GIT Ass

Git Reset to Commit

`git reset --hard <tag/branch/commit id>`

Notes:

- `git reset` without the `--hard` option resets the commit history, but not the files. With the `--hard` option the files in working tree are also reset.
- If you wish to commit that state, so remote repository also points to rolled back commit do: `git push <reponame> -f`

Git Reset to Commit Workaround

Rename your current master branch:

`git branch -m crazyexperiment`
Check out your good commit:

`git checkout c2e7af2b51`
Make your new master branch here:

`git checkout -b master`
Now you still have your crazy experiment around if you want to look at it later, but your master branch is back at your last known good point, ready to be added to. If you really want to throw away your experiment, you can use:

`git branch -D crazyexperiment`

Bestehendes Repo pushen

Sie haben bereits ein Git-Repository auf ihrem Computer. Veröffentlichen Sie es bei Bitbucket.

```
cd /path/to/my/repo
git remote add origin https://sbaOneCommerce@bitbucket.org/sbaOneCommerce/test2.git
git push -u origin --all # pushes up the repo and its refs for the first time
git push -u origin --tags # pushes up any tags
```

Änderungen hinzufügen und pushen

git status # zeigt Daten an die dem Commit hinzugefügt werden können
git commit –m „Kommentar“
git push –u origin master # Der Alias origin (Unser Bitbucket Repo) wird mit dem aktuellen Branch überschrieben

Remote hinzufügen

git remote add NAME Repolink

git remote add origin https://sbaOneCommerce@bitbucket.org/sbaOneCommerce/test2.git

git remote – zeigt alle Remotes an

Pull from Remote

git pull –u origin(source) master(branch) – pullt den Remote Branch

Push to Remote

git push –u origin(ziel) master(lokal Branch) – pusht den lokalen Branch an Remote

Regulärablauf Git

1) Neuen Branch von Master erstellen:
    a. Auf master: git checkout –b SBranch1
2) Commit IDS vergleichen (ob master & neuer Branch identisch sind):
    a. git branch –v
3) Änderungen an Daten vornehmen
4) Wenn fertig:
    a. git status – checken ob die modifizierungen angezeigt werden
    b. Dateien zum Commit hinzugfügen
        i. Alle mit git add .
        ii. Einzeln mit git add Filename
    c. Änderungen Commiten:
        i. git commit –m „Meine Neuen Änderungen“
5) Anschließend merge mit master oder weiter commiten bis merge fällig wird
    a. Merge mit master:
        i. git checkout master
        ii. git merge SBranch1

iii. Falls SBranch1 nichtmehr benötigt wird: git branch –D SBranch1

Git Username festlegen

git config --global user.name "Sebastian, OneCom"

Git Useremail festlegen

git config --global user.email s.baulechner@onecommerce.de

Git Fetch

Using refspecs explicitly:

$ git fetch origin +pu:pu maint:tmp

This updates (or creates, as necessary) branches pu and tmp in the local repository by fetching from the branches (respectively) pu and maint from the remote repository.

The pu branch will be updated even if it is does not fast-forward, because it is prefixed with a plus sign; tmp will not be.

Merge zurücksetzen (Nach Pull oder manuellem Merge)

git reset --merge ORIG_HEAD

Repository Branch löschen

```
git push origin :branch-name
```

Also, if you're on the Bitbucket website, you can remove branches you've pushed by going to the**Feature branches** tab under **Commits** on the site. There you'll find a little cog icon next to each of your pushed branches. Click that, then choose **Delete branch**. Just be sure you want to drop all the changes there!