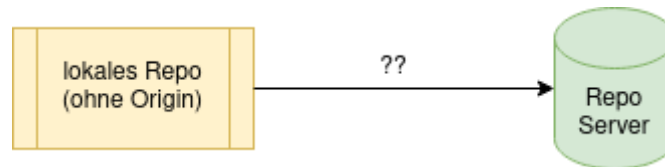


Ein lokales Repo auf einen Server Pushen

Wenn man (wie wir) zunächst ein lokales Repo angelegt hat, möchte man das möglicherweise auf einem zentralen Server wie codeberg, github oder gitea.schule.social vorhalten, um das Repo gemeinsam zu bearbeiten, oder auf verschiedene Rechner klonen zu können.

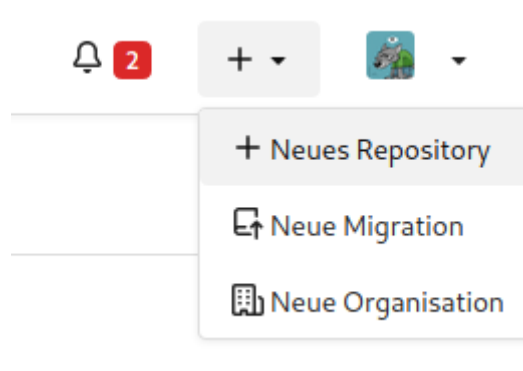


Um dieses Ziel zu erreichen geht man wie folgt vor:

- Man erzeugt auf dem Server ein Repo, in welches das lokale Repository gepusht werden soll
- Man setzt händisch den Origin des lokalen Repos auf die Adresse des auf dem Server erzeugten Repos
- Nun kann man das lokale Repo zum "neuen Origin" pushen.
- Von nun an arbeitet man im lokalen Klon weiter, indem man regelmäßig pullt und pusht.

Am Beispiel Gitea

Anmelden bei Gitea. Repository erstellen:



Besitzer *



Some organizations may not show up in the dropdown due to a maximum repository count limit

Repository-Name *

webseite **1**

Ein guter Repository-Name besteht normalerweise aus kurzen, unvergesslichen und einzigartigen Schlagwörtern.

Sichtbarkeit

☐ In privates Repository umwandeln

Nur der Besitzer oder Organisationsmitglieder mit entsprechender Berechtigung, werden in der Lage sein, es zu sehen.

Beschreibung

Template

Vorlage auswählen

Issue Label

Wähle ein Issue-Label-Set.

.gitignore

Wähle eine .gitignore-Vorlage aus.

Lizenz

Wähle eine Lizenz aus.

README

Default

☐ Repository initialisieren (Fügt .gitignore, License und README-Dateien hinzu)

Standardbranch

master

Signaturvertrauensmodell

Standardvertrauensmodell

Template

☐ Repository zu einem Template machen

2 Repository erstellen

Abbrechen

Im 3. Schritt wird die Anleitung für das weitere Vorgehen direkt angezeigt:

Kurzanleitung**Dieses Repository klonen** Benötigst du Hilfe beim Klonen? Öffne die [Hilfe](#).

HTTPS

<https://gitea.schule.social/sbel/webseite.git>**Erstelle ein neues Repository von der Kommandozeile aus**

```
touch README.md
git init

git add README.md
git commit -m "first commit"
git remote add origin https://gitea.schule.social/sbel/webseite.git
git push -u origin master
```

Bestehendes Repository via Kommandozeile pushen

```
git remote add origin https://gitea.schule.social/sbel/webseite.git
git push -u origin master
```

Für uns trifft der zweite Fall zu - wir wollen ein bestehendes Repo pushen, führen die angegebenen Befehle also in unserem lokalen Repo aus:

```
[frank@rita webseite]$ git remote add origin https://gitea.schule.social/sbel/webseite.git
[frank@rita webseite]$ git push -u origin master
error: Src-Refspec master entspricht keiner Referenz.
error: Fehler beim Versenden einiger Referenzen nach 'https://gitea.schule.social/sbel/webseite.git'
[frank@rita webseite]$ git push -u origin main
Username for 'https://gitea.schule.social': sbel
Password for 'https://sbel@gitea.schule.social':
Objekte aufzählen: 7, fertig.
Zähle Objekte: 100% (7/7), fertig.
Delta-Kompression verwendet bis zu 4 Threads.
Komprimiere Objekte: 100% (4/4), fertig.
Schreibe Objekte: 100% (7/7), 555 Bytes | 185.00 KiB/s, fertig.
Gesamt 7 (Delta 0), Wiederverwendet 0 (Delta 0), Pack wiederverwendet 0
remote: . Processing 1 references
remote: Processed 1 references in total
To https://gitea.schule.social/sbel/webseite.git
 * [new branch]      main -> main
Branch 'main' folgt nun Remote-Branch 'main' von 'origin'.
```

Die Fehlermeldung `error: Src-Refspec master entspricht keiner Referenz.` rührt daher, dass versucht wurde, den lokalen `master`-Branch zum Origin zu pushen, der lokale Branch heißt aber `main`, an dieser Stelle muss man also gegebenenfalls eingreifen.

Das Ergebnis auf dem Server sieht nun folgendermaßen aus:

 sbcl / webseite

Beobachten beenden 1 Favorisieren 0 Fork 0

<> Code Issues 0 Pull-Requests 0 Projekte 0 Releases 0 Wiki Aktivität Einstellungen

2 Commits 1 Branch 21 KiB

Branch: main Commit graph

2 Commits (main) Commits durchsuchen... ☐ Alle Branches Suchen

Autor	SHA1	Nachricht	Datum
 John Doe	893b24e60a	CSS hinzugefügt	vor 1 Stunde
 John Doe	bb0d027bd6	Erster commit	vor 5 Stunden

Mit dem Befehl `git clone https://gitea.schule.social/sbel/webseite.git` kann nun jeder dieses Repo klonen und bearbeiten. Änderung auf den Server zu pushen ist allerdings erst nach Authentifizierung möglich. Allerdings könnte man im Klon den Origin erneut ändern (z.B. zu einem GitHub Repo) und seine Änderungen dann dorthin pushen. In diesem Fall hat man einen Fork erzeugt.

Wichtig und sehr cool: Kein Fork vergisst seine Geschichte - da alle Commits durch eindeutige Hashes bezeichnet werden und alle Klone immer alle Commits enthalten können Repos über lange Zeit über verschiedene Server und Rechner geforkt und bearbeitet werden, und es ist (bei ausreichend systematischer Arbeitsweise der Autoren) jederzeit möglich, sie zu einem späteren Zeitpunkt wieder zusammenzuführen.

From:
<https://www.info-bw.de/> -

Permanent link:
<https://www.info-bw.de/faecher:informatik:oberstufe:git:lokal2remote:start?rev=1619639315>

Last update: **28.04.2021 19:48**

