# **Working with Git Using Visual Studio Code (VS Code) UI**

This document provides detailed steps on how to use Git in **Visual Studio Code (VS Code)** through the **UI** to manage your code changes. You will learn how to initialize a repository, stage files, commit changes, and push them to a remote repository like GitHub, all within the VS Code interface.

## **1. Prerequisites**

Before you start working with Git in VS Code, ensure that you have the following:

* **Git** installed on your system.
  + To verify if Git is installed, open a terminal and run:

->git --version

* **VS Code** installed on your system.

## **2. Open or Create a Project in VS Code**

1. Open **Visual Studio Code**.
2. To work with an existing project, go to **File** > **Open Folder** and select the folder of your project.
3. If you're starting a new project, create a new folder on your local machine, and then open that folder in VS Code.

## **3. Initialize a Git Repository**

### **Option 1: Initialize a Git Repository from VS Code UI**

1. Open the **Source Control** panel by clicking on the **Source Control icon** (the icon that looks like a branch) on the left sidebar.
2. If no repository is initialized in your folder, you will see a message like **"No source control providers are registered."**
3. Click the **Initialize Repository** button to create a new Git repository.
   1. This action will create a .git folder in your project directory, marking it as a Git repository.

### **Option 2: Initialize Git via Terminal in VS Code**

1. Open the **Integrated Terminal** in VS Code by pressing Ctrl + ~ (or Cmd + ~ on macOS).
2. Type the following command to initialize Git:

->git init

This will create the .git folder and initialize Git in your project directory.

## **4. Set Up Git (If Not Already Done)**

If you haven't set up Git with your username and email, do so via the terminal:

1. Open the **Integrated Terminal** in VS Code.
2. Run the following commands to set your global Git username and email:

->git config --global user.name "Your Name"  
 ->git config --global user.email "[youremail@example.com](mailto:youremail@example.com)"

To check your settings, run: git config --list

## **5. Stage and Commit Changes Using VS Code UI**

### **Stage Files**

1. Go to the **Source Control** panel (click the Source Control icon on the left sidebar).
2. You will see a list of files in the **Changes** section, indicating that these files are modified.
3. To stage a file:
   1. Hover over the file and click the **+** icon next to it.
   2. To stage all files, click the **+** icon next to **Changes**.

### **Commit Changes**

1. In the **Source Control** panel, you will see a text box where you can enter a **commit message**.
   1. Write a brief, descriptive message (e.g., "Updated README" or "Added new feature").
2. After entering the commit message, click the **✔️ (checkmark)** button at the top of the Source Control panel to commit the changes.

## **6. Push Changes to the Remote Repository**

### **Set Up a Remote Repository (e.g., GitHub)**

Before pushing your changes, you must link your local repository to a remote one (e.g., on **GitHub**):

1. Create a new repository on **GitHub** (or any other Git service).
   1. Go to [GitHub](https://github.com/) and create a new repository.
2. Copy the **repository URL** (HTTPS or SSH) from GitHub.

### **Link Remote Repository to Local Git Repository**

1. In the **Integrated Terminal** of VS Code, type the following command to add the remote repository:

git remote add origin <https://github.com/your-username/your-repository.git>

1. To verify the remote has been added, run:

git remote -v

### **Push Your Changes**

1. After committing your changes, click the **... (ellipsis)** icon in the top-right corner of the Source Control panel.
2. From the dropdown menu, select **Push** to push your changes to the remote repository.
   1. Alternatively, you can use the terminal:

git push -u origin main

If you are using a branch other than main (e.g., master), replace main with your branch name.

### **Authentication:**

* If prompted, enter your **GitHub** (or Git service) username and password or use a **Personal Access Token** if required by your Git host.

## **7. Additional Git Operations in VS Code UI**

### **Pull Changes from Remote Repository**

1. To fetch changes from your remote repository, click the **... (ellipsis)** icon in the top-right corner of the Source Control panel.
2. Select **Pull** to fetch and merge changes from the remote repository into your local branch.

Alternatively, use the terminal:

git pull origin main

### **Create a New Branch**

1. Click the branch name in the **bottom-left corner** of VS Code (it typically says master or main).
2. Click on **Create New Branch** and enter a name for your new branch.
3. After creating a new branch, you can start making changes in that branch.

### **Switch Between Branches**

1. To switch branches, click the branch name in the bottom-left corner.
2. Select the branch you want to switch to.

### **View Commit History**

To view the commit history of your repository, you can install the **Git Graph** extension from the VS Code marketplace:

1. Go to the **Extensions** panel (click the Extensions icon on the left sidebar).
2. Search for **Git Graph** and click **Install**.
3. Once installed, open **Git Graph** from the **View** menu to see a graphical history of your commits.

## **Conclusion**

With these steps, you can effectively manage your Git repository using the Visual Studio Code UI. VS Code makes it easy to stage files, commit changes, push to a remote repository, and more—all without leaving the editor.

By integrating Git into your workflow within VS Code, you can streamline version control, collaborate with others, and keep your codebase organized.