

origin

Wofür steht eigentlich "origin"?

Wenn du ein Repository mit `git clone <URL>` von GitHub, GitLab oder Bitbucket auf deinen Rechner holst, passiert im Hintergrund automatisch Folgendes:

1. Git lädt den Code herunter.
2. Git merkt sich die URL, von der der Code kam.
3. Weil sich niemand lange URLs wie `https://github.com/user/projekt-name.git` merken oder sie ständig eintippen will, gibt Git dieser URL den Standard-Spitznamen `origin` (engl. für *Ursprung* oder *Herkunft*).

1. Beim Anlegen eines Remotes (`git remote add...`)

Wenn du ein brandneues Projekt lokal auf deinem Rechner startest (per `git init`) und es danach auf GitHub hochladen willst, existiert diese automatische Verknüpfung noch nicht. Du musst sie selbst anlegen:

```
git remote add origin https://github.com/deinname/dein-projekt.git
```

Dieser Befehl sagt Git: „Hey, merk dir bitte diese lange GitHub-URL unter dem kurzen Namen `origin`.“ (Hinweis: Du könntest das Ding hier auch `banane` oder `server` nennen. Aber `origin` hat sich als weltweiter Standard etabliert, so wie `main` oder `master` für den Haupt-Branch).

2. Beim Pushen eines neuen Branches (`git push -u origin...`)

Wenn du einen neuen lokalen Branch namens `feature-xyz` erstellt hast und ihn das erste Mal hochladen willst, sagst du:

```
git push -u origin feature-xyz
```

Das bedeutet übersetzt: „Git, pushe meinen lokalen Branch `feature-xyz` zu dem Server, den ich vorhin `origin` genannt habe, und richte dort einen gleichnamigen Branch ein.“ (Das `-u` sorgt nur dafür, dass sich Git diese Verbindung für die Zukunft merkt, sodass danach ein einfaches `git push` reicht).