

Step-by-Step Guide to Using Git with GitHub

Introduction

This guide explains how to use Git with GitHub for your programming projects. We will cover the steps to initialize a repository, add files to the staging area, commit them, and finally push them to GitHub. We will also explain how to set up access tokens and SSH keys to secure your connections with GitHub.

1. Initialize a Git Repository

Create a new directory for your project and initialize a Git repository:

```
mkdir my_project
cd my_project
git init
```

Explanation:

`mkdir my_project`: Creates a new directory named `my_project`.
`cd my_project`: Changes the current directory to enter the `my_project` directory.
`git init`: Initializes a new Git repository in the current directory.

2. Create Files and Add Them to the Staging Area

Create files and add them to the staging area:

```
echo "# My Project" > README.md
git add README.md
```

Explanation:

`echo "# My Project" > README.md`: Creates a `README.md` file with the content `# My Project`.
`git add README.md`: Adds the `README.md` file to the staging area.

3. Commit the Changes

Commit the added files with a descriptive message:

```
git commit -m "Initial commit"
```

Explanation:

`git commit`: Records the changes in the repository's history with a descriptive message. The text after `-m` is the commit message, which should describe the changes made.

4. Add a Remote Repository on GitHub

Add a remote repository for your GitHub project:

```
git remote add origin https://github.com/your_username/my_project.git
```

Explanation:

`git remote add origin`: Links the local repository to a remote repository named `origin`. Replace `https://github.com/your_username/my_project.git` with the URL of your GitHub repository.

5. Push the Changes to GitHub with a Token

Push the local changes to the remote repository on GitHub using a personal access token:

```
git push -u origin main
```

Explanation:

`git push`: Sends the local commits to the remote repository on GitHub. `-u origin main` sets `origin` as the default remote for the `main` branch. You will be prompted to enter your GitHub username and your personal access token instead of your password.

6. Generate a Personal Access Token on GitHub

To create a personal access token on GitHub:

1. Go to <https://github.com/settings/tokens>
2. Click on Generate new token
3. Give your token a name and select the necessary permissions (usually repo)

4. Click on Generate token and copy the generated token

7. Push the Changes to GitHub with an SSH Key

Generate a new SSH key:

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

Explanation:

ssh-keygen: Generates a new SSH key.

-t rsa -b 4096: Specifies the RSA key type with a length of 4096 bits.

-C "your_email@example.com": Adds a label with your email.

Add your SSH key to the SSH agent:

```
eval "$(ssh-agent -s)"  
ssh-add ~/.ssh/id_rsa
```

Explanation:

eval "\$(ssh-agent -s)": Starts the SSH agent in the background.

ssh-add ~/.ssh/id_rsa: Adds your private key to the SSH agent.

Add your SSH key to GitHub:

```
cat ~/.ssh/id_rsa.pub
```

Explanation:

Use this command to display the contents of your public key, then copy it and add it to your GitHub account under "Settings" > "SSH and GPG keys" > "New SSH key".

Complete Example

Here is a complete example following the steps above:

```
mkdir my_project  
cd my_project  
git init  
echo "# My Project" > README.md  
git add README.md  
git commit -m "Initial commit"  
git remote add origin https://github.com/your_username/my_project.git  
git push -u origin main  
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"  
eval "$(ssh-agent -s)"  
ssh-add ~/.ssh/id_rsa  
cat ~/.ssh/id_rsa.pub
```