Git notice

depository : <https://redmine-df.telecom-bretagne.eu/projects/s5-vadesecure/settings/repositories>

How to configure a git repository :

* create a new folder, this will be your own git local repository
* open a git application (git bash / linux shell ) and place it in the folder
* execute this command : git clone <https://redmine-df.telecom-bretagne.eu/git/s5-vadesecure> *you create the connection between your git repository and the server*
* enter you IMT username and password (if requested)
* The files are now imported locally, you can do all the modifications you want, you are in local !

To import the modification if you already have a local depository

* git pull *It’s not necessary to use git clone each time. Once you connection is established, a simple git pull is enough !*

To export you local depository to the server :

* git add \* *you add all you modifications in the local repository*
* git commit -m “*your message*” *You download locally your change. It’s very important to add a message for that the other member of the group understood the objective of the changes*
* git pull *It’s necessary to import again the repository before to export. Indeed, if you don’t accomplish that, you risk to crush the changes of your fellows ! That will not crush your change, only import the changes of the other. Sometimes you can meet merge problems (see after)*
* git push *you export your changes*

Merge problems :

* Git put together automatically all the local modifications of the fellows. But sometimes, two fellows change the same part of the code and git doesn’t know which version to choose, it’s a merge problem.
* In the code, git will put in a row the two versions. The simplest method is to return to the code and to adjust the conflict, after add, commit and push.

Branch :

* you can create several branches in one git repository. That behaves like 2 different repositories but you can switch easily from the branch in your local repository. That enables us to work in two directions in the same project.
* git branch *find all the branch of the project*
* git branche -d ‘*name*’ *delete the branch*
* git checkout ‘*name*’ *enable us to create a new branch or to switch from branch*
* git merge ‘*name’*  *enable us to merge two branches*
* git rebase ‘*name*’ *enable us to applicate a commit to another branch*
* git fetch ‘*name*’ *enable us to import all the file of the aimed branch which aren’t not in the branch where we work. However, this command doesn’t merge the 2 branches ( it’s git merge )*

Useful command :

* git status *print the modified/new/deleted files*
* git stash *temporary commit, avoid to put a message which will pollute the historic*
* git show *print the information of the git repository*