Git: Step-by-Step Guide to Push and Pull Files

Prerequisites

- Git must be installed on your machine. https://git-scm.com/downloads

- You must have a remote Git repository (GitHub/GitLab/Bitbucket) created.

PART 1: Send (Push) Files to Remote Repository

Step 1: Initialize Local Git Repository

git init

Step 2: Configure Git

git config --global user.name "Your Name"

git config --global user.email "your_email@example.com"

Step 3: Add Files to Staging Area

git add .

Step 4: Commit the Changes

git commit -m "Initial commit"

Step 5: Add Remote Repository

git remote add origin https://github.com/username/repo-name.git

git remote -v

Git: Step-by-Step Guide to Push and Pull Files

Step 6: Push Files to Remote Repository

git push -u origin main

(or git push -u origin master)

Then: git push

PART 2: Pull Files from Remote Repository

Option 1: Clone Remote Repository (First Time)

git clone https://github.com/username/repo-name.git

Option 2: Pull Changes into Existing Local Repo

git pull origin main

(or git pull origin master)

Common Git Workflow (Summary)

1. Make changes to files.

2. git add .

3. git commit -m "your message"

4. git push

Other Useful Git Commands

git status

Git: Step-by-Step Guide to Push and Pull Files

git log

git checkout branch-name

git checkout -b new-branch-name

git merge branch-name

Troubleshooting

Authentication Errors: Use a Personal Access Token (PAT)

Push Rejected: Run git pull --rebase origin main, then git push

Working with Staging Area in Git

Adding One File to the Staging Area

To add a single file:

git add filename.txt

Removing a File from Staging Area (Unstage)

To remove a file from staging and return it to the working directory:

git reset filename.txt

Adding Multiple Files to Staging Area

To add two specific files:

git add file1.txt file2.txt

To add all files:

Git: Step-by-Step Guide to Push and Pull Files

git add .

Recovering from a Commit

1. Move last commit back to staging area (soft reset):

git reset --soft HEAD~1

2. Move last commit back to working directory (mixed reset):

git reset --mixed HEAD~1

3. Discard last commit completely (hard reset):

git reset --hard HEAD~1

Use hard reset carefully. It deletes changes permanently.

View Differences Between Working Directory, Staging, and Commits

To view what's changed but not staged:

git diff

To view what's staged but not committed:

git diff --cached

To view all staged and unstaged changes:

git status

Git: Step-by-Step Guide to Push and Pull Files

Importing a Cloned Git Project into Eclipse

After cloning a project using:

git clone https://github.com/username/repo-name.git

Follow these steps in Eclipse:

1. Open Eclipse IDE.

2. Go to: File > Import.

3. Choose: Git > Projects from Git.

4. Select: 'Existing local repository' and click Next.

5. Browse and select the cloned repository folder.

6. Choose 'Import existing Eclipse projects' or 'Import as general project' depending on the repo.

7. Click Finish to complete import.

Working with Git Branches

To view all branches:

git branch -a

To create a new branch:

git branch new-branch-name

To switch to an existing branch:

git checkout branch-name

To create and switch to a branch in one command:

Git: Step-by-Step Guide to Push and Pull Files

git checkout -b new-branch-name

To pull latest updates for a specific branch:

git pull origin branch-name

To set upstream and track remote branch:

git push --set-upstream origin branch-name

Switching Branches and Pushing Changes (Git Perspective)

1. View all local and remote branches:

   git branch -a

2. Switch to an existing branch:

   git checkout branch-name

3. Make changes in your files as needed.

4. Stage your changes:

   git add .

5. Commit your changes:

   git commit -m "Your meaningful commit message"

6. Push your changes to the same branch in remote repository:

Git: Step-by-Step Guide to Push and Pull Files

```
git push origin branch-name
```

If it's the first time pushing this branch, you may need:

```
git push --set-upstream origin branch-name
```

Switching to a Branch from Eclipse (Git Perspective)

1. Open Eclipse.
2. Go to the 'Git' perspective: Window > Perspective > Open Perspective > Other > Git.
3. In the Git Repositories view, right-click your repository.
4. Select 'Switch To' > 'New Branch' or choose from existing branches.
5. Select the desired branch and click 'Checkout'.
6. Eclipse will switch to that branch and update the workspace accordingly.

Committing and Pushing Changes to GitHub from Eclipse

1. After editing files, go to the Git Staging view in Eclipse.
2. Drag files from 'Unstaged Changes' to 'Staged Changes'.
3. Enter a commit message.
4. Click the 'Commit and Push' button (or 'Commit' to commit only).
5. Eclipse will push the changes to the currently checked-out branch on the remote repository.

Note: If it's a new branch, Eclipse may prompt you to set upstream tracking.

Full Git Workflow to Add Files to an Existing GitHub Repo (Same Branch)

Step 1: Clone the remote GitHub repository

git clone https://github.com/your-username/your-repo.git

cd your-repo

Step 2: Pull the latest changes from the remote master branch

git pull origin master

Step 3: Copy your local files into this cloned folder

cp -r /path/to/your/local/project/* .

Replace with your actual file path

Step 4: Stage the new/updated files

git add .

Step 5: Commit your changes

git commit -m "Added my project files"

Step 6: Push to the remote master branch

git push origin master