Git create new repo

Here's a step-by-step guide to send a new project to GitHub using Visual Studio Code (VS Code):

---

1. Prepare Your GitHub Repository

1. Log in to your GitHub account.
2. Click the **+** icon in the top-right corner and select **New Repository**.
3. Fill in the repository details:
    - Repository Name (e.g., my-new-project)
    - Choose **Public** or **Private**.
    - Do not initialize with a README (this avoids conflicts).
4. Click **Create Repository**.
    - GitHub will display instructions on how to push an existing repository or create a new one.

---

2. Open Your Project in VS Code

1. Launch **VS Code**.
2. Open the folder containing your project:
    - Go to **File** > **Open Folder** (or **Open...** on Mac).

---

3. Initialize Git in Your Project

1. Open the **Terminal** in VS Code:
    - Go to **View** > **Terminal**.
2. Run the following commands:

bash

Copy code

git init

This initializes a new Git repository in your project folder.

---

4. Add Files to Git

1. Stage all your project files:

bash

Copy code

git add .

This stages all files for commit.

2. Commit the changes with a message:

bash

Copy code

git commit -m "Initial commit"

---

5. Connect to the GitHub Repository

1. Copy the repository URL from GitHub:
    - Go to your repository page on GitHub.
    - Click the **Code** button and copy the HTTPS URL (e.g., https://github.com/username/my-new-project.git).
2. Add the remote repository in VS Code:

bash

Copy code

git remote add origin <repository-URL>

Replace <repository-URL> with the URL you copied.

---

6. Push Your Code to GitHub

1. Push the code to the main branch:

bash

Copy code

git branch -M main

git push -u origin main

- git branch -M main ensures your local branch is named main (to match GitHub's default branch).
- git push -u origin main pushes your code and sets up a tracking branch.

---

7. Verify on GitHub

1. Go back to your GitHub repository in your browser.
2. Refresh the page to see your files uploaded.

---

8. Use VS Code for Future Updates

1. Make changes to your project files.
2. In the terminal:

bash

Copy code

git add .

git commit -m "Your message here"

git push

This updates your GitHub repository with the latest changes.

Git pull

To add new files to a previously uploaded project on GitHub from a new PC, follow these steps:

---

1. Set Up Git on Your New PC

1. **Install Git** (if not already installed):
    - Download Git and install it.
2. **Configure Git**:
   Open a terminal (Command Prompt, PowerShell, or Git Bash) and set your username and email:

bash

Copy code

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

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

---

2. Clone Your GitHub Repository

1. Open a terminal.
2. Navigate to the folder where you want to clone the repository:

bash

Copy code

cd path/to/your/directory

3. Clone the repository:

bash

Copy code

git clone https://github.com/username/repository.git

Replace username and repository with your GitHub username and repository name.

---

3. Add New Files

1. Navigate into the cloned repository:

bash

Copy code

cd repository

2. Copy or create the new files you want to add into this folder.

---

4. Stage the New Files

1. Check the current status to see the untracked files:

bash

Copy code

git status

2. Stage all new and modified files:

bash

Copy code

git add .

If you only want to stage specific files, list them explicitly:

bash

Copy code

git add filename1 filename2

---

5. Commit Your Changes

1. Commit the changes with a message:

bash

Copy code

git commit -m "Added new files"

---

6. Push the Changes to GitHub

1. Push your changes to the repository:

bash

Copy code

git push origin main

Replace main with your branch name if it’s different (e.g., master).

---

7. Verify Changes

1. Go to your GitHub repository in a web browser and confirm the new files are uploaded.

If you encounter issues with authentication (e.g., username and password):

- Use a **Personal Access Token** instead of a password. Follow this guide to create a token.