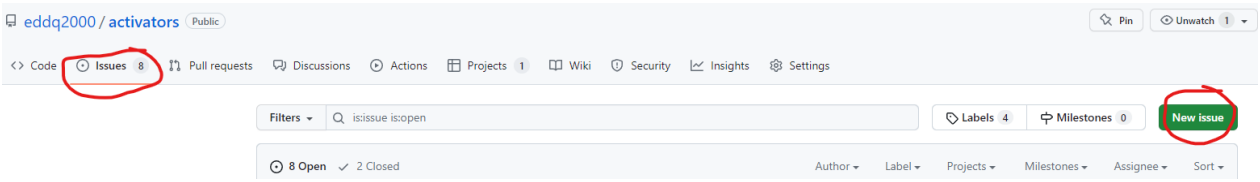
Workflow

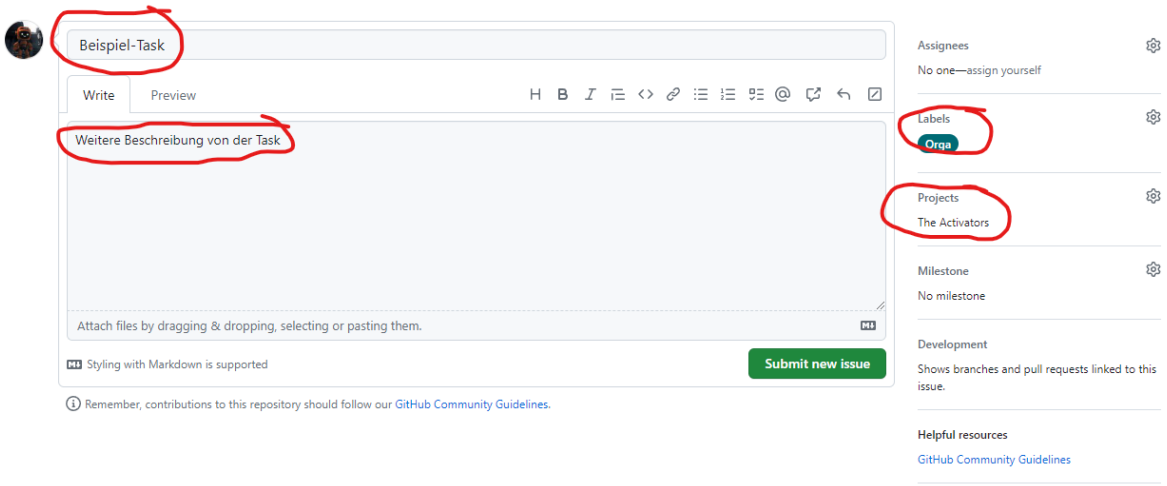
(1. **Ticket erstellen)**

Wenn es eine Aufgabe gibt wird immer als erstes ein Ticket erstellt. Im Repository kann man unter „Issues“ ein neues Ticket erstellen:

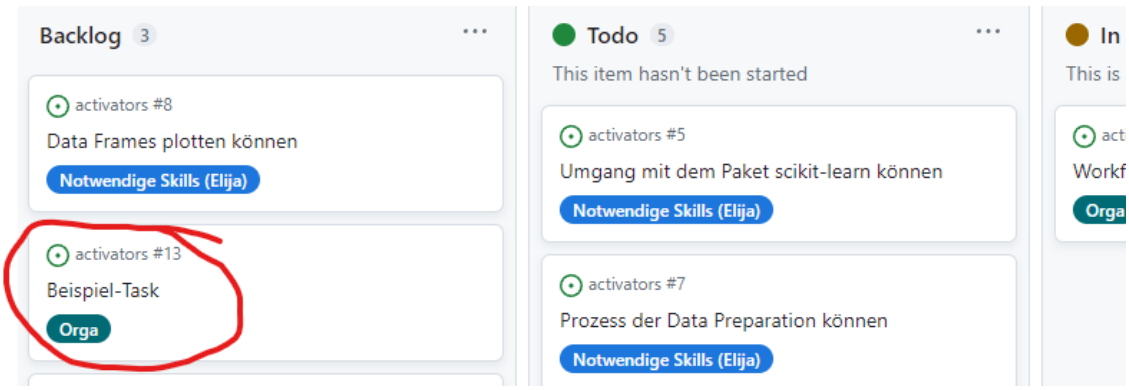


Beim Erstellen folgende Angaben machen:

- Titel geben
- Evtl. Mit Beschreibung erweitern
- Das Ticket einem Label zuordnen
- Das Ticket dem Projekt „Activators“ zuordnen

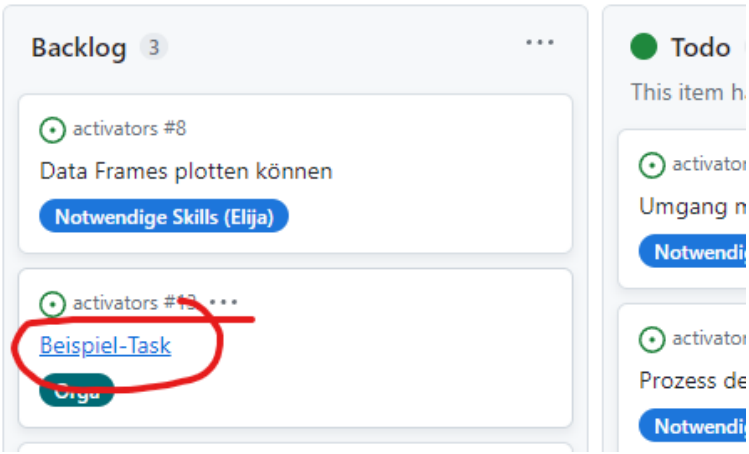


Sobald ihr das Ticket erstellt habt, sollte es auf dem projekt-Board zu sehen sein:

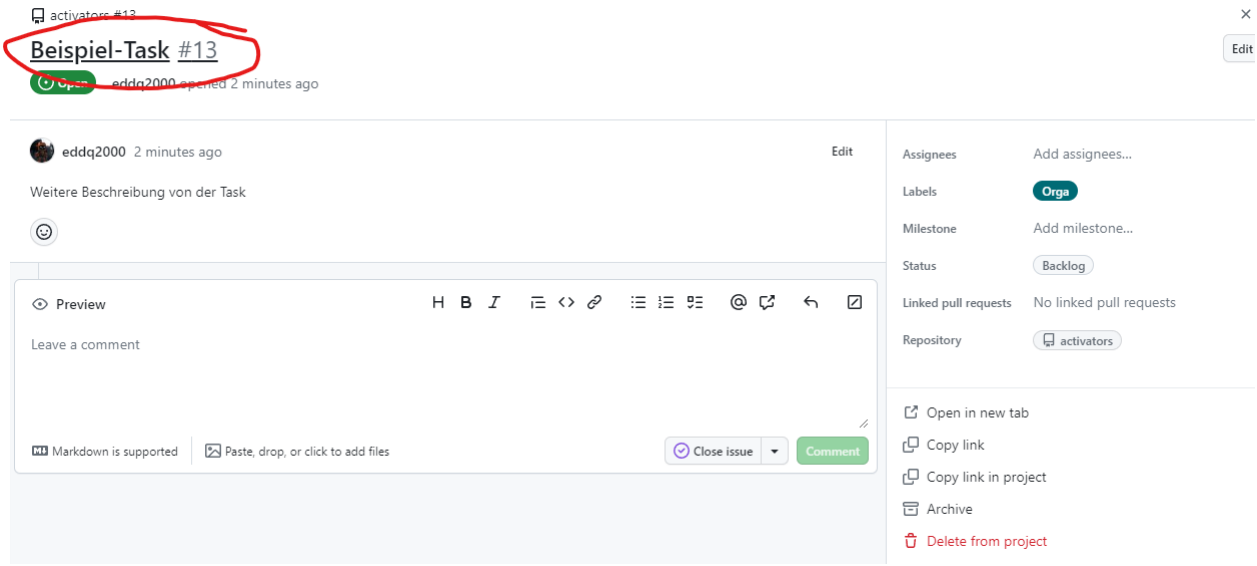


2. Ticket zuweisen

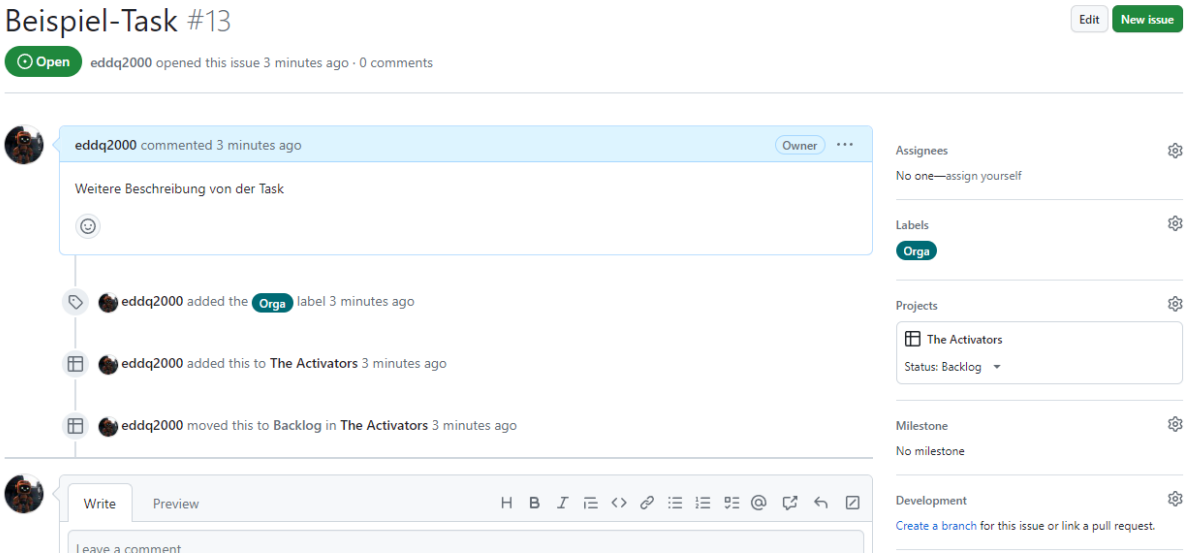
Um euch einem Ticket zuzuweisen geht ihr auf das Projekt-Board und klickt auf das Ticket, welches ihr bearbeiten wollt:



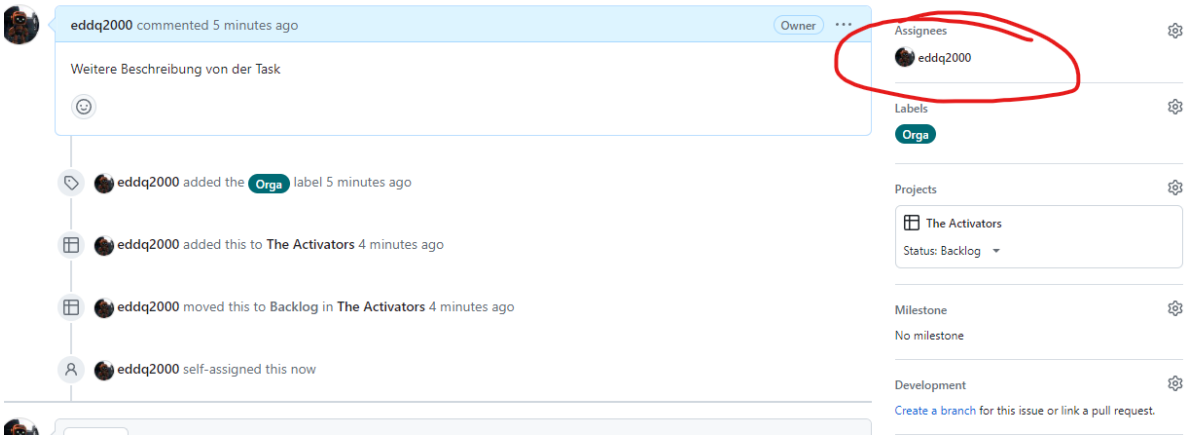
In dieser Ansicht klickt ihr dann nochmal auf den Titel des Tickets:

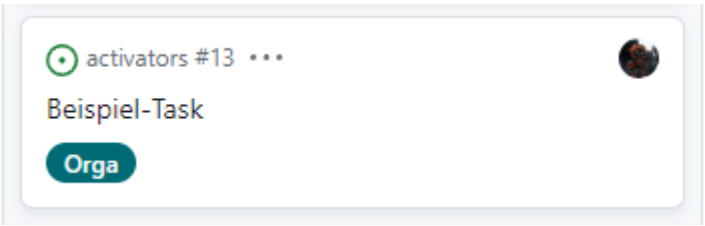


Dann sollte sich folgende Ansicht öffnen:



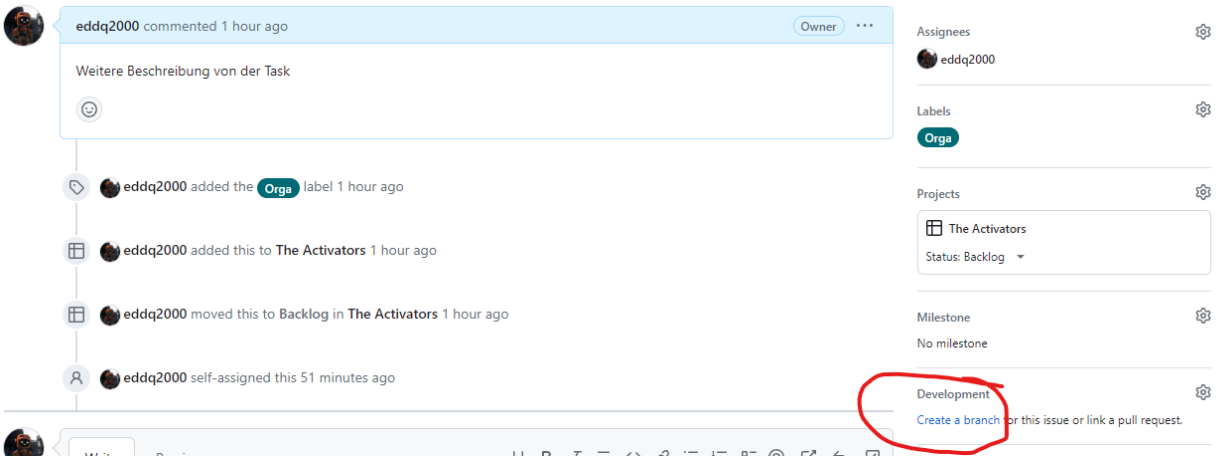
Jetzt könnt ihr euch rechts oben zu dem Ticket assignen, was bedutet, dass jeder sehen kann, dass ihr diese Aufgabe bearbeitet:



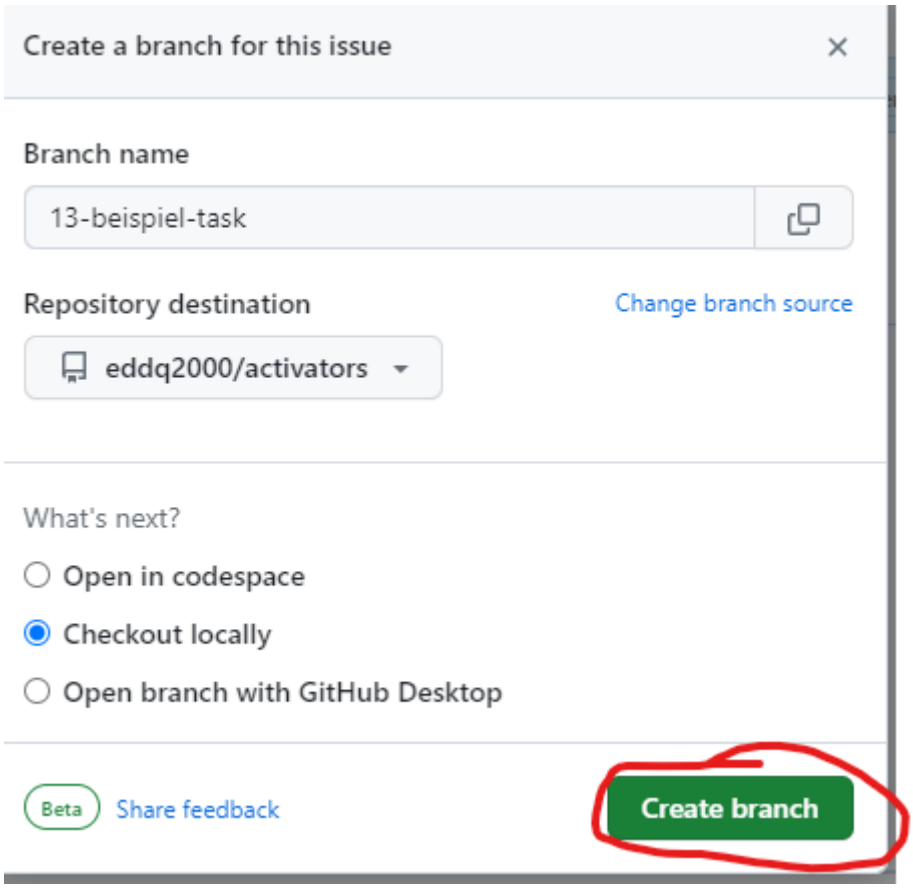


3. Branch erstellen und lokal bearbeiten

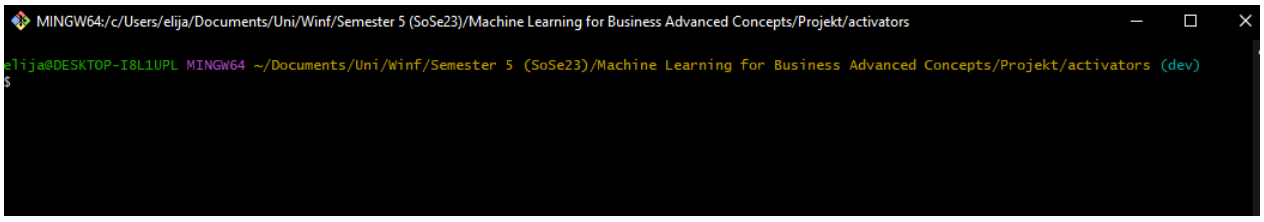
Um einen Branch zu erstellen clickt ihr in eurem Ticket einfach auf „Create a branch…“:



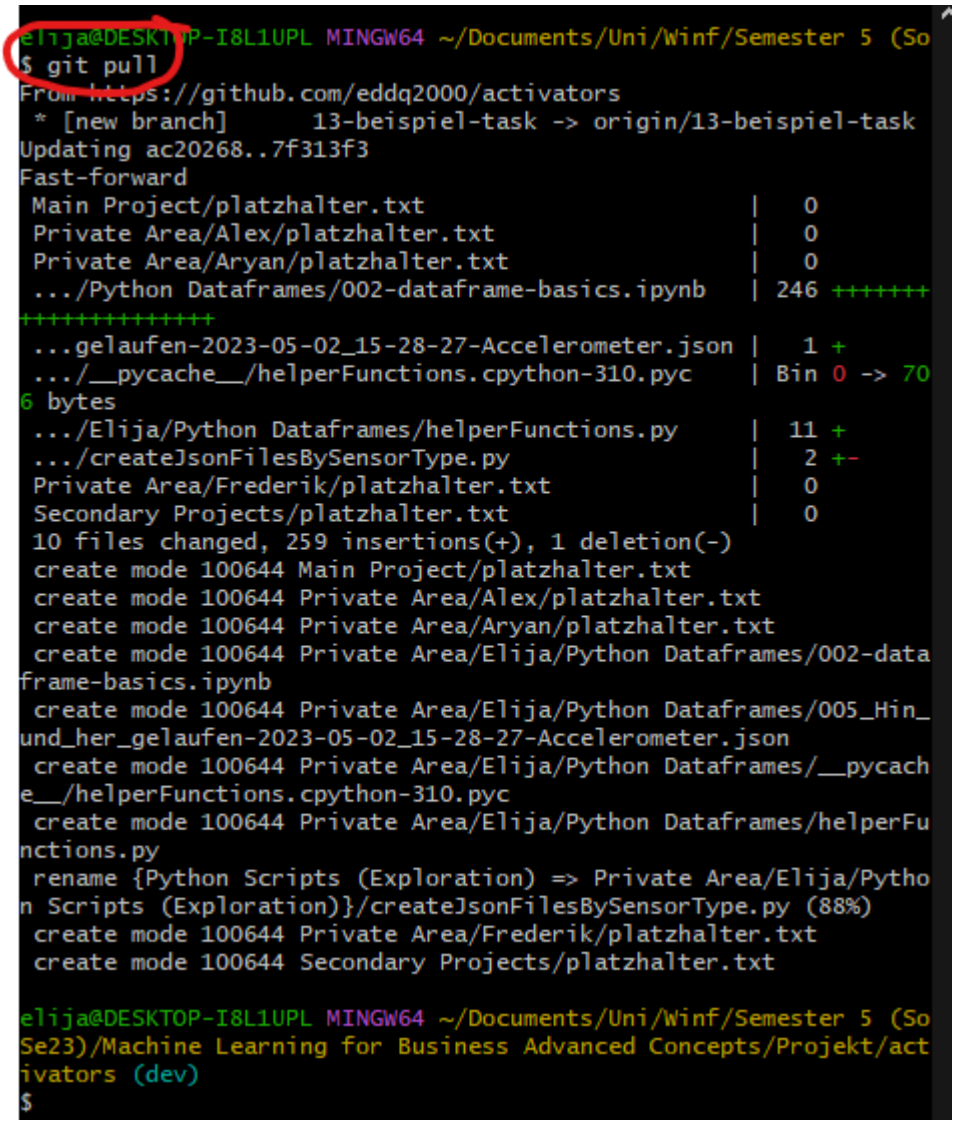
Drückt dann einfach auf „Create Branch“:



Jetzt öffnet ihr in eurem lokalen Repository die Git Bash-Konsole:

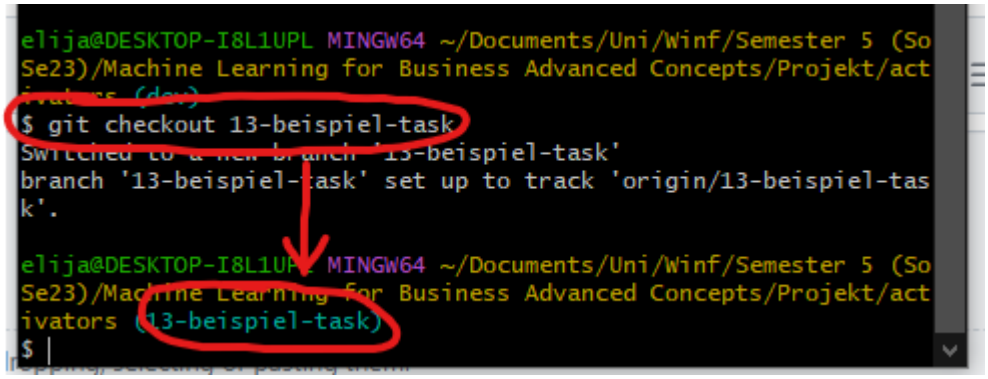


Als erstes wollt ihr den aktuellen Stand vom Repository auf Github (wo auch euer neuer branch ist) auf euer lokales Repository übertragen. Dazu gebt ihr in den Befehl *„git pull“* ein:



Jetzt wollt ihr auf den Branch wechseln den ihr erstellt habt. Dafür gebt ihr folgenden Befehl ein:

„git checkout <brnach-name>“:

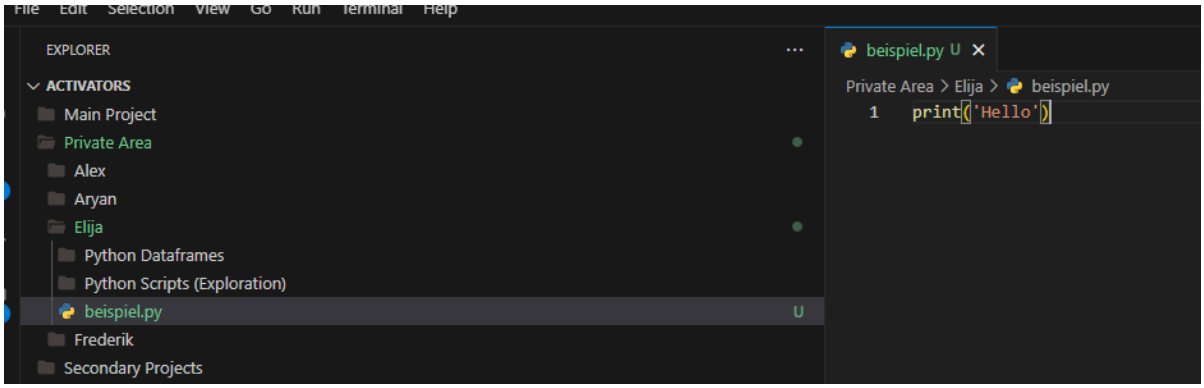


Es sollte nun in der bash Konsole in Klammern der richtige branch stehen.

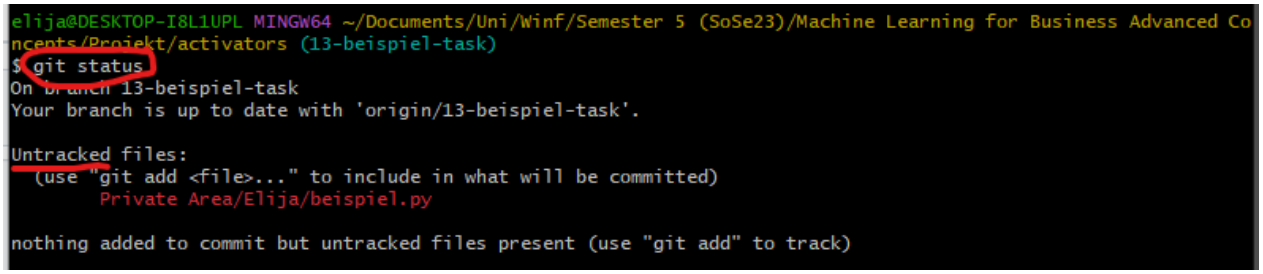
⇨ Das bedutet, dass sich dein lokales Repository jetzt auf deinem eigenen Branch befindet und alles was du hier im Repository änderst nur auf diesem Branch passiert.

4. Arbeiten auf deinem Branch

Befindest du dich auf dem richtigen Branch, kannst du anfangen Änderungen zu machen. Z.B. fügst du eine neue Datei hinzu (oder änderst was in anderen Dateien im Repository):



Gibst du in die Bash-Konsole *„git status“* eingibst, kannst du genau sehen, was für Änderungen du alles gemacht hast:

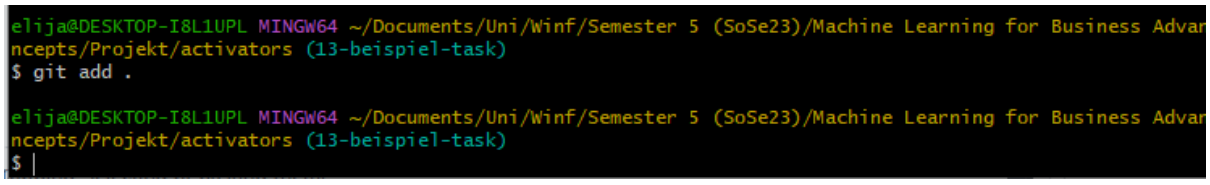


(Hier steht zum Beispiel, dass ich eine ganz neue Datei hinzugefügt habe)

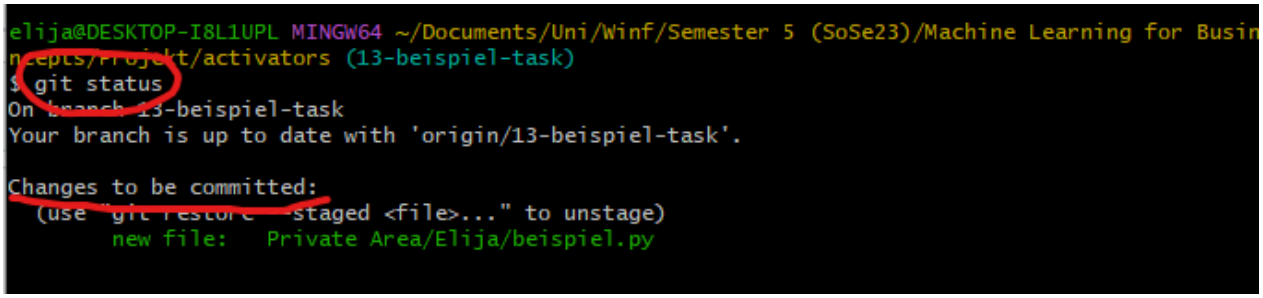
4.1 Stagen und Committen

Sobald du deine Arbeit vom Ticket erledigt hast oder einen logischen Abschnitt fertig hast, kannst du deine Änderungen in einem Commit zusammenfassen und so **speichern**.

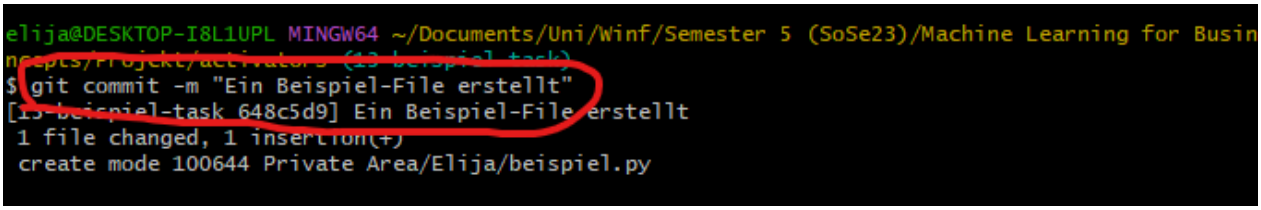
Dafür musst du zuerst deine Änderungen stagen, was so viel beduetet wie, du machst deine Änderungen dafür bereit, um sie zu committen. Das machst du mit *„git add .“* (Den Punkt nicht vergessen!). Das sieht dann so aus:



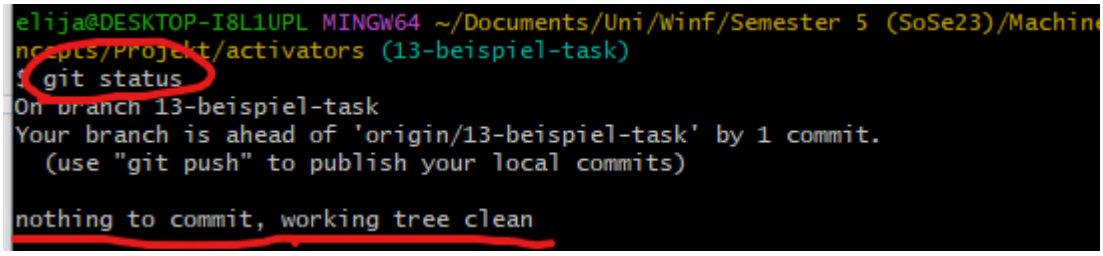
Mit „*git status*“ siehts du jetzt folgendes:



Jetzt kannst du deine Änderungen comitten: *git commit -m “<Eine Commit Nachricht>“*

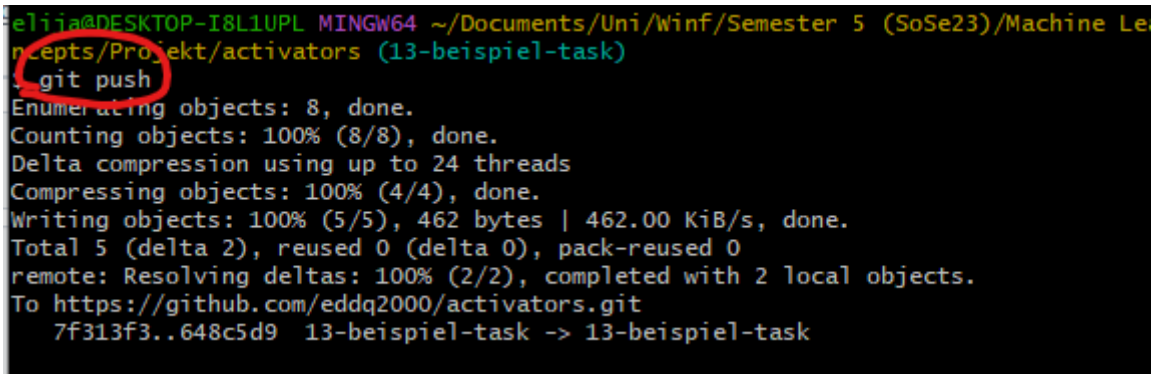


Gehst du nun nochmal auf „*git status*“, siehst du folgendes:



5. Deine Commits auf Github pushen

Um deine Commits, welche bisher nur lokal auf deinem Rechner sind, auf Github zu pushen gibst du ganz einfach „*git push*“ ein:

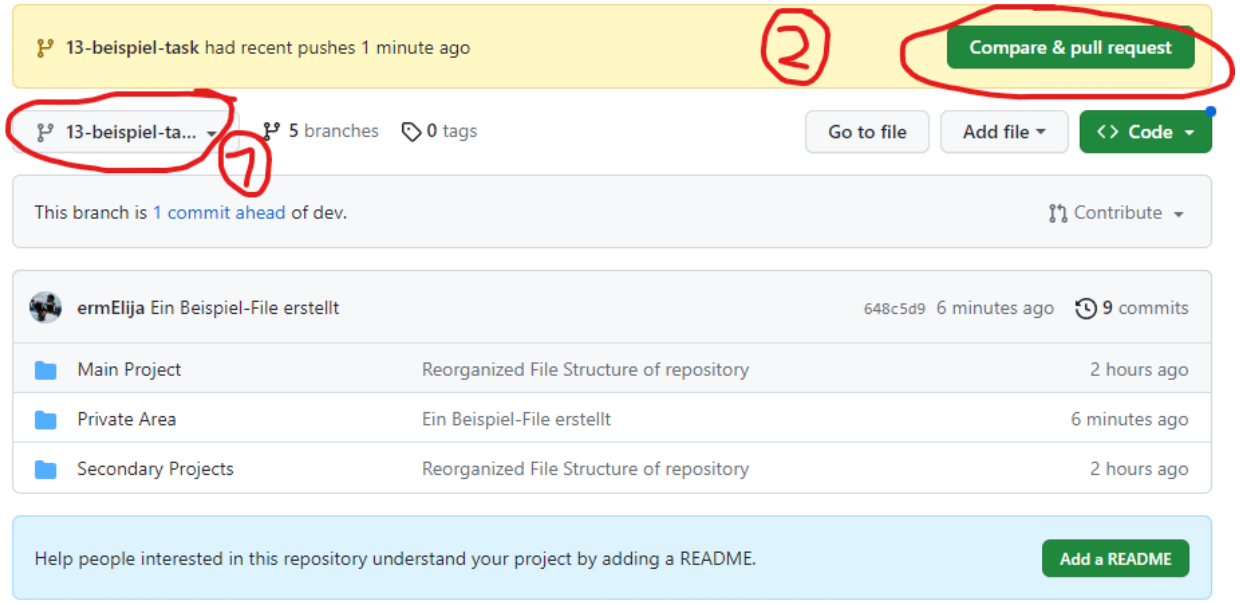


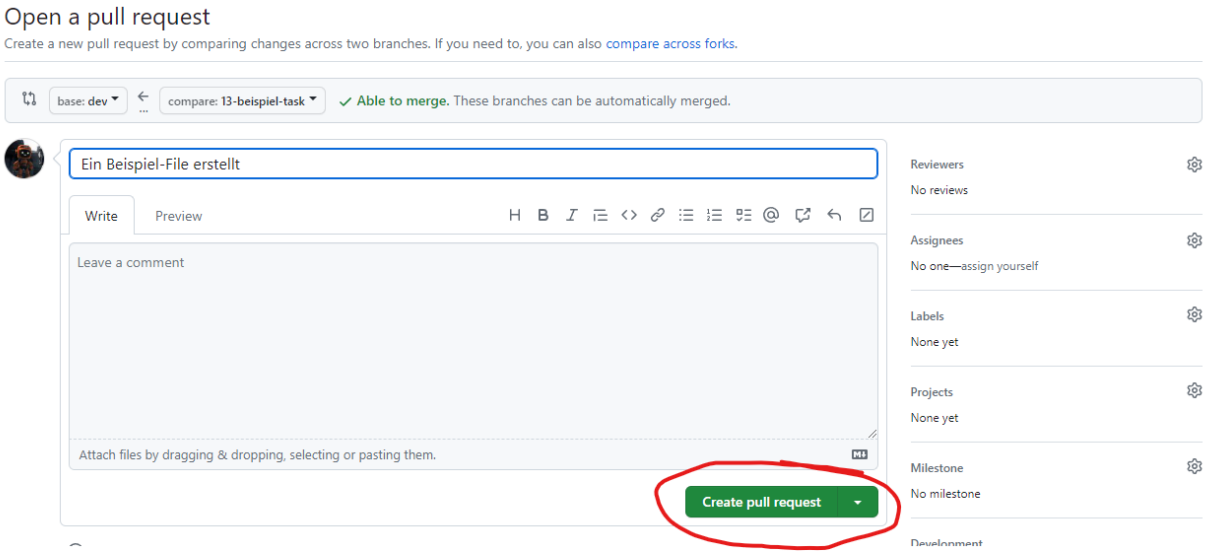
Jetzt sind deine Commits auf Github unter deinem Branch gespeichert.

6. Einen Pull-Request stellen

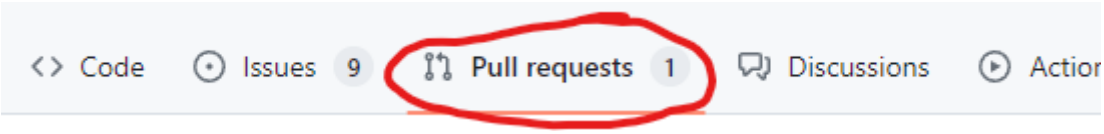
Bist du fertig mit der Arbeit für dein Ticket, dann kannst du einen Pull-Request stellen.

Bisher ist deine Arbeit noch auf dem speziellen Branch, den du für das Ticket erstellt hast. Wir wollen die Arbeit von jedem in einem Branch „dev“ zusammentragen. Dafür müssen der Branch „dev“ mit deinem Branch zusammengetragen (merged) werden. Dafür stellst du auf Github ganz einfach eine Pull-Request:





Jetzt ist ein Pull-Request erstellt:



7. Reviewen und mergen

Bevor wir einen Branch mergen, sollte nochmal jemand anders kurz über deine Commits/Änderungen drüber schauen und sagen, ob das passt, oder ob etwas fehlt, bzw. nicht funktioniert etc.

Wurde dein Ticket von einer anderen Person reviewed, kann dein branch gemercht werden und deine Änderungen sind offiziell Teil unseres main branches („dev“)