

Ah! You're asking:

"How to work with the GitHub HTTPS link using CLI (command-line)?"

Let's break it down clearly:


1 How to Use GitHub HTML (HTTPS) Link with Git CLI

2 1. Clone Repo via HTTPS (in CLI)

bash

CopyEdit

```
git clone https://github.com/Suupratik/My-practice2.git
```

 This will download your project folder locally.

3 2. Set Remote Origin via HTTPS

If your folder is already initialized (you've done git init), then connect:

bash

CopyEdit

```
git remote add origin https://github.com/Suupratik/My-practice2.git
```

 To switch from SSH to HTTPS:

bash

CopyEdit

```
git remote set-url origin https://github.com/Suupratik/My-practice2.git
```

4 3. Add, Commit & Push (Using HTTPS)

bash

CopyEdit

```
git add .
```

```
git commit -m "First commit via HTTPS"
```

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

 **It will now ask:**

```
makefile
```

```
CopyEdit
```

```
Username: your-github-username
```

```
Password: <paste your personal access token here>
```

 You **cannot use your GitHub password anymore**, so:

5 How to Generate a GitHub Personal Access Token (PAT)

1. Go to: <https://github.com/settings/tokens>
 2. Click "**Generate new token (classic)**"
 3. Give a name like "CLI Push Token"
 4. Select scopes:
 - ☒ repo
 - ☒ workflow (optional for CI/CD later)
 5. Click **Generate token**
 6. **Copy the token** — use it as your password in CLI
-

6 4. Cache the Credentials (Optional)

So you don't have to type the token every time:

```
bash
```

```
CopyEdit
```

```
git config --global credential.helper cache
```

Or store it permanently (⚠ less secure):

bash

CopyEdit

git config --global credential.helper store

7 Summary: Git via CLI using HTTPS

Action	Command
Clone via HTTPS	git clone https://github.com/username/repo.git
Add remote HTTPS	git remote add origin https://github.com/...
Push via HTTPS	git push -u origin main
Token required (not pwd)	Use GitHub PAT instead of password
Cache token (optional)	git config --global credential.helper cache