1. **GIT Configuration for IntelliJ:**

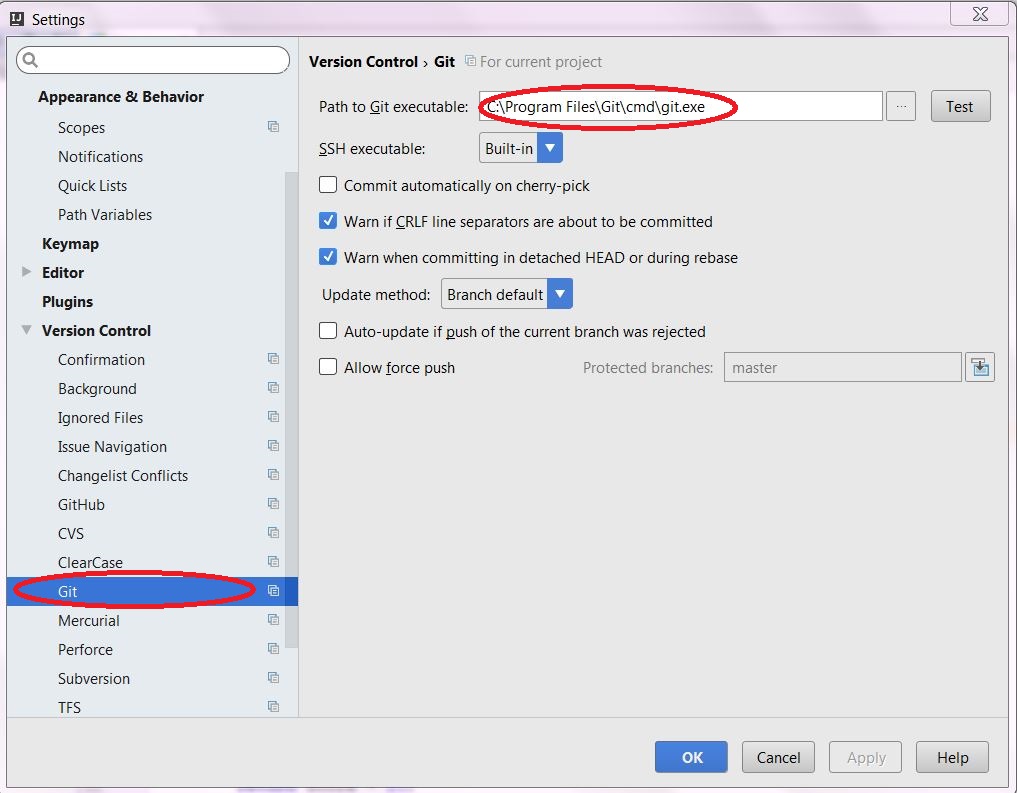
* Install GIT on your computer using below link.

<https://git-scm.com/downloads>

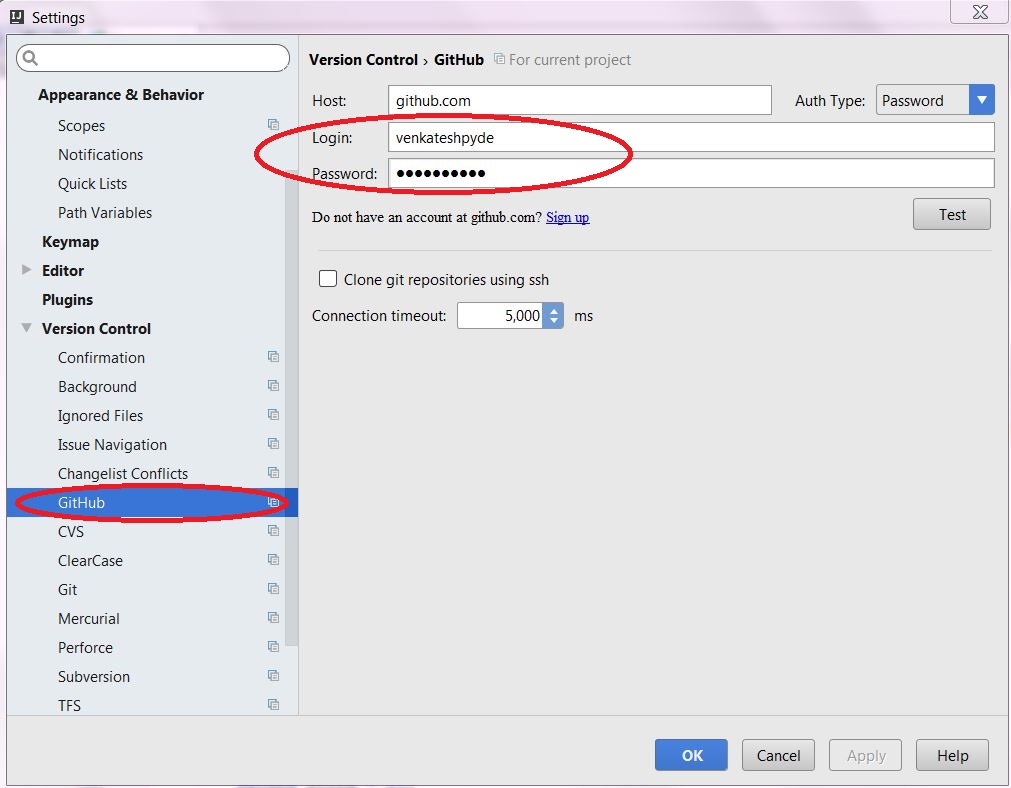
To configure GIT for IntelliJ, follow below steps:

In IntelliJ, Go to ***File ▸ Open Settings and select Version Control ▸ GIT***.

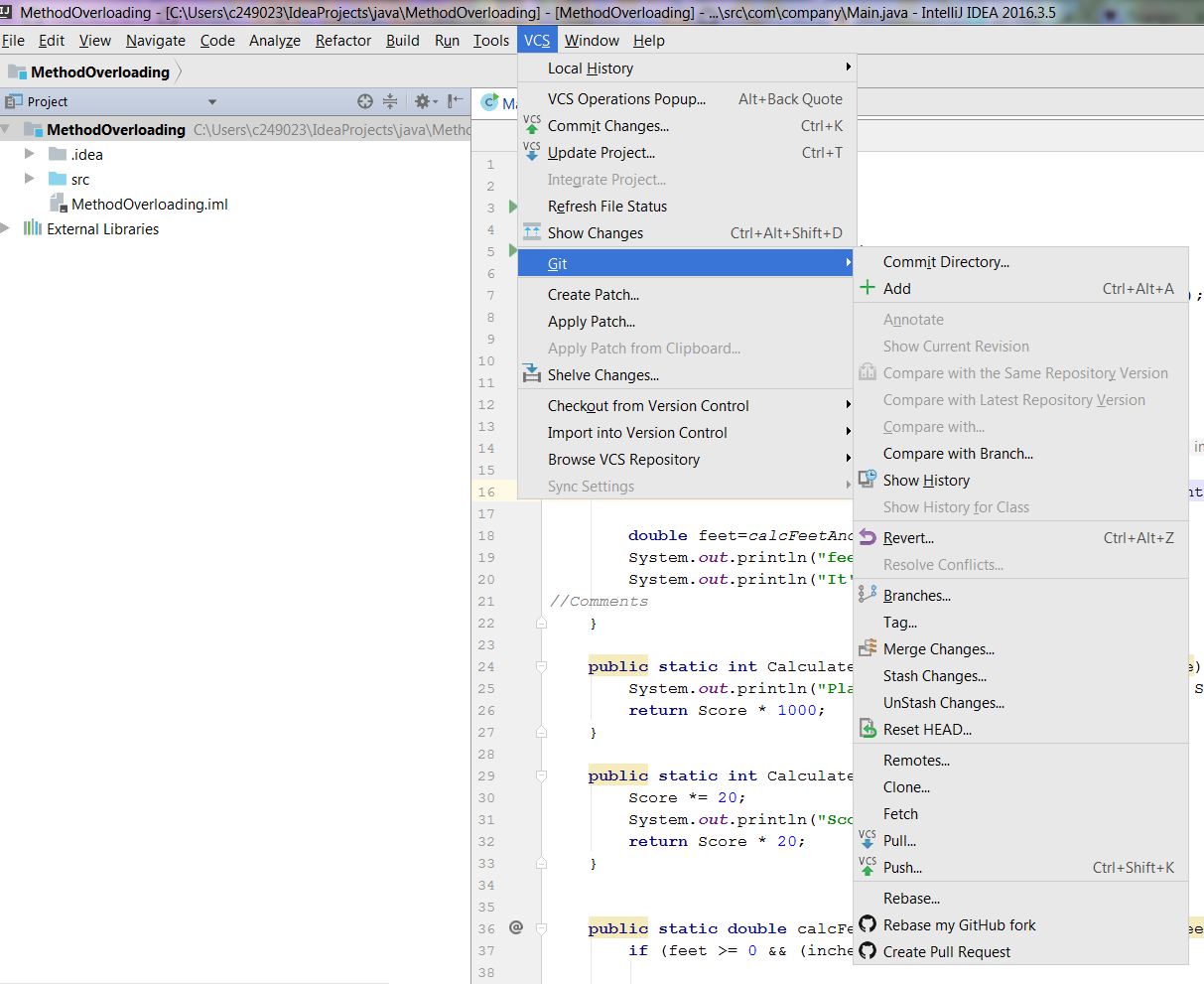
Make sure that GIT executable file (git.exe) path is correct and leave all other options as default.



* Open ***GitHub*** in same tab and enter your ***GitHub username and password*** to be stored and used by IntelliJ and press ***OK.***

******

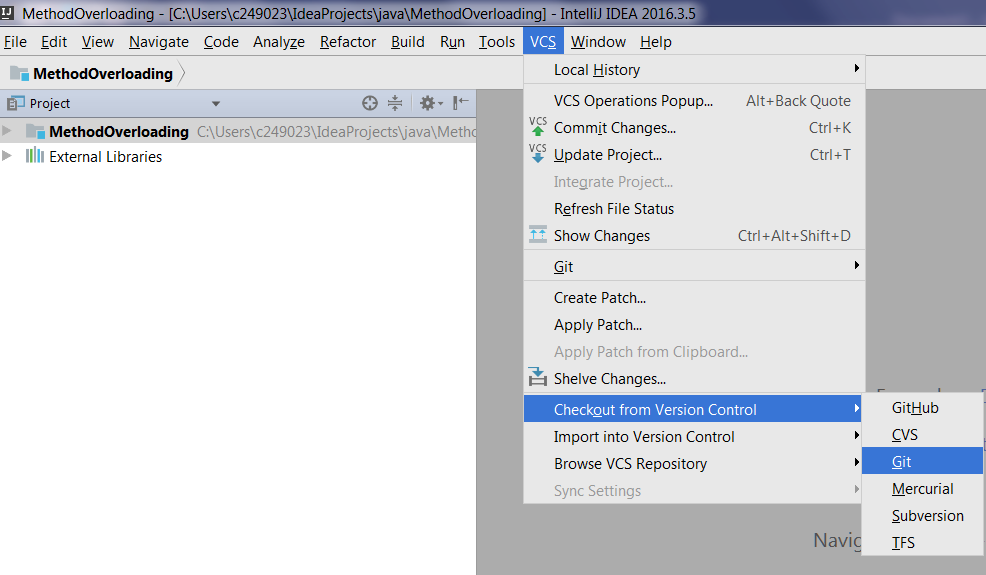
* When Git integration with IntelliJ IDEA is enabled, the **Git** item appears on the **VCS** menu, and on the context menus of the ***[Editor](https://www.jetbrains.com/help/idea/2016.3/intellij-idea-editor.html) and******[Project](https://www.jetbrains.com/help/idea/2016.3/project-tool-window.html)*** views.



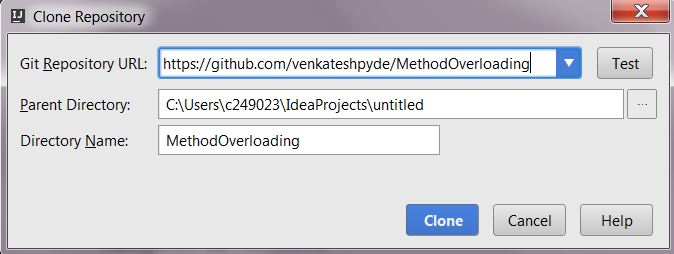
1. **Setting Up a Local Git Repository:**

## **To clone a remote Git repository**

* On the main menu, choose ***VCS ▸ Checkout from Version Control ▸ Git****.*



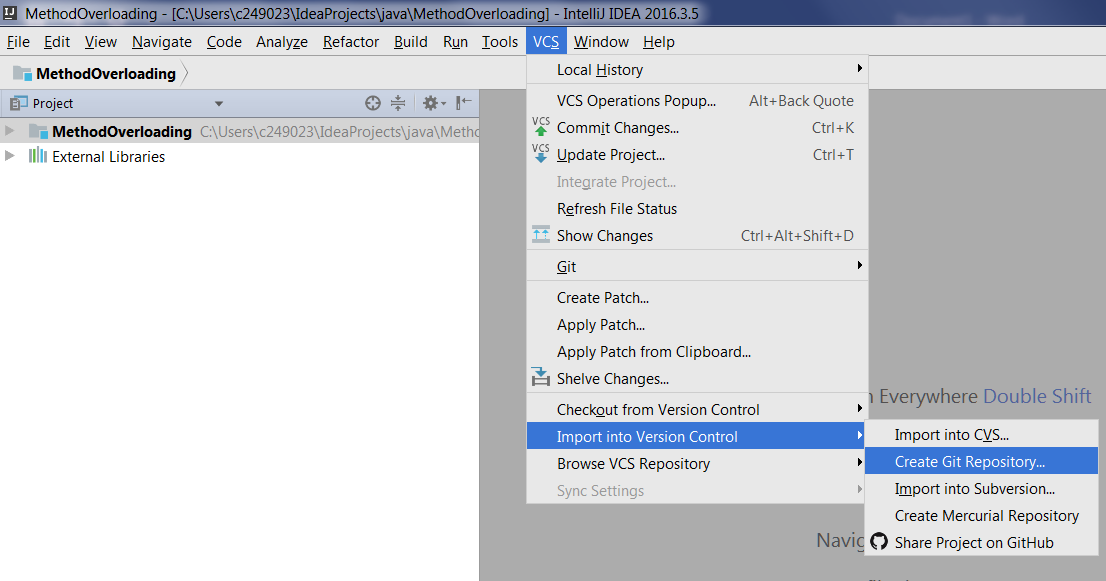
* The ***[Clone Repository](https://www.jetbrains.com/help/idea/2016.3/clone-repository-dialog.html)*** dialog box opens.



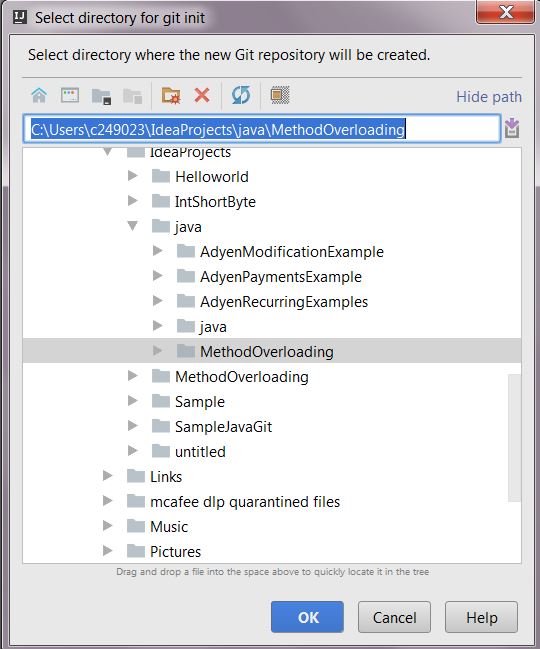
* In the ***Git Repository URL*** text box, type the URL of the remote repository, which you want to clone.
* Click the ***Test*** button next to the ***Git Repository URL*** text box to check that connection to the remote repository can be established successfully.
* In the ***Parent Directory*** text box, specify the directory where you want IntelliJ IDEA to create a folder for your local Git repository. Use /help/img/idea/2016.3/browseButton.png, if necessary.
* In the ***Directory Name*** text box, specify the name of the new folder into which the repository will be cloned. Click ***Clone***.
* Create a new project based on the cloned data by accepting the corresponding suggestion displayed by IntelliJ IDEA.
* Git root mapping will be automatically set to the project root directory.

## **To clone a local Git repository**

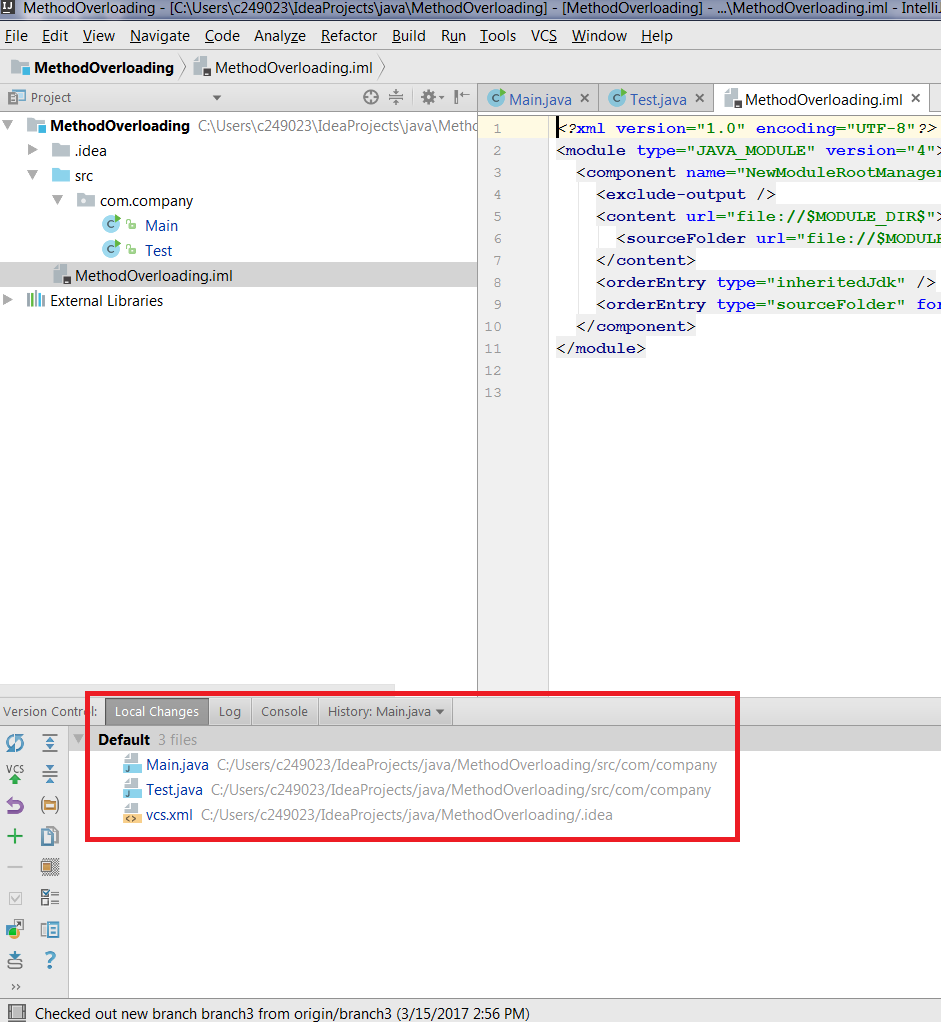
* Open the project you want to store in a repository.
* On the main menu, choose ***VCS ▸ Import into Version Control ▸ Create Git Repository****.*



* In the [dialog that opens](https://www.jetbrains.com/help/idea/2016.3/select-path-dialog.html), specify the directory where you want to create a new Git repository and click ***OK***.



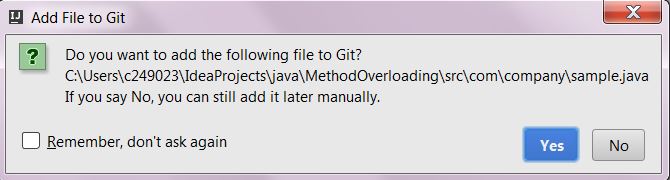
* Put the required [files under Git](https://www.jetbrains.com/help/idea/2016.3/adding-files-to-a-local-git-repository.html) version control. The files appear in the ***Local Changes*** tab of the [Version Control](https://www.jetbrains.com/help/idea/2016.3/version-control-tool-window.html) tool window, under the ***Default*** changelist.



# Adding Files to a Local Git Repository:

* After a Git repository for a project is [initialized](https://www.jetbrains.com/help/idea/2016.3/setting-up-a-local-git-repository.html), you need to add the project data to it.
* If you have specified Git as the version control system for your project in the ***Settings ▸******[Version Control](https://www.jetbrains.com/help/idea/2016.3/version-control.html)*** page, IntelliJ IDEA suggests to put each newly created file under Git version control.

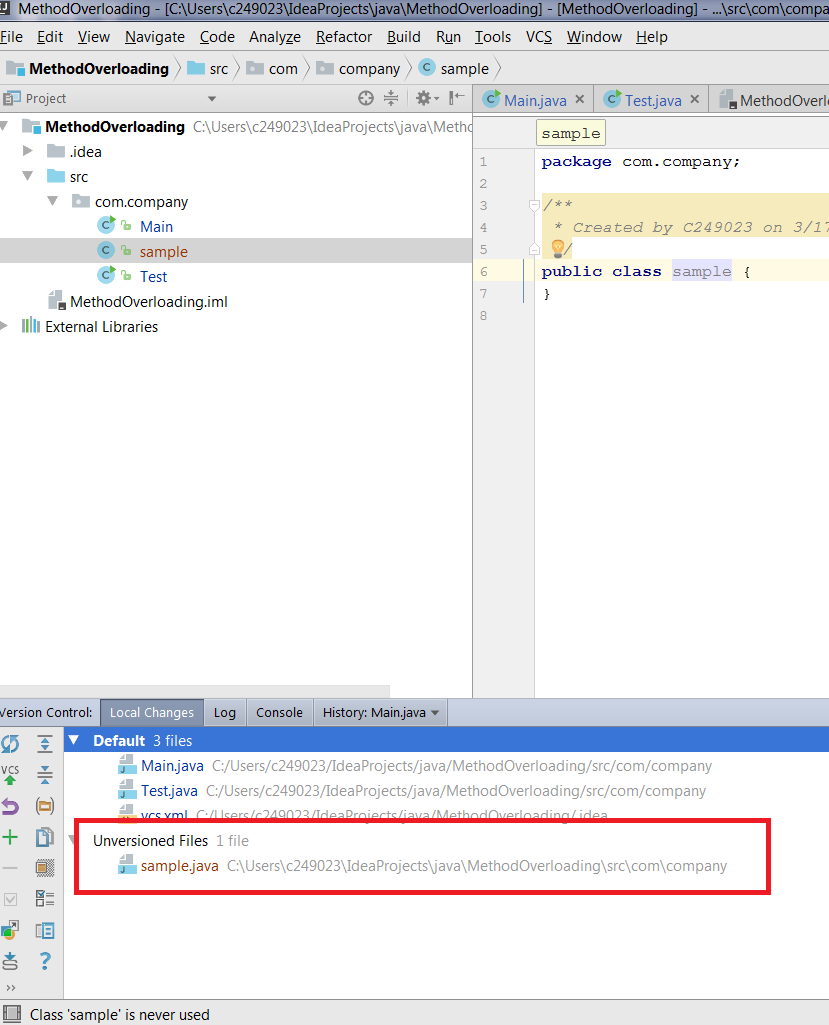
Whenever you create a new file in that project, the following popup will come:



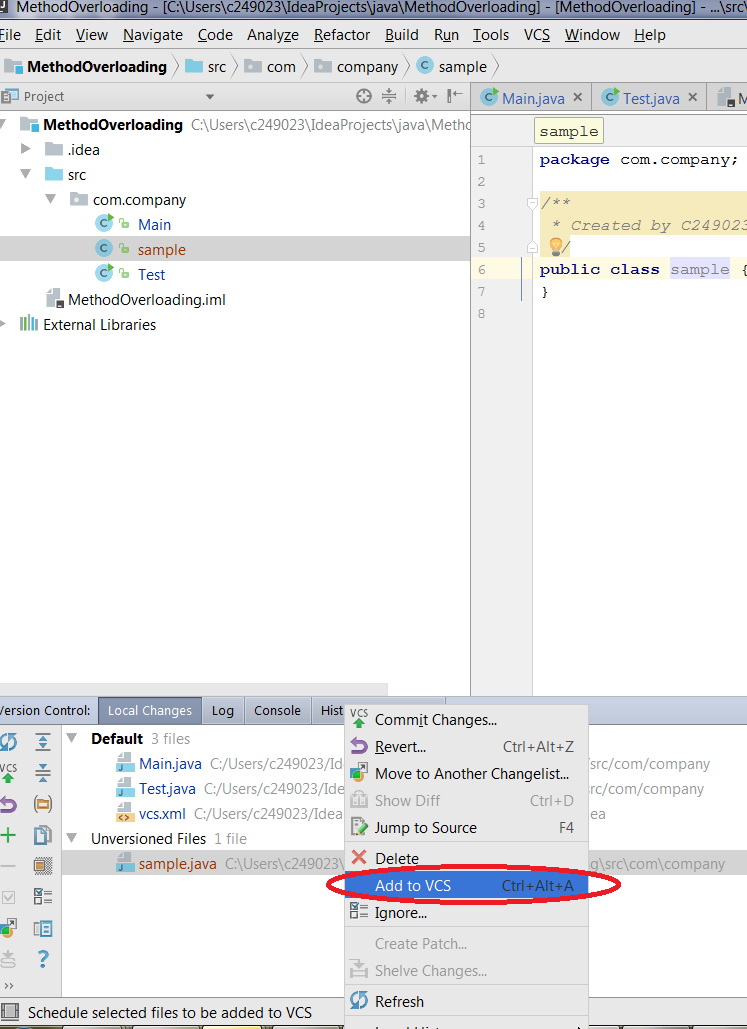
## **3.1** **Adding all currently unversioned files to Git control**

If you want to add any new or unversioned file to Git manually, follow below steps:

* Switch to the [Version Control](https://www.jetbrains.com/help/idea/2016.3/version-control-tool-window.html) tool window and switch to the [Local Changes](https://www.jetbrains.com/help/idea/2016.3/local-changes-tab.html) tab. tool window.



* Expand the ***Unversioned Files*** node and choose ***Add to VCS*** from the context menu.



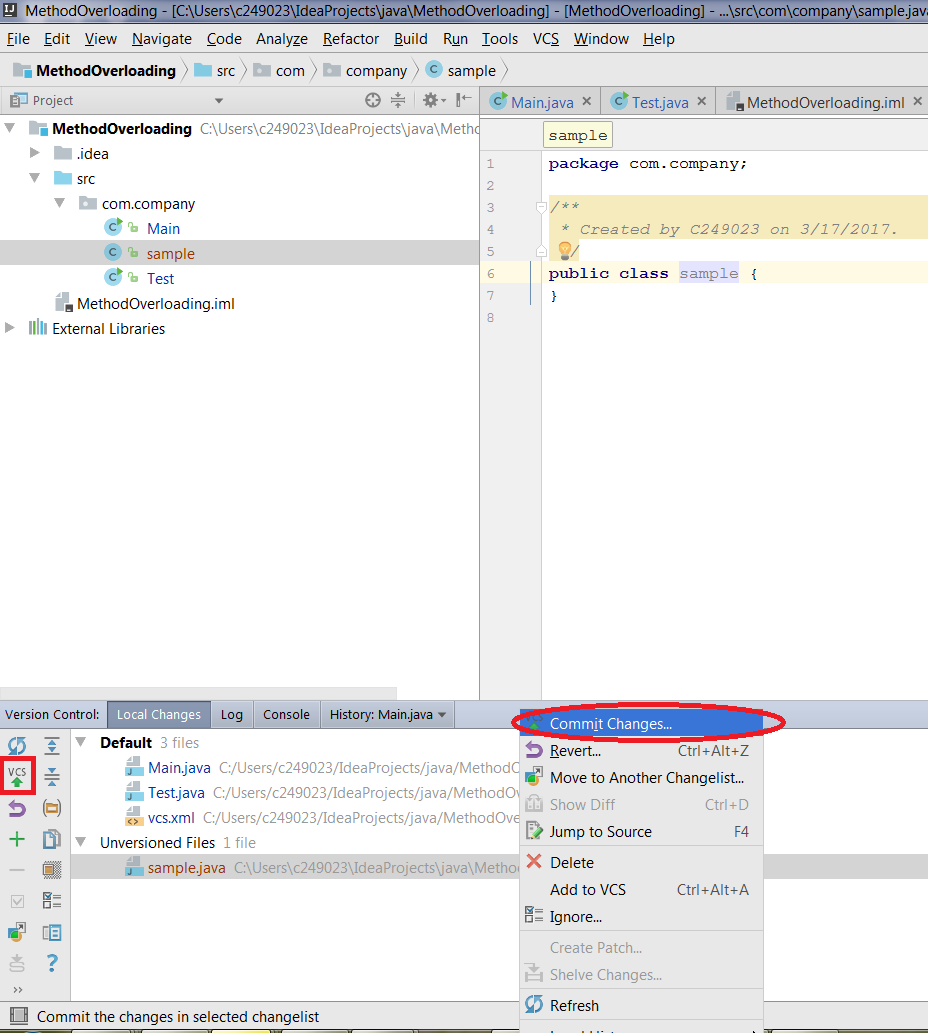
# Committing Changes to a Local Git Repository

## To commit changes to a local repository,

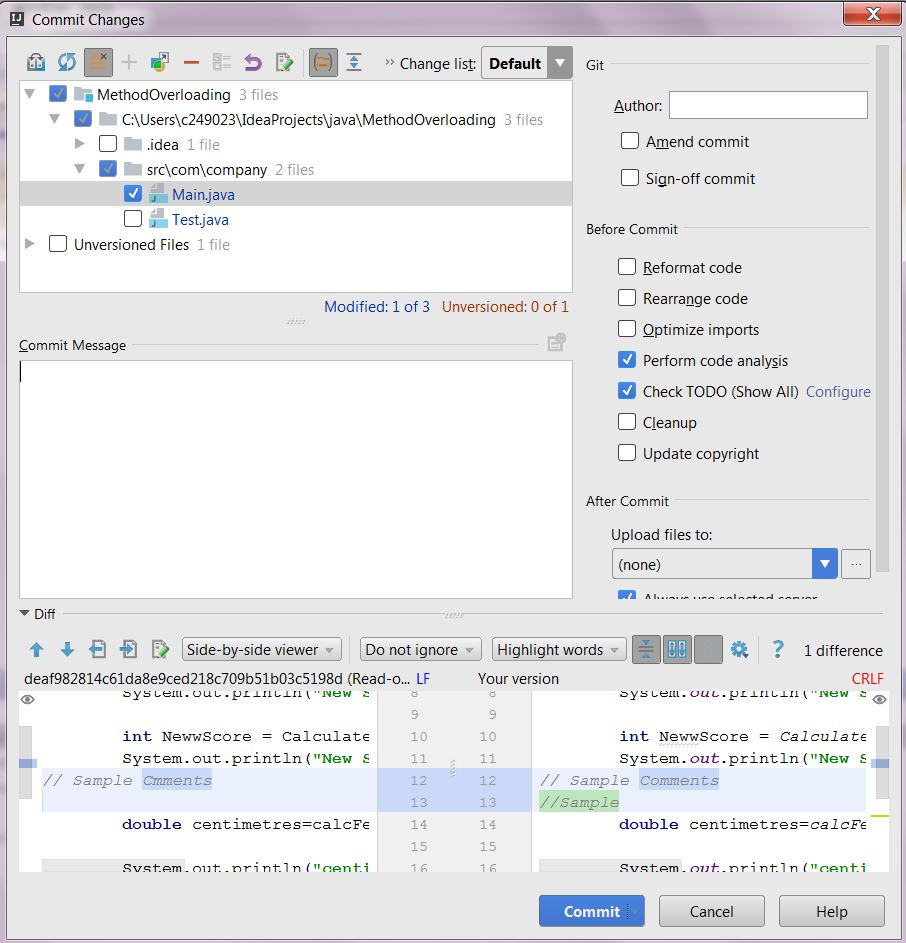
* Open the [Version Control](https://www.jetbrains.com/help/idea/2016.3/version-control-tool-window.html) tool window by choosing

***View ▸ Tool Windows ▸ Version Control.***

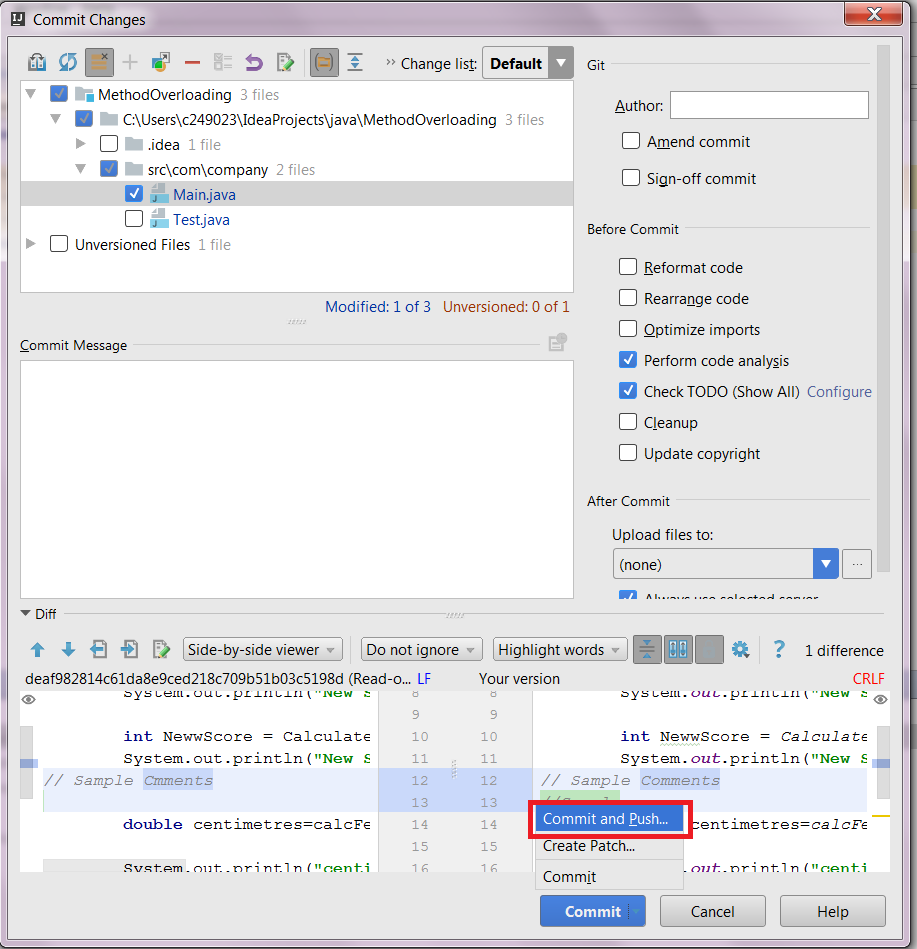
* In the ***[Local Changes](https://www.jetbrains.com/help/idea/2016.3/local-changes-tab.html)*** [tab](https://www.jetbrains.com/help/idea/2016.3/local-changes-tab.html), select the files or folders you want to commit.
* Open the [Commit Changes](https://www.jetbrains.com/help/idea/2016.3/commit-changes-dialog.html) dialog box by doing one of the following:
  1. On the tool window toolbar, click the Commit Changes buttonIcon.
  2. On the main menu, choose ***VCS ▸ Commit Changes***.
  3. On the main menu, choose ***VCS ▸ Git ▸ Commit File***.



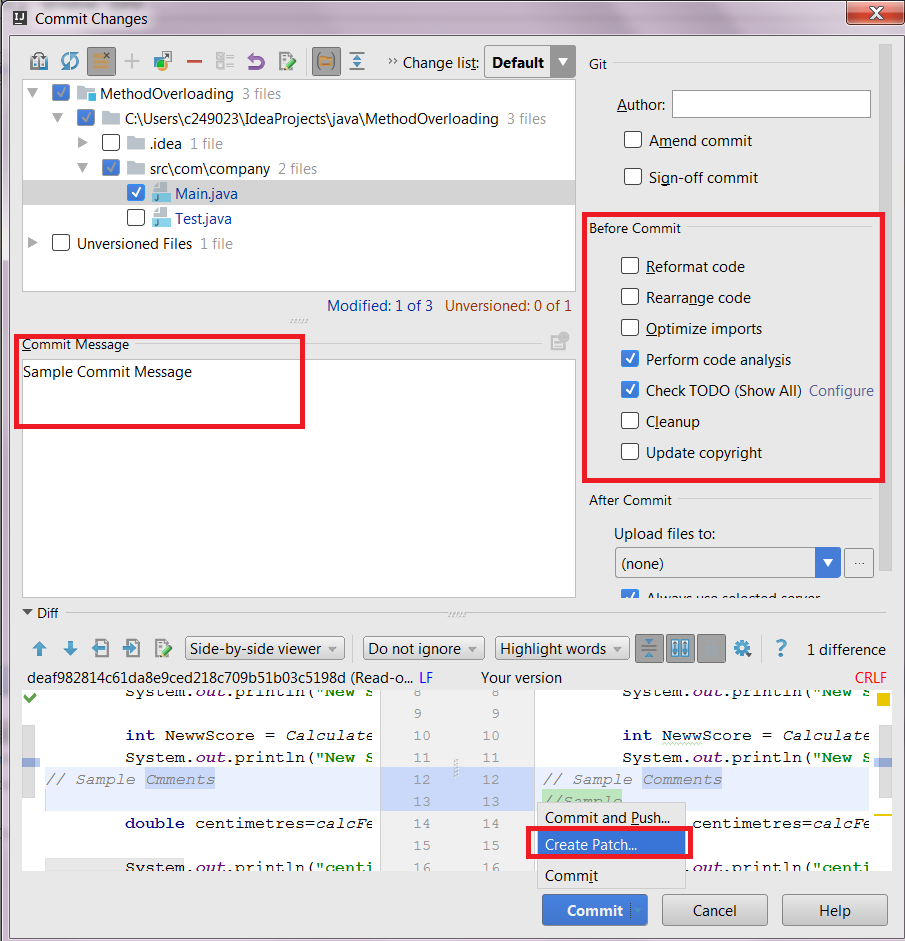
* The Commit Changes dialog lists all files that have changed since the last commit. Select the check boxes next to the files you want to commit now.



* To upload the committed changes to the remote repository immediately, select the ***Push Changes*** check box. The changes will be [pushed](https://www.jetbrains.com/help/idea/2016.3/pushing-changes-to-the-upstream-git-push.html) to the pre-configured origin.



* Add a comment in the ***Commit Message*** section and select the ***Before Commit options*** you want IntelliJ IDEA to perform on the files before committing them to the local repository.



* To save the changes in a text file, click the ***Create Patch***button.
* Click ***Commit***.

# Fetching Changes from a Remote Git Repository

The Fetch operation involves downloading changes from a remote repository without applying them locally.

## To fetch changes from a remote repository,

* On the main menu, choose ***VCS ▸ Git ▸ Fetch***

IntelliJ IDEA retrieves the changes from the repository silently.

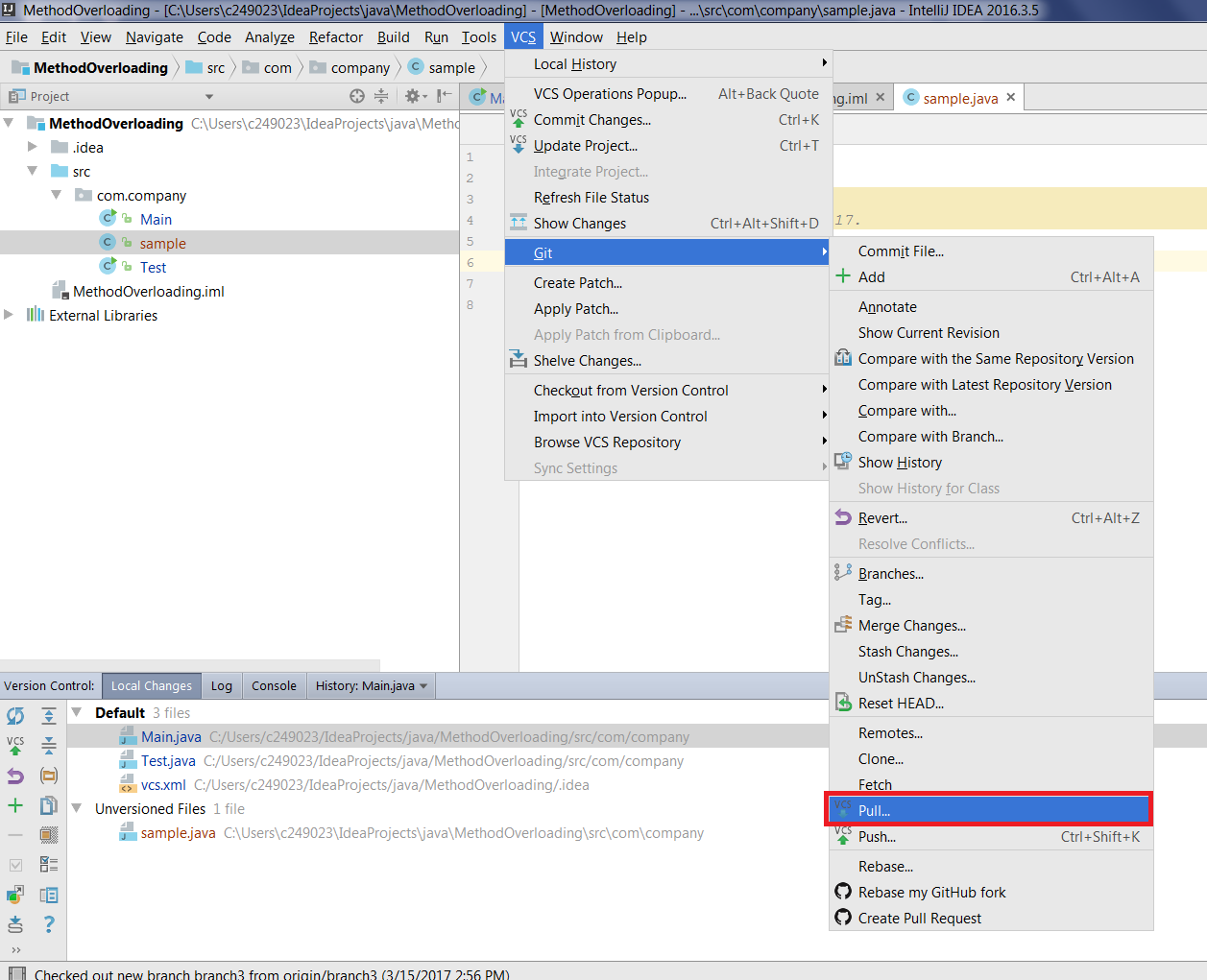
# 

# Pulling Changes from the Upstream (Git Pull)

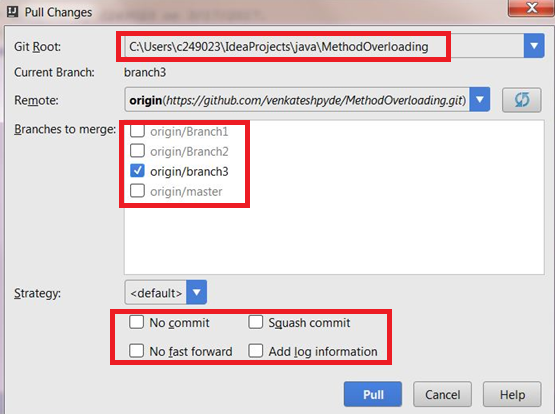
* Refreshing a local Git repository with the changes from the remote repository (**Pull**) involves retrieving changes (**Fetch**) and applying them to the local data (**Merge**).
* The Git integration with IntelliJ IDEA provides interface for specifying the mandatory **Pull** settings and for customizing the **Pull** procedure.

## To pull changes from a remote repository,

1. On the main menu, choose ***VCS ▸ Git ▸ Pull Changes***. The [Pull Changes](https://www.jetbrains.com/help/idea/2016.3/pull-changes-dialog.html) dialog box opens.



1. Specify the required ***Git root*** and the ***URL address*** or alias of the source remote repository.
2. From the ***Branches to Merge*** list, select the remote branches you want to merge to the current local branch.
3. Specify the ***Pull Strategy*** and select ***Pull.***

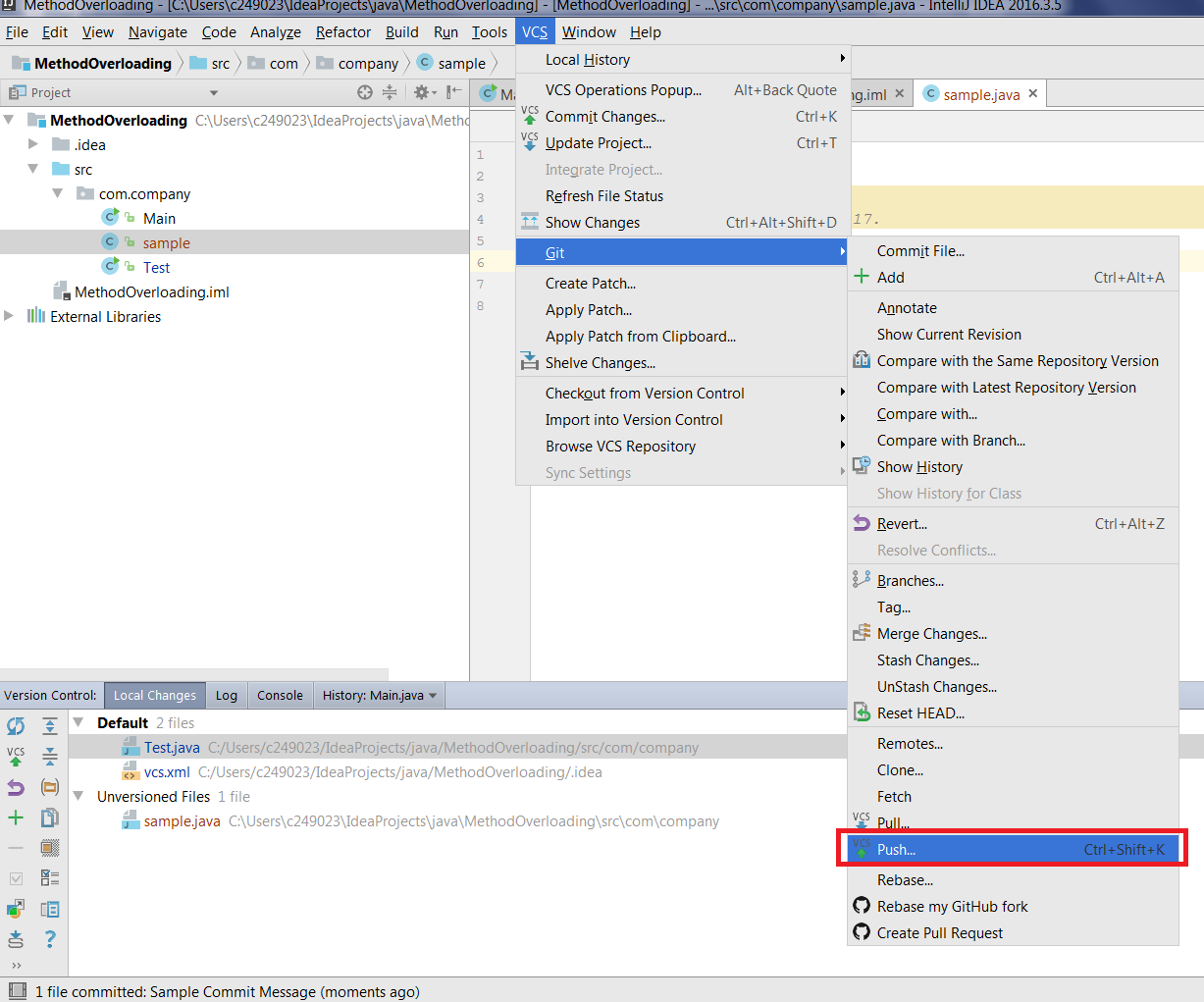


# Pushing Changes to the Upstream (Git Push)

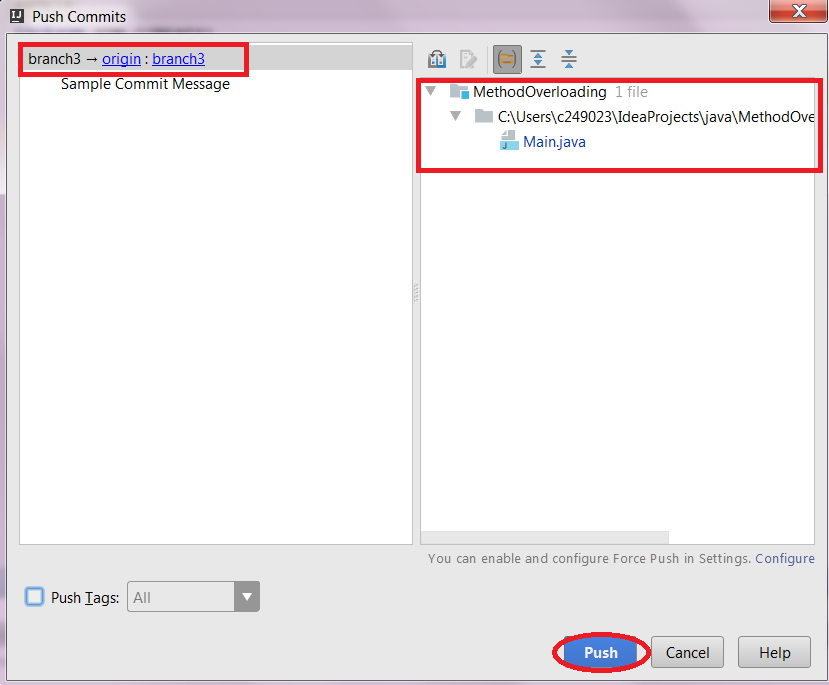
Git integration with IntelliJ IDEA supports uploading changes from the [current branch](https://www.jetbrains.com/help/idea/2016.3/checking-out-switching-between-branches.html" \l "current) to the remote branch.

To push changes, do the following:

* From the main menu, choose **VCS ▸ Git ▸ Push**.



* The [Push Commits dialog](https://www.jetbrains.com/help/idea/2016.3/push-dialog-mercurial-git.html) opens showing all Git repositories (for multi-repository projects) and listing all commits made in the current branch in each repository since the last push.
* If you want to modify the target branch where you want to push, you can do this by clicking on the ***origin:branchname*** and select the appropriate one.
* Click ***Push*** to commit changes to remote repository.

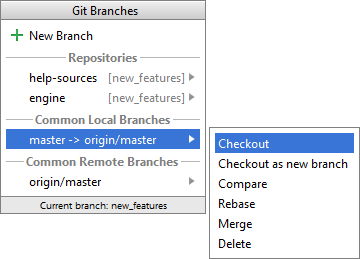


# Managing Branches

# The *Git Branches* popup menu contains all commands that are required to manage Git branches, and can be invoked in a number of ways. It is placed in the bottom-right corner of IntelliJ IDEA as shown below.

# 

* Using options in this popup, GIT branch operations like ***Checking out (Switching between)*** *branches,* ***Merging, Deleting*** *and* ***Comparing*** *branches* etc. can be done.



# Stashing and Unstashing Changes

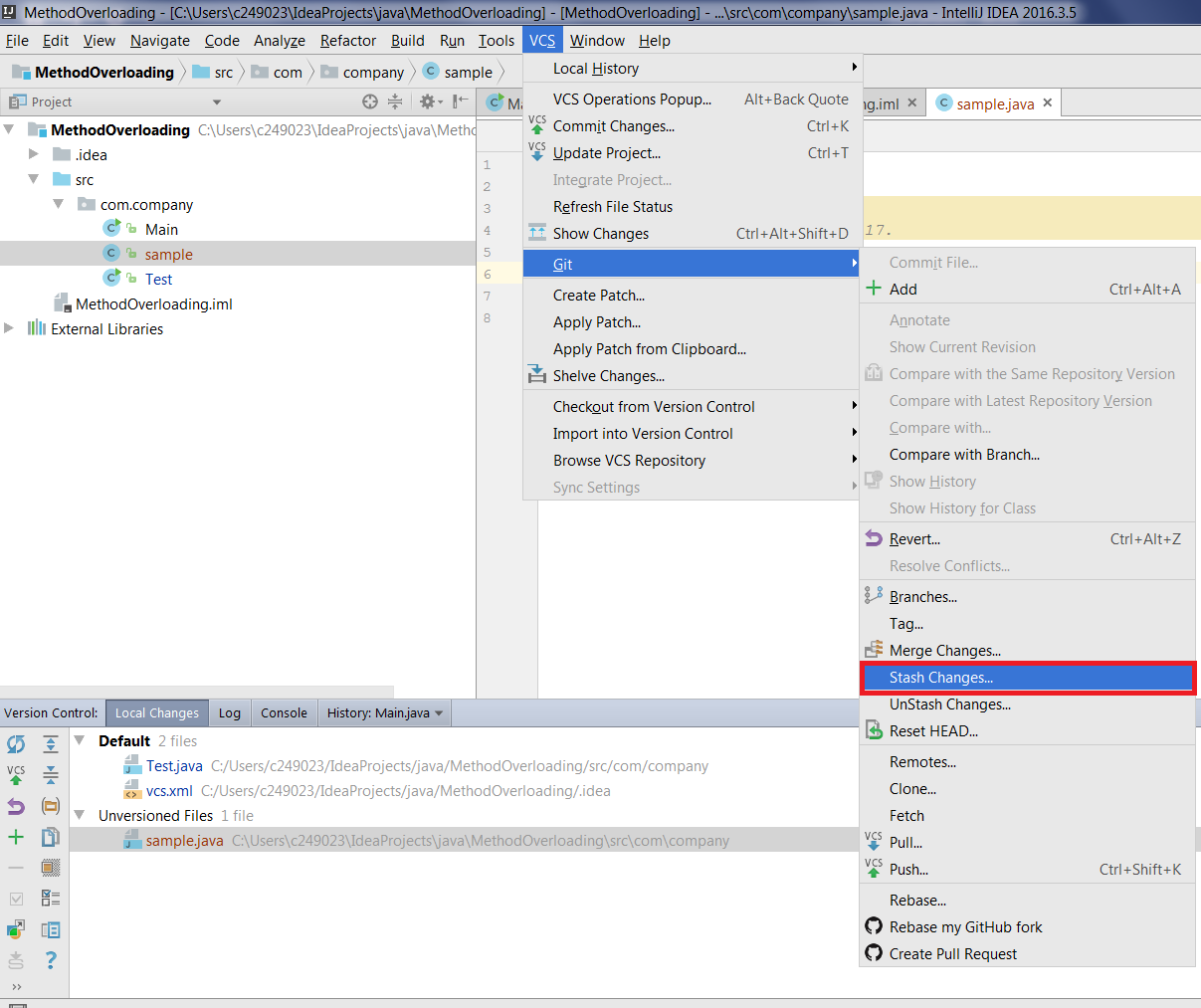
Sometimes it may be necessary to revert your working copy to match the HEAD commit but you do not want to lose the work you have already done. This may happen if you learn that there are upstream changes that are possibly relevant to what you are doing, or if you need to make some urgent fixes.

* [Stashing](https://www.jetbrains.com/help/idea/2016.3/stashing-and-unstashing-changes.html" \l "stash) involves recording the difference between the HEAD commit and the current state of the working directory (stash). Changes to the index can be stashed as well.

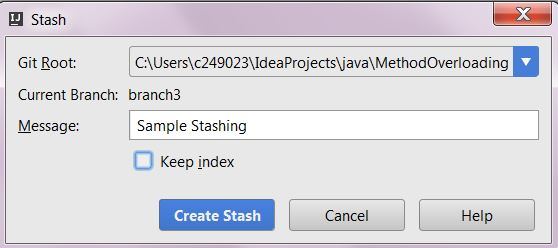
* [Unstashing](https://www.jetbrains.com/help/idea/2016.3/stashing-and-unstashing-changes.html" \l "unstash) involves applying a stored stash to a branch.
* You can apply a stash to an [existing branch](https://www.jetbrains.com/help/idea/2016.3/stashing-and-unstashing-changes.html" \l "1) or create a [new branch](https://www.jetbrains.com/help/idea/2016.3/stashing-and-unstashing-changes.html" \l "2) on its basis.
* A stash can be applied as many times as you need to any branch you need, just [switch](https://www.jetbrains.com/help/idea/2016.3/checking-out-switching-between-branches.html) to the required branch. Keep in mind that:
* Applying a stash after a series of commits results in conflicts that need to be resolved.
* A stash cannot be applied to a "dirty" working copy that is, a working copy to which changes have been made since the latest commit.
* You can store stashes in a list as long as you need and [remove](https://www.jetbrains.com/help/idea/2016.3/stashing-and-unstashing-changes.html" \l "remove) them from the list when necessary.

### **9.1 Saving changes to a new stash:**

* From the main menu, choose **VCS ▸ Git ▸ Stash Changes**. The [Stash](https://www.jetbrains.com/help/idea/2016.3/stash-dialog.html) dialog box opens.



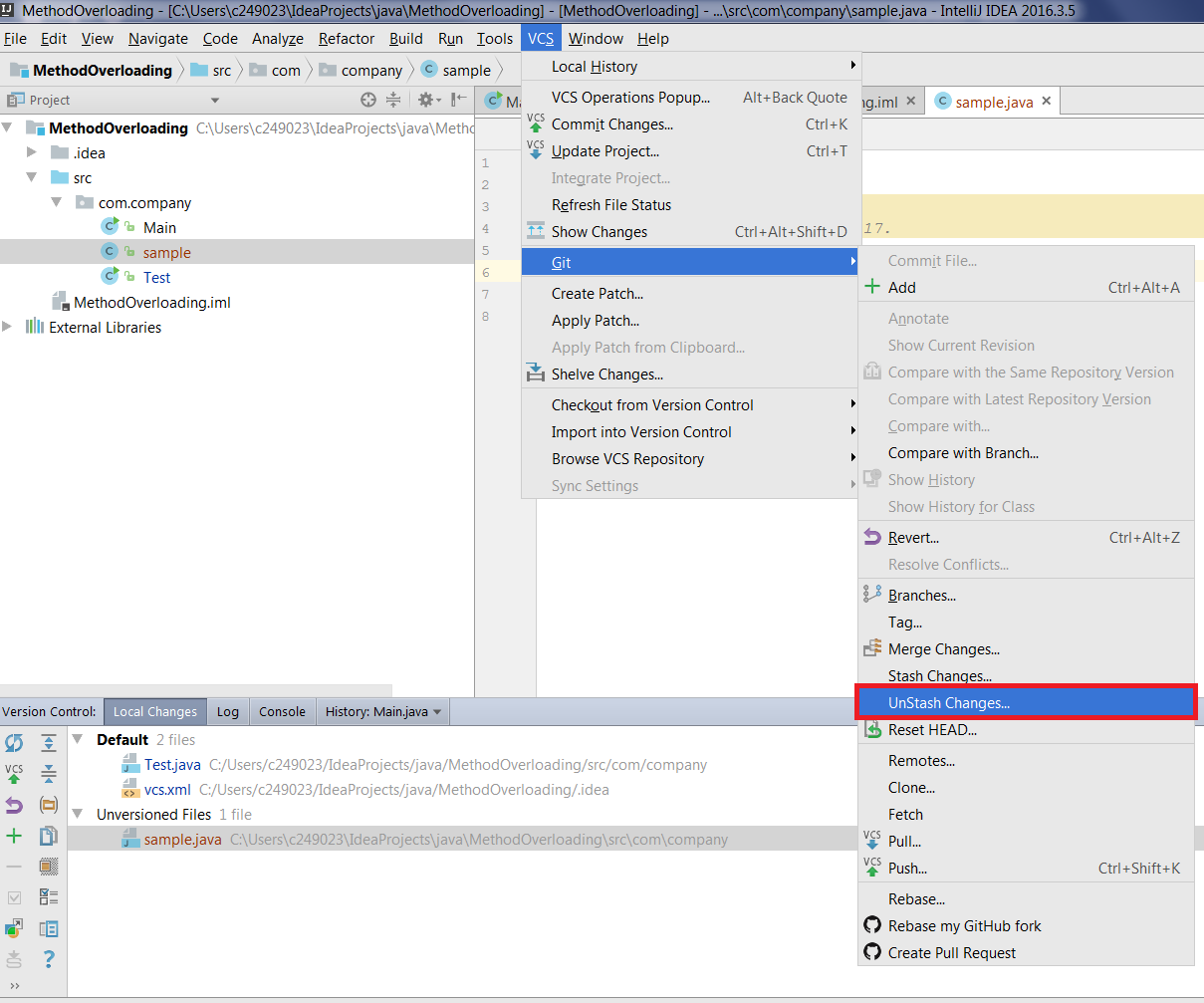
* Select the relevant Git root and make sure that the correct branch is checked out.
* To stash the local changes and bring the changes staged in the index to your working tree for examination and testing, select the **Keep Index** check box.
* In the **Message** text box, describe the changes to be stashed and click ***Create Stash***.



### **9.2 Applying a stash**

## To apply a stash to the same branch,

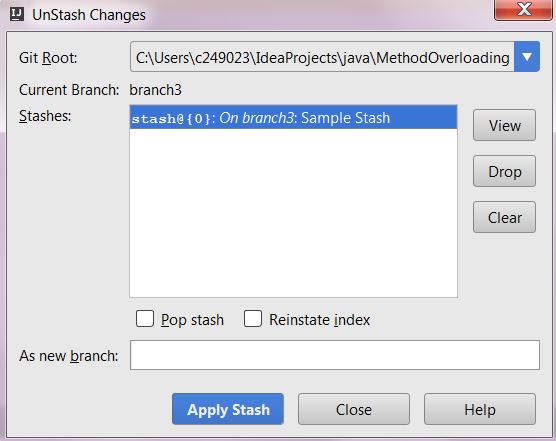
* From the main menu, choose ***VCS ▸ Git ▸ Unstash Changes***. The [Unstash Changes](https://www.jetbrains.com/help/idea/2016.3/unstash-changes-dialog.html) dialog box opens.



* Select the ***Git root*** where you want to apply a stash and make sure that the correct branch is checked out.
* Examine the stash descriptions and the names of the branches where specific stashes were created to find the stash you need.

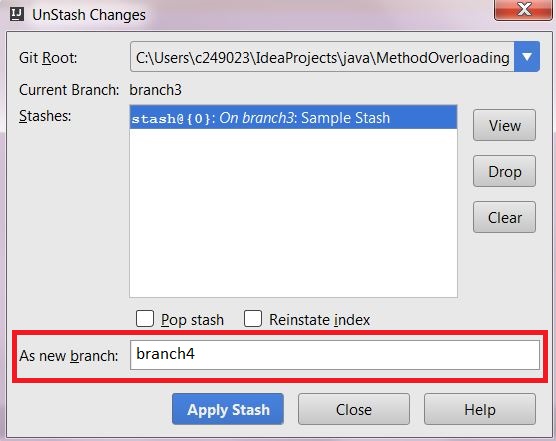
In the ***Stashes*** list, select the relevant stash.

* Click the ***View*** button to open the **Paths affected in commit** dialog box and find out which files are affected in the selected stash.



## To create a new branch on the basis of a stash

* On the main menu, choose **VCS ▸ Git ▸ Unstash Changes**. The [Unstash Changes](https://www.jetbrains.com/help/idea/2016.3/unstash-changes-dialog.html) dialog box opens.
* Select the Git root where you want to apply a stash.
* In the **Stashes** list, select the relevant stash.
* In the **As new branch** text box, type the name of the new branch to be created.



### **9.3 Removing stashes**

To remove all stashes from the list,

* On the main menu, choose ***VCS ▸ Git ▸ Unstash Changes***. The [Unstash Changes](https://www.jetbrains.com/help/idea/2016.3/unstash-changes-dialog.html) dialog box opens.
* In the ***Stashes*** list, select the stashes to be removed and click the ***Drop*** button.

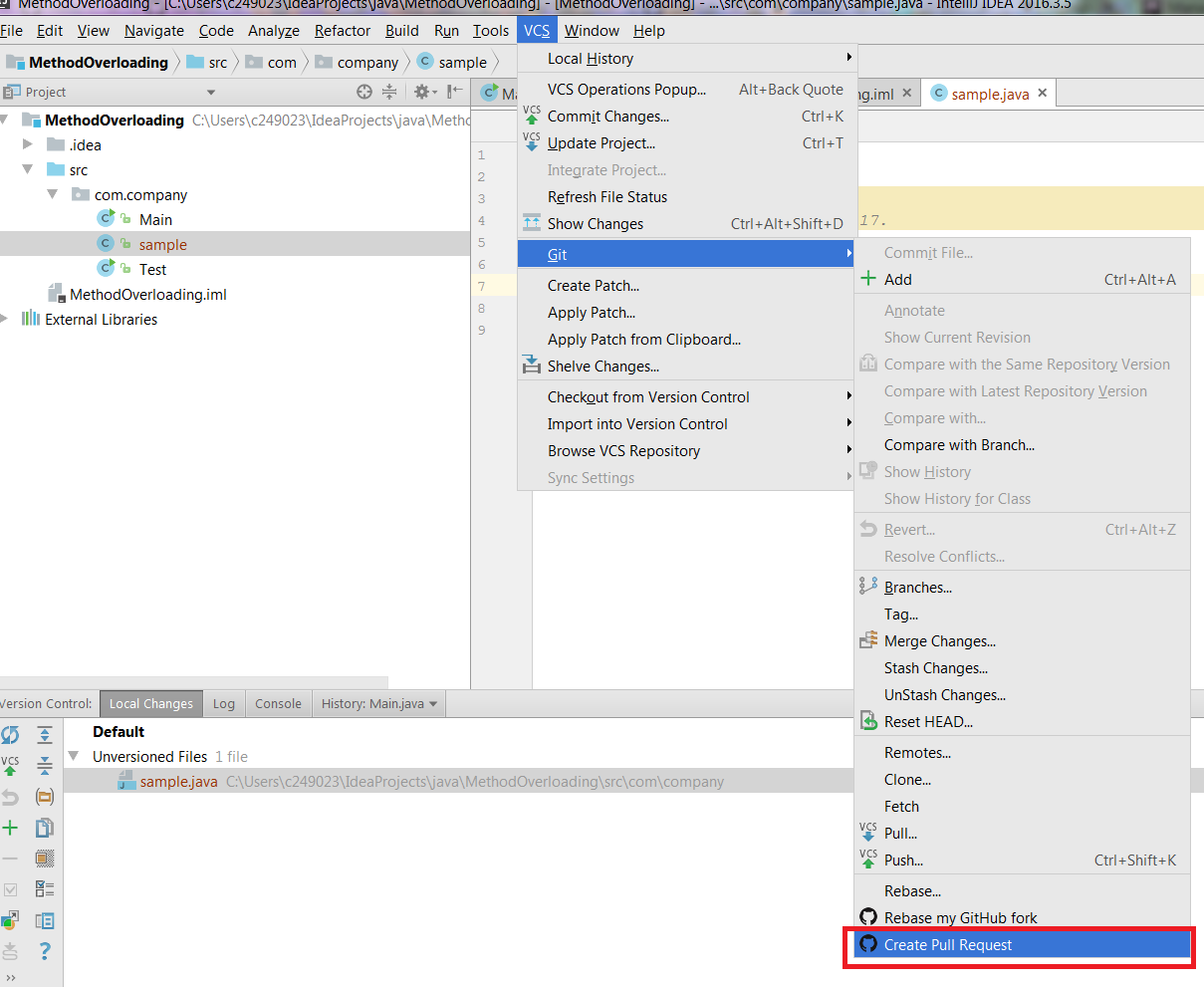
# 

# Creating a Pull Request

* Creating [Pull Requests](https://help.github.com/articles/using-pull-requests) is a nice way to share your code with the community and to follow the collaborative development.
* With Pull Request, you can tell others about the changes you have made and ask for comments, review, or just share the knowledge.
* With all this going on, a Pull Request appears in the original repository only after approval.
* A Pull Request can be prepared right from IntelliJ IDEA without switching to the browser.

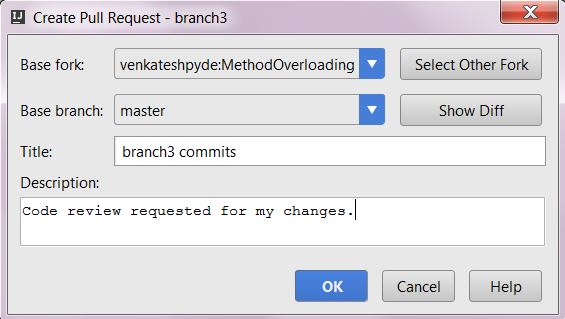
To create a pull request:

1. On the main menu, choose ***VCS ▸ Git ▸ Create Pull Request****.*



The ***Create Pull Request*** dialog box opens.

1. In the ***Base Form*** field, specify the repository to apply the changes to. IntelliJ IDEA attempts to retrieve all the relevant repositories and shows them in the drop-down list. Choose the repository from the drop-down list or click ***Select Other Fork*** and from the ***Form Owner*** drop-down list choose the name of the user who is the owner of the target repository.
2. In the ***Base Branch*** field, specify the branch to apply the changes to. Click ***Show Diff*** to view the list of commits to be included in the pull request. To view the details of a commit, select it and switch to the ***Log tab***, which shows a list of files included in the selected commit list. See also [Merging, Deleting, and Comparing Branches](https://www.jetbrains.com/help/idea/2016.3/merging-deleting-and-comparing-branches.html).
3. In the **Title** field, specify the name for your request.
4. In the **Description** field, optionally provide a brief description of the changes to be applied through the request.



Press ***OK*** to create ***Pull Request.*** The following success message will be displayed.



The Pull request can be tracked using that blue hyperlink.

