git filter-repo

Jeder hat schon einmal versehentlich eine Config-Datei mit sensiblen Daten in ein Repository gepusht, und das ist auch okay! Wichtig ist dabei nur, dass man diese Datei danach wieder spurenlos aus dem Repository entfernt, damit geheime Informationen nicht in die falschen Hände geraten können.

Der Befehl `git filter-repo` ist ein leistungsstarkes Werkzeug, das verwendet wird, um **Git-Repositorys** zu bereinigen und zu transformieren. Es ist schneller und flexibler als der frühere `git filter-branch`. Mit `git filter-repo` kannst du z. B. sensible Daten aus der Historie entfernen, Branches umbenennen oder die Struktur des Repositorys verändern.

Vorbereitungen

Da `git filter-repo` nicht standardmäßig bei Git mitgeliefert wird, musst du es zuerst installieren.

- Auf Linux (Arch): `pacman -S git-filter-repo`
- Auf Windows: Eine Anleitung zur Installation findest du **hier** (https://github.com/newren/git-filter-repo/blob/main/INSTALL.md#simple-installation).

Wichtig

- **Erstelle ein Backup deines Repositorys**: Das Entfernen einer Datei aus der Git-Historie überschreibt Commits und kann nicht rückgängig gemacht werden.
- **Koordiniere dich mit deinen Teamkollegen**: Das Umschreiben der Historie erfordert einen **Force-Push** auf das Remote-Repository, daher ist es wichtig, dass alle anderen Beteiligten dies wissen. Eventuell muss für das betroffene Git-Repository die Branch Protection deaktiviert werden.

Speichern der Url mit Blob-Hash

Um nach diesem Vorgang einfach überprüfen zu können, ob die unerwünschte Datei tatsächlich entfernt wurde, sollte der direkte Link zu dieser Datei aufgehoben werden. Ein Beispiel zu solch einem Link sieht folgendermaßen aus:

https://git.agrarforschung.at/datenpool/Datapool/-/blob/129fa4adc48ca2ac3623dc4f3369a73497ec1a64/Datapool/server/config/config.bak.js?ref_type=heads
(https://git.agrarforschung.at/datenpool/Datapool/-/blob/129fa4adc48ca2ac3623dc4f3369a73497ec1a64/Datapool/server/config/config.bak.js?ref_type=heads)

Wichtig ist, dass dieser Link den Blob-Hash beinhaltet, welcher direkt auf die unerwünschte Datei verweist. Nach Durchführung der folgenden Schritte sollte dieser Link anstatt der Datei einen 404 Error anzeigen.

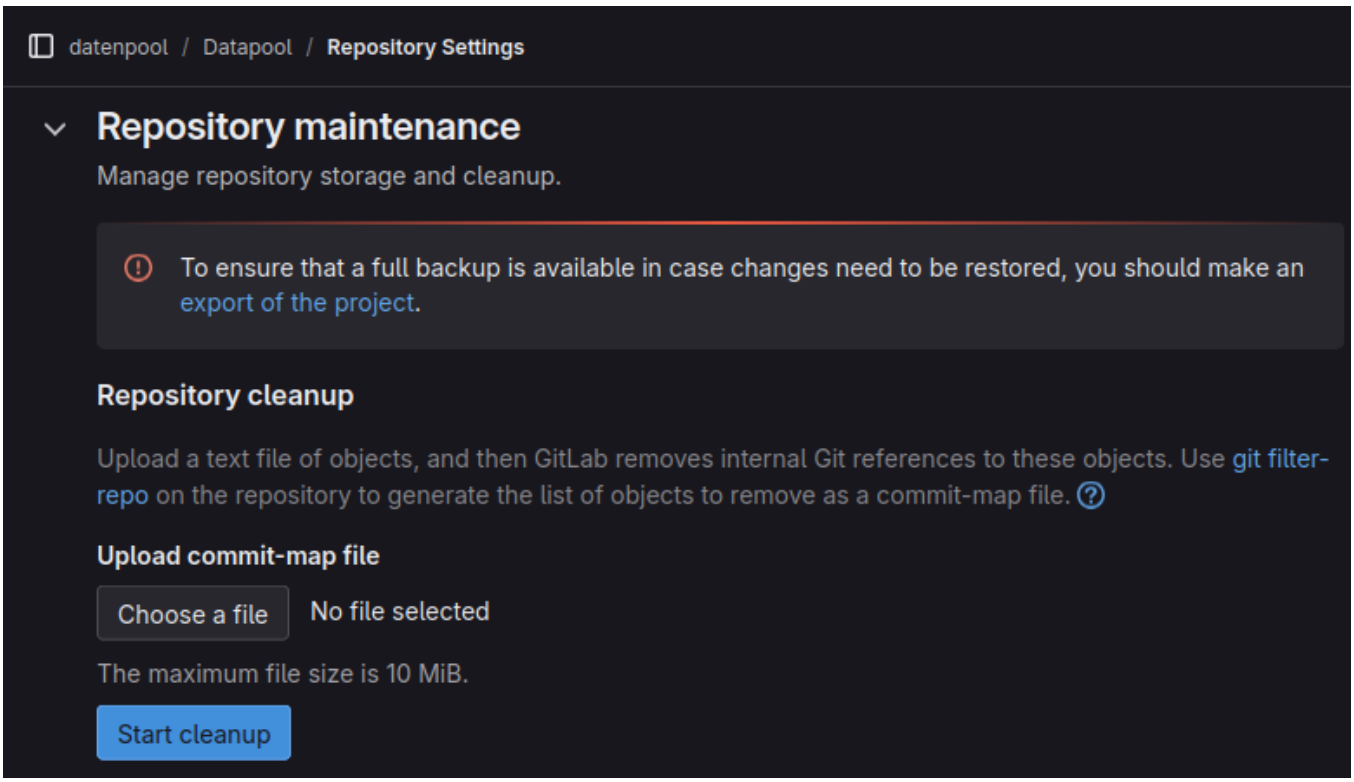
Entfernen einer unerwünschten Datei

1. Mit folgendem Befehl, wird die datei **config.bak.js** aus allen Commits entfernt:
   - `git filter-repo --path config.bak.js --invert-paths`
   - **`--path config.bak.js`**: Gibt die Datei an, die entfernt werden soll.
   - **`--invert-paths`**: Entfernt die angegebene Datei aus der gesamten Historie.
2. Datei lokal löschen oder der .gitignore hinzufügen (optional)
   1. Auf Existenz der Datei überprüfung (Linux): `git ls-files | grep config.bak.js`
      - Falls die Datei hier noch auftaucht, kann diese mittels `git rm --cached path/to/config.bak.js` aus dem lokalen Repository gelöscht werden
   2. Aktualisiere **`.gitignore`**
      - Damit die Datei nicht erneut gepusht wird, sollte sie der `.gitignore`-Datei hinzugefügt werden

3. Force-Push zum Remote-Repository
    - Da du die Historie überschrieben hast, musst du die Änderungen nun mit einem **Force-Push** in das Remote-Repository hochladen: `git push origin --force --all`
4. Tags Force-Pushen (falls erforderlich)
    - Wenn du Tags hast, die auf Commits zeigen, die die Datei config.bak.js enthalten haben, solltest du auch die Tags force-pushen: `git push origin --force --tags`

Bereinigung des Repository-Caches

Nach der Ausführung von `git filter-repo` ist die unerwünschte Datei aus allen Commits entfernt. Falls jedoch der Blob-Hash dieser Datei bekannt ist, kann nach wie vor auf die Datei zugegriffen werden. Gitlab stellt eine Funktion zur automatischen Bereinigung des Repository-Caches zur Verfügung (unter Settings -> Repository -> Repository Maintenance).

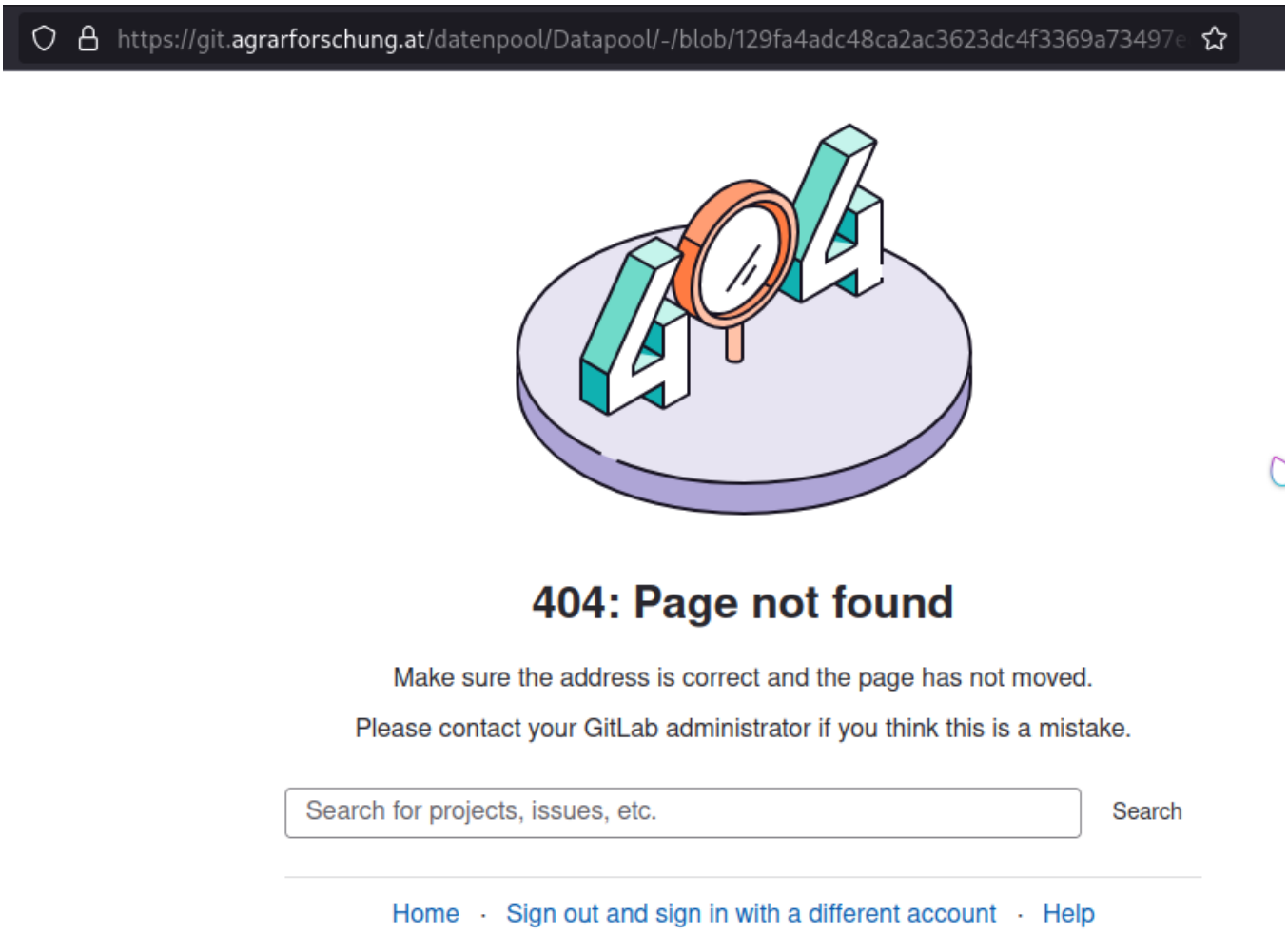


(https://doku.agrarforschung.at/uploads/images/gallery/2024-11/tL0image.png)

Im lokalen Repository findet sich unter `.git/filter-repo/` die Datei `commit-map.json`. Diese kann in der Ansicht oben ausgewählt werden. Mit einen Klick auf "Start cleanup" werden alle nicht-referenzierten Dateien aus dem Git Cache entfernt.

Überprüfung

Die unerwünschte Datei sollte nun entfernt sein



(https://doku.agrarforschung.at/uploads/images/gallery/2024-11/v8Iimage.png)

Version #10

Erstellt: 2024-11-27 14:33:12 CET von Andreas Mattes

Zuletzt aktualisiert: 2026-09-23 11:24:01 CEST von roland.neissl