GITHUB User Guide



Table of Contents

[1. Introduction 4](#_Toc526503969)

[2. GITHUB Account setup 4](#_Toc526503970)

[3. Creating Repositories 5](#_Toc526503971)

[3.1 Creating a new repository 5](#_Toc526503972)

[3.2 Adding an existing local project to GitHub 5](#_Toc526503973)

[3.3 Cloning a repository 7](#_Toc526503974)

[4. Branch creation in GITHUB 8](#_Toc526503975)

[5. Pull Requests 9](#_Toc526503976)

[5.1 Creating a pull request 9](#_Toc526503977)

[5.2 Addressing and resolving merge conflicts 10](#_Toc526503978)

[5.3 Resolving a merge conflict using command line 11](#_Toc526503979)

[5.3.1 Competing line change merge conflicts 11](#_Toc526503980)

[5.4 Removed file merge conflicts 12](#_Toc526503981)

**Revision History**

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 10/01/2018 | 1.0 | Initial Documentation | PX Team |
|  |  |  |  |

# Introduction

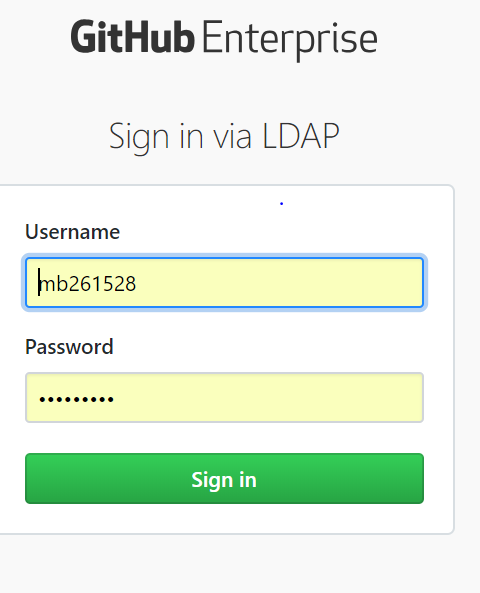
This document describes how to use GitHub as a tool for version control and the procedure performed by DevOps Team to push or pull code from the remote repositories. It explains how to create a repository, clone a repository, and create a branch. This document also explores on how to use git commands and the GitHub desktop application. It also explains the folder structure and hierarchy that BCBSA uses for all its repositories

# GITHUB Account setup

Create an adhoc request in service now to have the user windows ID added to GG\_Git-Users group in LDAP. This ensure the user to have access to GITHUB. Refer to RITM0263026 as a sample request while requesting access.

Once access is granted user your windows LAN ID to login to <https://git.bcbsa.com>

Now use the credentials you normally use to login to your machine to sign in to BCBSA git enterprise account

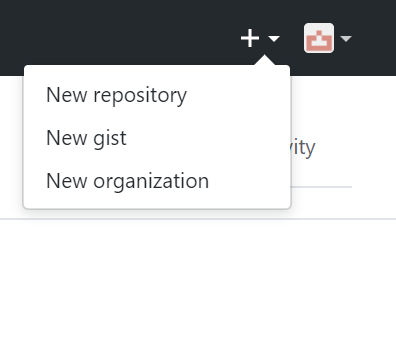


# Creating Repositories

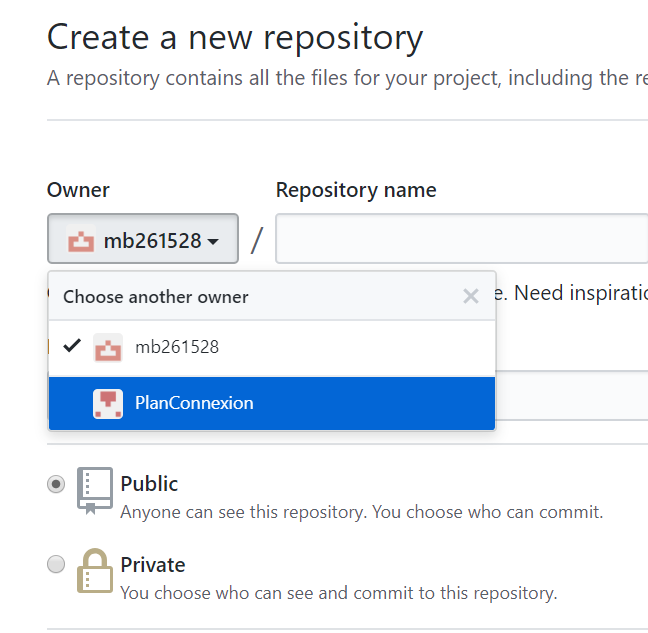
## Creating a new repository

You can create a new repository after signing into your enterprise account where you have sufficient permissions

1. In the upper-right corner of any page, click +, and then click **New repository.**



1. In the Owner drop-down, select the account you wish to create the repository on.



1. You can choose to make the repository either public or private. Public repositories are visible to the public, while private repositories are only accessible to the team who part of the repository owners.
2. Additional optional items can pre-populate in the repository. If you are importing an existing repository to GitHub, do not choose any of these options, as you may introduce a merge conflict. You can choose to add these files using the command line later.
3. When finished, click **Create repository**.

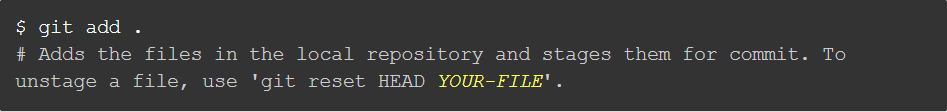
## Adding an existing local project to GitHub

Putting your existing work on GitHub can let you share and collaborate in lots of great ways.

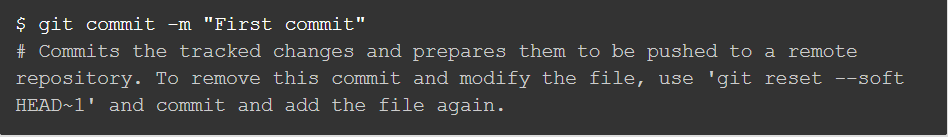
1. Create a new repository on GitHub. To avoid errors, do not initialize the new repository with README, license, or gitignore files. You can add these files after your project is pushed to GitHub.
2. Open Git Bash (or Command Prompt).
3. Change the current working directory to your local project.
4. Initialize the local directory as a Git repository.



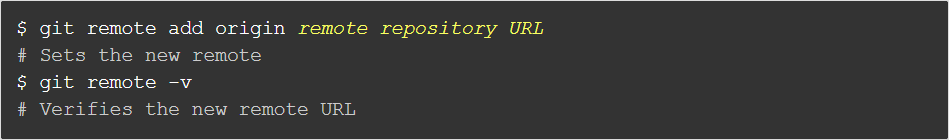
1. Add the files in your new local repository. This stages them for the first commit.



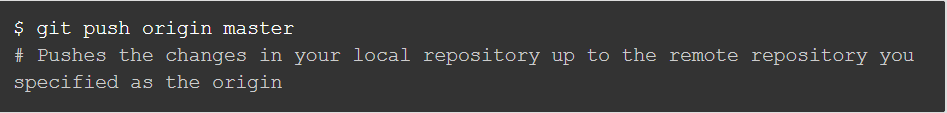
1. Commit the files that you've staged in your local repository.



1. At the top of your GitHub repository's Quick Setup page, click  to copy the remote repository URL.
2. In the Command prompt, add the URL for the remote repository (the URL is the link that you copied in the above step) where your local repository will be pushed.



Push the changes in your local repository to GitHub.

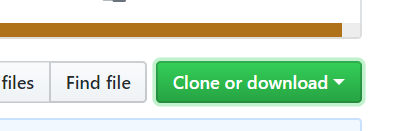


## Cloning a repository

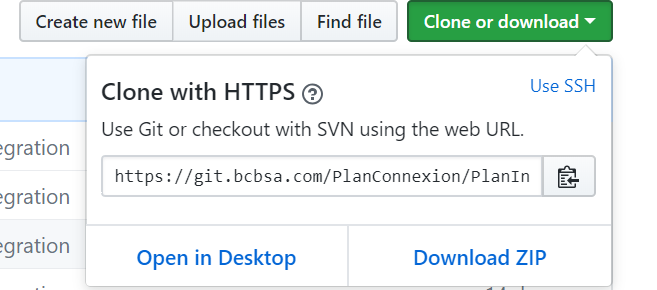
When you create a repository on GitHub, it exists as a remote repository. You can clone your repository to create a local copy on your computer and sync between the two locations.

This procedure assumes you have already created a repository on GitHub, or have an existing repository owned by someone else you'd like to contribute to.

1. On GitHub, navigate to the main page of the repository.
2. Under the repository name, click **Clone or download**.



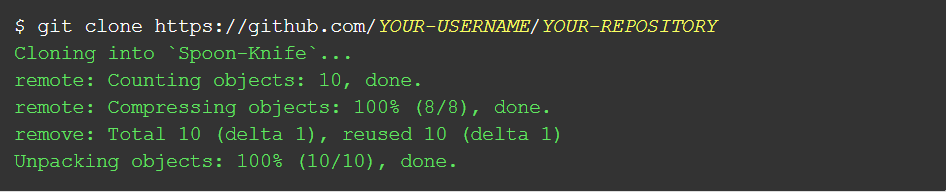
1. In the Clone with HTTPs section, click  to copy the clone URL for the repository.



1. Open Git Bash (or Command Prompt).
2. Change the current working directory to the location where you want the cloned directory to be made.
3. Type git clone, and then paste the URL you copied in Step 2.



1. Press **Enter**. Your local clone will be created.



# Branch creation in GITHUB

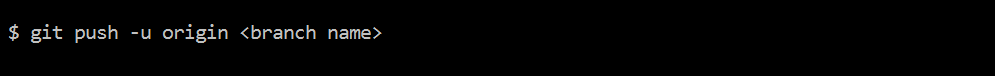
Branching in version control is the duplication of an object under version control (such as a source code file or a directory tree). Modifications can happen in parallel along both branches and the originating branch is sometimes called the parent branch (master in GitHub), the upstream branch (or simply upstream, especially if the branches are maintained by different organizations or individuals).

You can create or delete branches directly on GitHub.

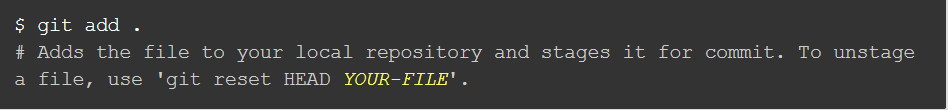
1. On GitHub, navigate to the main page of the repository.
2. Click the branch selector menu.
3. Type a unique name for your new branch.
4. Press **Enter**.
5. You can switch between different branches or between a branch and master by using the command git checkout. To switch to the parent branch we use git checkout master.



1. To publish a branch to GitHub that has been locally created use the command git push –u origin



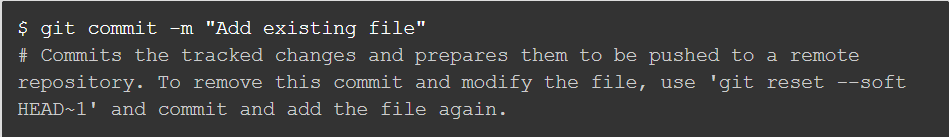
1. Once a branch is created and some work is done in that branch and if you want those changes to be reflected in the remote GitHub repository we use commands like add (for adding the new file that has been created), commit (for committing the changes made to an existing file and the new files that have been created if any) and push (for pushing the changes made locally to the remote GitHub repository)
2. For adding a new file to the repository you have to move the file you'd like to upload to GitHub into the local directory that was created when you cloned the repository.
3. Open Git Bash (or Command Prompt).
4. Change the current working directory to your local repository.
5. Stage the file for commit to your local repository.



1. To view the status of the work done type in git status.



1. Commit the file that you've staged in your local repository. If you just made changes to existing files then you can skip step 10 and directly commit your changes.



1. Push the changes in your local repository to GitHub.

# Pull Requests

Pull requests let you tell others about changes you've pushed to a branch in a repository on GitHub. Once a pull request is opened, you can discuss and review the potential changes with collaborators and add follow-up commits before your changes are merged into the base branch.

Once you've created a pull request, you can push commits from your topic branch to add them to your existing pull request. These commits will appear in chronological order within your pull request and the changes will be visible in the "Files changed" tab.

## Creating a pull request

Create a pull request to propose and collaborate on changes to a repository. These changes are proposed in a *branch*, which ensures that the master branch only contains finished and approved work.

Pull requests can only be opened if there are differences between your branch and the upstream branch. You can specify which branch you'd like to merge your changes into when you create your pull request.

1. On GitHub, navigate to the main page of the repository.
2. In the "Branch" menu, choose the branch that contains your commits.
3. To the right of the Branch menu, click **New pull request**.
4. Use the *base* branch dropdown menu to select the branch you'd like to merge your changes into, then use the *compare* branch drop-down menu to choose the topic branch you made your changes in.
5. Type a title and description for your pull request.
6. Click **Create pull request**.