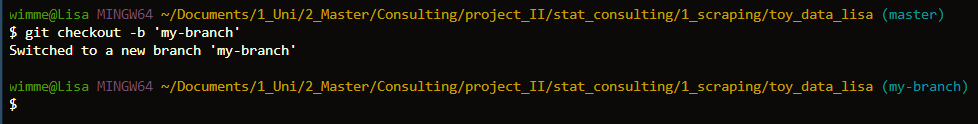
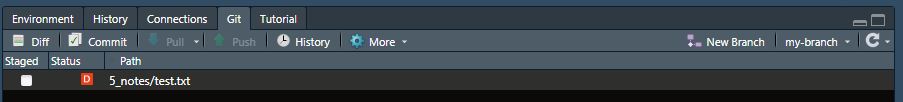
**Prozess: Auf Feature Branch arbeiten, Pull Request erstellen & in Master mergen**

Sinn und Zweck: parallel arbeiten oder Dinge weiterentwickeln, ohne Änderungen am Master vorzunehmen

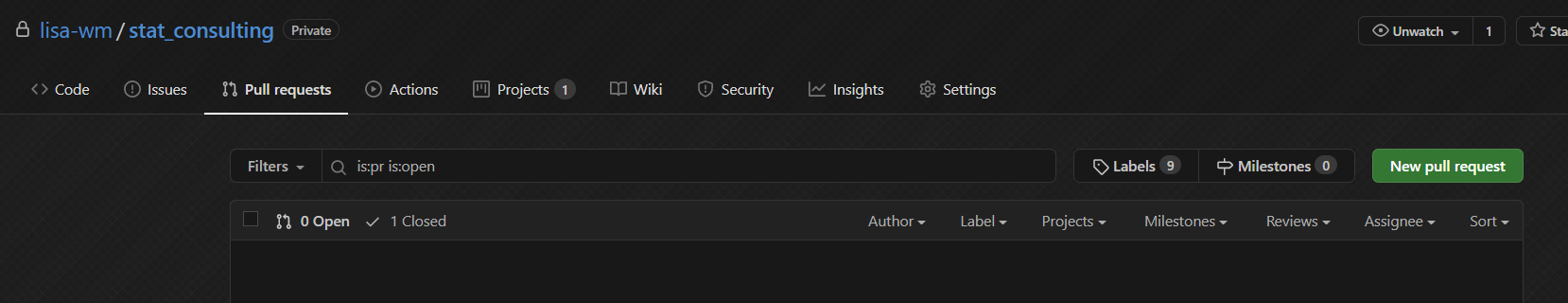
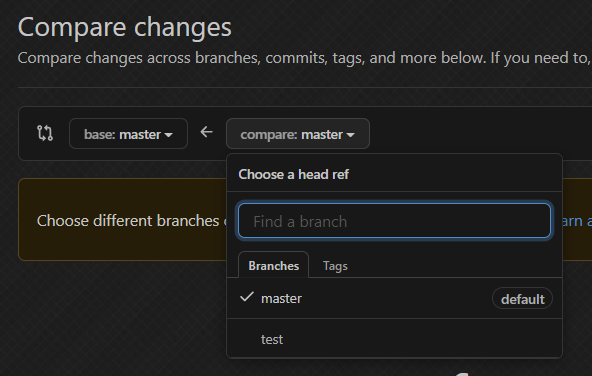
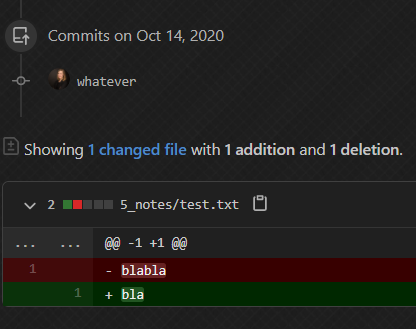
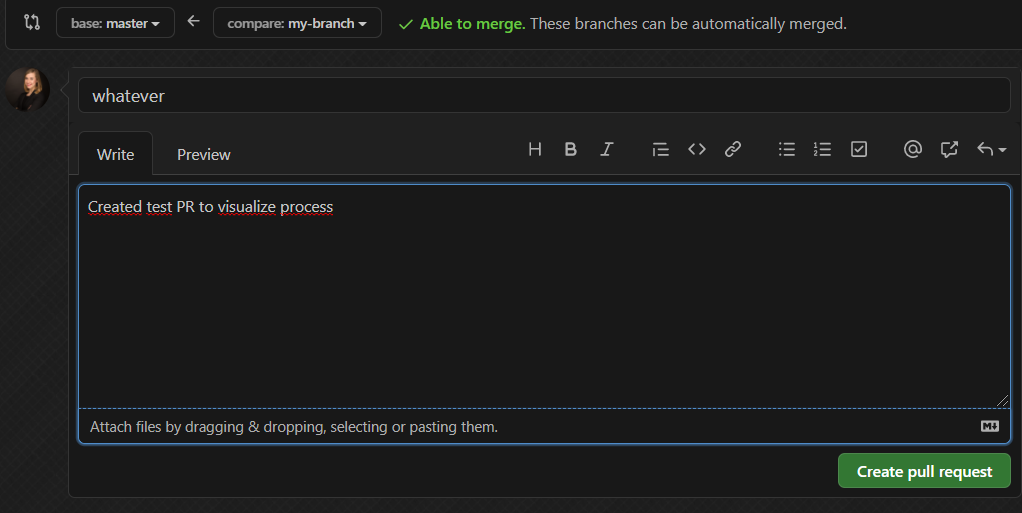
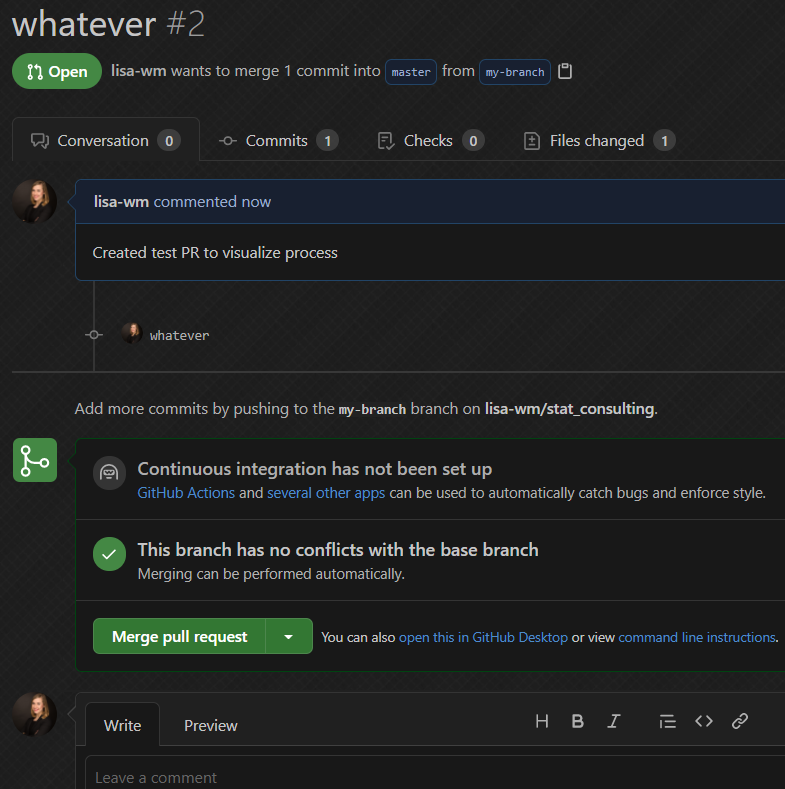
**Schritt 1: Neuen Branch erstellen und dorthin wechseln**

* Dafür ist es am besten, wenn der Master auf dem aktuellen Stand ist, damit man von dort abzweigt und nicht von einem veralteten Status. Deshalb zuerst  
  git pull
* Dann neuen Branch erstellen (-b) und gleichzeitig dorthin wechseln (checkout):  
  git checkout -b ‘my-branch‘
* Dann sollte die Endung im Terminal auf my-branch stehen:  
  

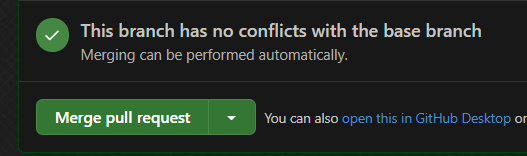
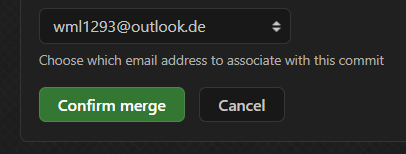
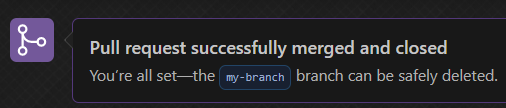
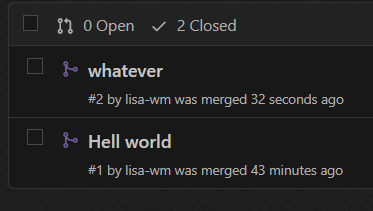
**Schritt 2: Auf Branch arbeiten**

* Ganz normal Files hinzufügen, verändern, löschen, …
* Oben rechts in der git pane sieht man die Veränderungen und auch den Branch, auf dem man arbeitet:  
  
* Wenn man die Sachen pusht, muss man den Origin mit angeben:  
  git add .  
  git commit -m ‘did some fancy shit‘  
  git push origin my-branch

**Schritt 3: Pull Request erstellen**

* Das ist die Anfrage, die auf my-branch durchgeführten Änderungen in den Master zu mergen.
* Ein PR kann beliebig viele Commits enthalten – ist generell auch immer besser, häufig zu committen. Wenn man nämlich einen Commit rückgängig machen muss, macht man halt lieber eine kleine, isolierte Änderung rückgängig als die Arbeit von Tagen
* Sieht in Git dann so aus:  
  
* Hier kann man verändern, von welchem Branch in den Master gemerged werden soll:  
  
* Und hier sieht man, was genau verändert wurde – das sollte man immer checken, um zu vermeiden, dass man unerwünschte Änderungen pusht (deshalb auch lieber kleine, häufige Commits, die sind leichter zu prüfen). Ich hab hier die Zeile mit blabla gelöscht und bla neu eingefügt:  
  
* Dann auf create pull request klicken.
* Es öffnet sich ein Kommentarfeld – da wäre es super, wenn man vermerkt, was man verändert hat (grob):  
  
* Wenn der PR dann erstellt ist, sieht das folgendermaßen aus – man sieht alle einzelnen Commits, betroffenen Files, und kann auch noch später Kommentare hinzufügen:  
  
* Grundsätzlich kann man den PR so lange stehen lassen und weiter auf my-branch arbeiten, wie man will. Ziel sollte aber immer sein, bald zu mergen – wenn der andere an denselben Files arbeitet, kann es zu Merge-Konflikten kommen.

**Schritt 4: Pull Request mergen**

* Wenn man dann bereit ist, das Ganze in den Master zu mergen, klickt man hierauf – Git sagt einem auch, ob es Konflikte gibt:  
  
* Dann noch bestätigen:  
    
  
* Hier sehen wir dann die Historie:  
  

**Schritt 5: Weiterarbeiten**

* Danach müssen alle Beteiligten mit git pull den neuesten Stand, inklusive des PR, ziehen.
* Wenn man den Branch nicht mehr braucht, kann man ihn löschen (dazu muss man auf einem anderen Branch sein, z. B. master):  
  git branch -d my-branch