Pavneet Singh

Setting Up a React Project from GitHub

Pavneet Singh

Oct 20, 2020 • 8 Min read • 1,641 Views

Oct 20, 2020 • 8 Min read • 1,641 Views

Web Development Front End Web Dev... Client-side Framew... React

Introduction

7

- Introduction
- Basic Terms and Command of Git
- Prerequisite:
- Clone a Repository Using SSH Link

Introduction

GitHub is the most widely used hosting service provider for projects and files to manage data changes effectively. Apart from repository hosting, GitHub also offers many other services like `gists`, CI/CD integration, package publication, GitHub APIs, GitHub Pages, sponsors, and much more. The `create-react-app` tool automatically adds a `.gitignore` file that contains the names or patterns to ignore files/directories while pushing the code to the

- Clone a Repository Using HTTP Link
- Clone a Repository Using GitHub CLI
- Alternative Options to Set Up a Repository
- Run a Cloned React Project
- Tips
- Conclusion
- Top ^

GitHub server. `git` and `GitHub` are widely used to develop software in a collaborative environment. This guide explains the details of setting up a React project from a GitHub repository using different methods.

Basic Terms and Command of Git

There is some important terminology related to `git` files and commands that are required to understand how `git` works:

- `git` is a tool to manage the history of a project using `git` commands. The history details are stored in a hidden directory named `.git`.
- `repository` is a conventional name of a `git` project hosted on the GitHub server.
- `.gitignore` files contain the names (or patterns) of the files or directories that will neither be tracked nor uploaded to a GitHub repository by `git`.
- `remote` is the command used to add SSH or HTTPS URL links of a GitHub repository.
- `origin` is just a conventional name for a GitHub repository URL.
- `staged` can be visualized as a bucket of files or directories whose changes are ready to be stored. The `add` command is used to stage changes.
- `commit` is used to store the state of all the staged files with an optional message.

- `pull` is used to copy the code from a remote branch in the current project.
- `push` is used to move the committed changes to a remote repository.

Prerequisite:

Install the following tools to set up a GitHub project:

- The `git` tool is used to set up an environment to execute `git` commands, so download and install the git tool.

Optional

- `Putty` is a tool for windows to generate SSH keys. Download and install the Putty tool as per your Windows OS type (32 or 64).

Clone a Repository Using SSH Link

Cloning is the process of creating a local copy of a remote repository. A GitHub repository can be cloned using an SSH or HTML link. SSH is a protocol to securely communicate with a server using a handshake mechanism and public-key cryptography technique. A secure connection allows you to execute `git`

instructions from the command line (terminal) without confirming GitHub credentials for every push/pull operation.

Follow the below steps to create an SSH public/private key pair and add the public key to the `GitHub` account:

1. Add GitHub account details

`git` maintains a global and local (per project) configuration file that is used to store required details like email, user-name, editing software, etc. Update the value of your GitHub account user name and email in the `git` configuration file:

bash

```
git config --global user.name "Your name here"
git config --global user.email "your_email@example.com"
```

2. Generate SSH keys

SSH keys can be generated using `git` bash or the Putty tool to generate keys. Follow the below steps to generate SSH keys on Mac or Linux:

If an SSH key already exists then you can use the existing key.

- Generate SSH Keys using `ssh-keygen`:

bash

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

Press enter for every input to generate the key.

> *A `passphrase` can be used to provide an extra layer of security to SSH keys. If a passphrase if used then `git` will prompt to enter the `passphrase` before using the SSH key, although `passphrase` can be saved in `ssh-agent` like key-chain access to automatically provide the value of the `passphrase`.*

- Copy the generated SSH key:

bash

```
# Mac
pbcopy < ~/.ssh/id_rsa.pub
```

On Linux, get the content of the SSH key using the `cat` command:

bash

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

For Windows, the key can be generated using `git` `bash` (or putty), so open the `git` `bash` console and type:

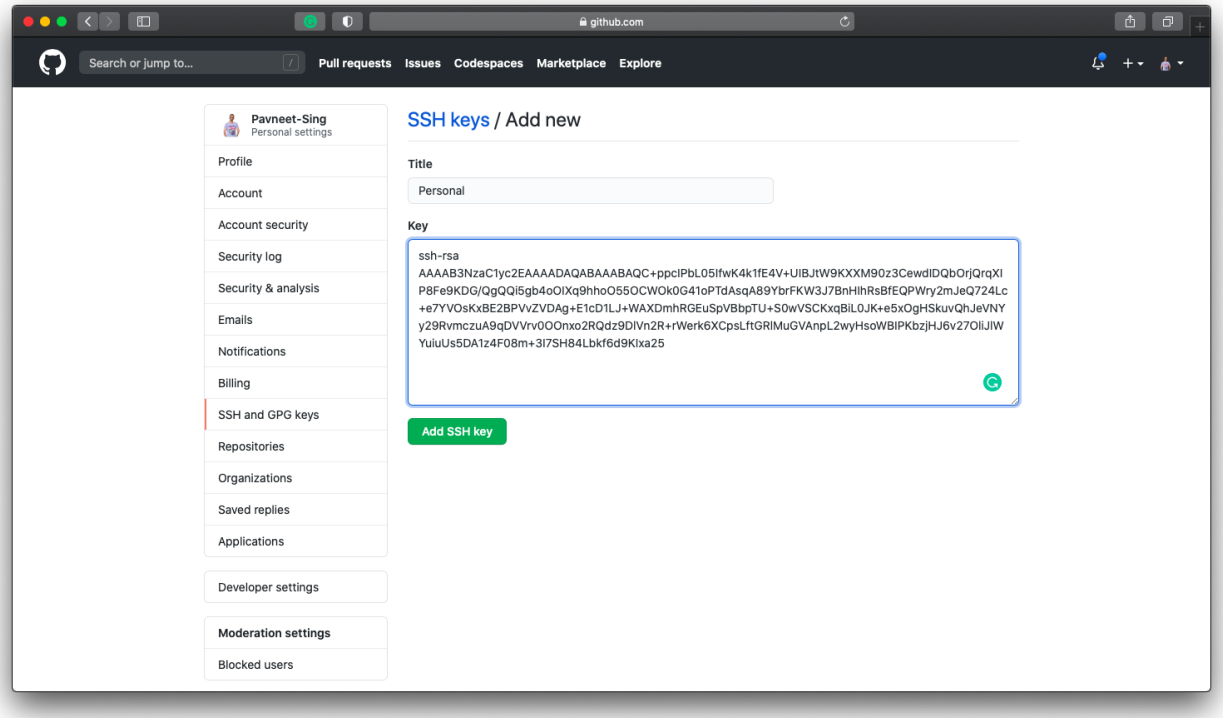
bash

```
ssh-keygen -t rsa
```

Now copy the content of `your_home_directory/.ssh/id_rsa.pub` file.

3. Add SSH to GitHub account

Open settings from the profile icon, select **SSH and GPG keys**, and add the copied SSH key:



4. Clone Project

Use the `git clone` command to clone the project in the current directory, using an `SSH` link:

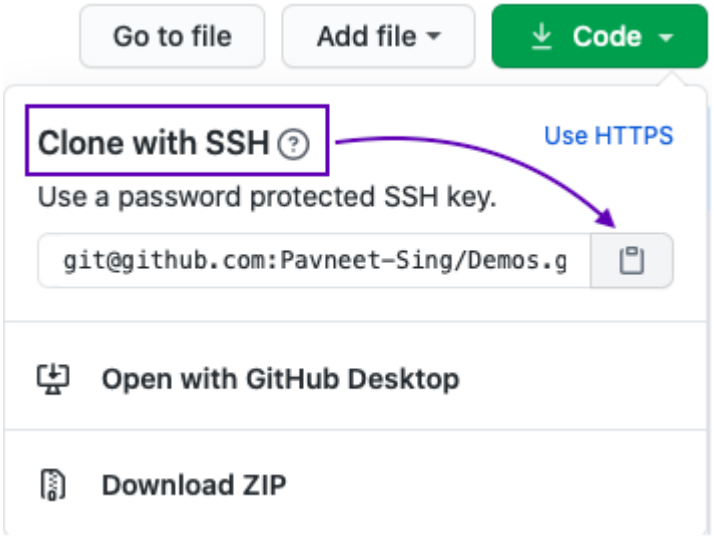
bash

```
git clone git@github.com:/UserName/RepoName.git
```

> *An SSH link of a GitHub repository can be only be retrieved via a logged-in GitHub account.*

Clone a Repository Using HTTP Link

An HTTP URL is used to clone any public or private repository from a GitHub account. The major drawback of using an HTTP URL is that `git` will ask for a user-name and password for authentication before performing any operation on a GitHub repository. To clone a GitHub repository, copy the HTTP URL of the GitHub repository



execute the `clone` command:

bash

```
git clone https://github.com/UserName/RepoName.git
```

Clone a Repository Using GitHub CLI

The GitHub CLI brings capabilities of GitHub web UI to the command line to perform operations like creating a pull request, track issues, fork a repository, etc. Use the `auth` command to authenticate the account and clone the project using the `clone` command:

bash

```
gh auth login
gh repo clone UserName/RepoName
```

- The `auth` command can take a `--web` flag to authenticate using a browser. It can also accept authentication token using a `--with-token` flag.

bash

```
gh auth login --with-token < myGitHubToken.txt
```

- The `clone` a command allows to omit the current user name and can work with the repository name associated with the logged-in user account:

```bash
gh repo clone RepoName
```

Alternative Options to Set Up a Repository

There are two other official ways to set up a GitHub repository:

1. Use the download option to get a compressed file of a codebase and uncompress it.
2. Install the GitHub Desktop tool and choose the **Open with GitHub Desktop** option on a repository.

Run a Cloned React Project

The `node_modules` directory is not a part of a cloned repository and should be downloaded using the `npm install` command to download all the direct and transitive dependencies mentioned in the `package.json` file:

```sh
# make sure that you are in the root directory of the project, use" pwd" c
cd RepoName
npm install
```

It will take some time to download all the dependencies into a `node_modules` directory, and after the download completion, run the project:

```sh
npm start
```

Tips

- A `node_modules` directory can take up more than 200MB, so it should not be a part of a repository.
- If `node_modules` is already a part of a repository then it can be removed using `git rm -r --cached node_modules` command, though make sure to commit and push the changes to the remote server first.

Conclusion

A GitHub repository can be cloned using `git` and `gh` tools. Use an SSH key to auto-authenticate. There are many free software

available to manage `git` projects. Try out the `GitHub CLI` tools to bring all the features of the GitHub UI to terminal. Happy coding!