle.git> <--- HTTPS

Create a password for HTTPS

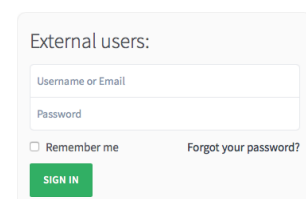
The first time you log on to gitlab.ida.liu.se, a Gitlab account will be set up for you. There will be no Gitlab-specific password set as the authentication through the web interface uses your LiU credentials. However, to access Git repositories at Gitlab through HTTPS, you will have to set a password manually in Gitlab so that your IDE may provide it to access repositories. To create one, you need to do the following after login into Gitlab:

1. Press the link **Profile Settings**
2. Press the link **Password**
3. Press the link **Forget your password?**
4. It is important to log out now
5. You will get an email with instructions to create a new password. Follow the instructions in the mail.



A password to use `git` over HTTPS has now been created. You can check if it works by logging into Gitlab here:

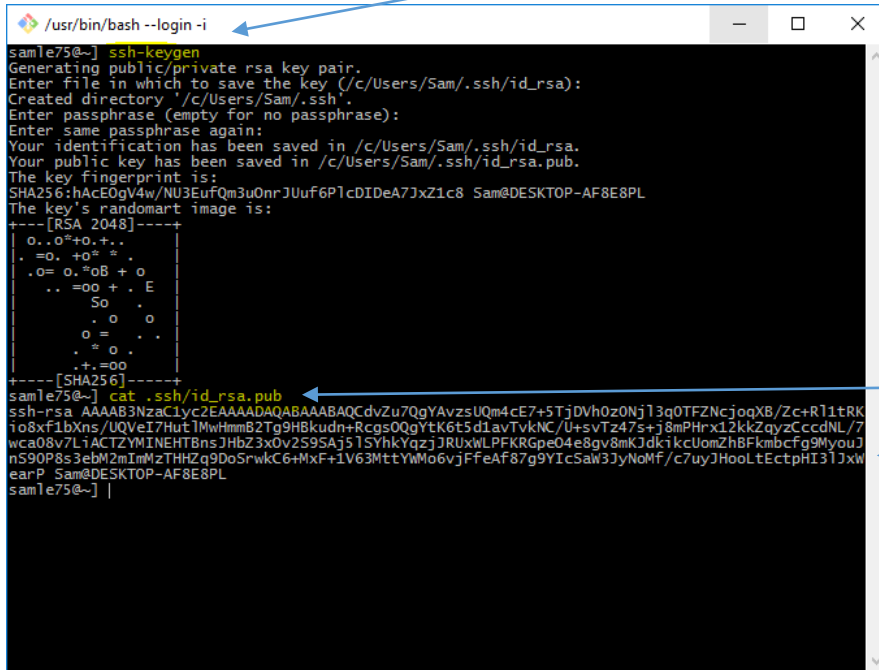
Use your LiU-id as username.

A screenshot of the Gitlab login form. It has a light gray background and a white border. At the top, it says 'External users:'. Below that, there are two input fields: 'Username or Email' and 'Password'. Under the 'Password' field, there is a checkbox labeled 'Remember me' and a link labeled 'Forgot your password?'. At the bottom, there is a green button labeled 'SIGN IN'.

Connect a computer to Gitlab with SSH

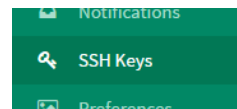
To create and add the SSH key to Gitlab you need to do the following:

1. Open **Git Bash** (on Windows) or a Terminal (on MacOs/Linux)
2. Write the command **ssh-keygen**
3. Press the **Enter** key for default values
4. Write the command **cat .ssh/id_rsa.pub**



```
/usr/bin/bash --login -i
samle75@~] ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Sam/.ssh/id_rsa):
Created directory '/c/Users/Sam/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Sam/.ssh/id_rsa.
Your public key has been saved in /c/Users/Sam/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:hAcE0gV4w/NU3EufQm3u0nrJUuf6PlcDIdeA7JxZ1c8 Sam@DESKTOP-AF8E8PL
The key's randomart image is:
+-----[RSA 2048]-----+
|  o..o^+o+.+. |
| .=.o.+o^+^  |
| .o=.o.^oB+.o |
| ..=oo+. .E  |
|    So       |
|    .o.o     |
|    o=.     |
|    ^oo     |
|    +.oo    |
|    +.oo    |
+-----[SHA256]-----+
samle75@~] cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCDvZu7QgYAvzsUQm4cE7+5TjDVh0z0Nj13oTFZncjoqXB/Zc+R1tRK
io8xF1bXns/UQVeI7Hut1MwHmmb2Tg9HBkudn+Rcgs0QgYtK6t5d1avTvkNC/U+svTz47s+18mPHrXl2kkZqyzCcedNL/7
wca08v7L1ACTZYMINEHTBnsJHbz3x0v2S9SAj51SYrkYqzjJRUxwLpFKRGpe04e8gv8mKJdkiKcUomZhBFkmbcFg9MyouJ
nS90P8s3ebM2mTmMzTHHZg9DoSrwcC6+MxF+1V63MtYmMo6vjFfeAf87g9YIcSan3JyNoMf/c7uyJHooltEctpHI31Jxw
earP Sam@DESKTOP-AF8E8PL
samle75@~] |
```

5. Copy the key and paste it into Gitlab by pressing **Profile Settings** and then **SSH Keys**.

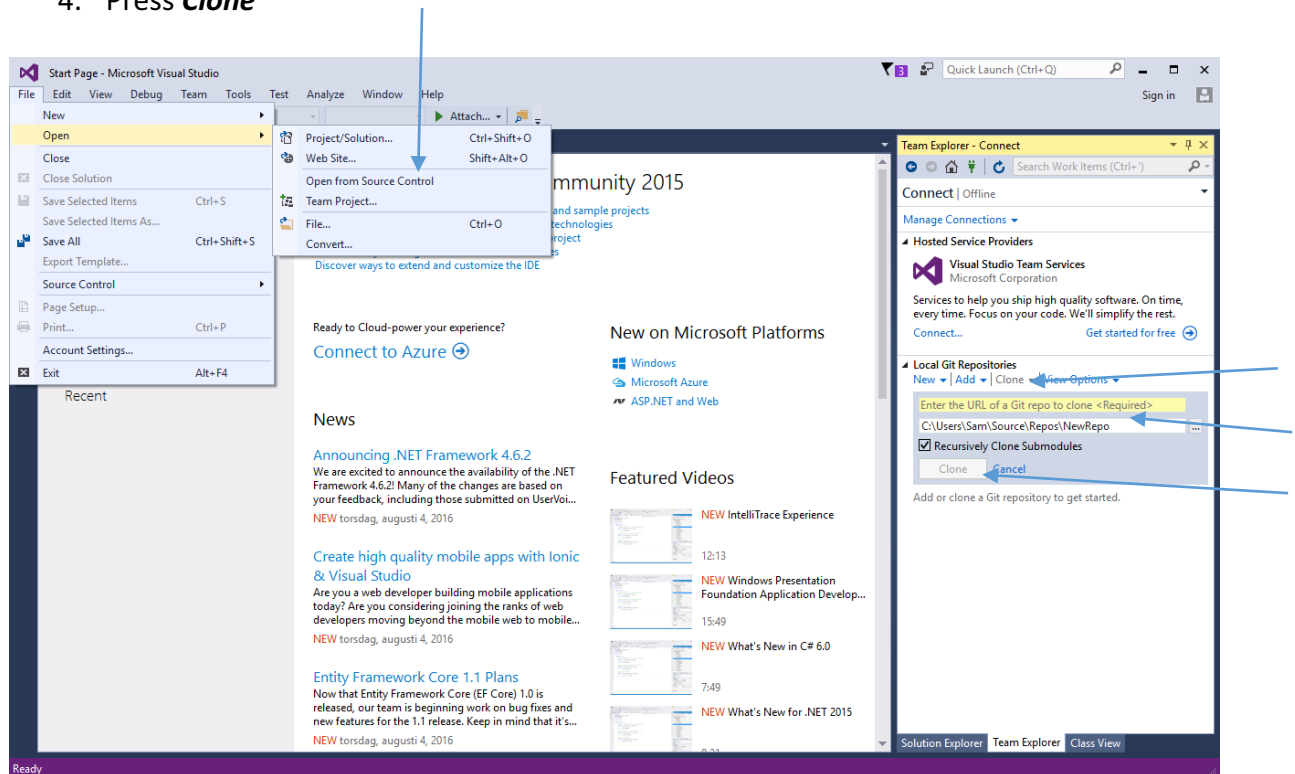


Command line Git should work from now on. You can try by enter the following command in Git Bash: **git clone git@gitlab.ida.liu.se:TDDD04/triangle.git**
If everything goes well you will have a folder with the name triangle.

How to get Git to work on Visual Studio.

Since Visual Studio doesn't support SSH you will need to create a password for HTTPS usage. Follow the guide above if you haven't created a password. To connect Visual Studio to Gitlab you need to do the following:

1. Click **File->Open->Open from Source Control**
2. Click on **Clone**
3. Enter the HTTPS-address of the Git repo that you want to clone and where you want to clone it to
4. Press **Clone**



5. Now you will need to enter the Gitlab HTTPS username and password
6. You can open and start working on the project by opening the .sln-file

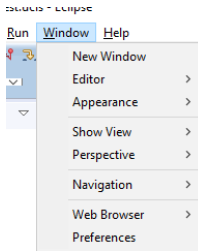
Read more about using git in Visual Studio at this link:

<https://msdn.microsoft.com/magazine/mt767697>

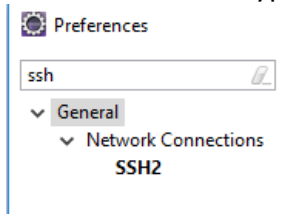
How to get Git to work on Eclipse

If you have problems with `git` not working in Eclipse, there might be some SSH-settings you need to change. See if you have the right SSH public key search path by doing the following:

1. Click *Window->Preferences*



2. At the search-box type `ssh` and click on `ssh2`



3. See if the `SSH2 home` folder is pointing to the same SSH related keys that you have on Gitlab

More info about using Git in Eclipse here:

<http://www.vogella.com/tutorials/EclipseGit/article.html>