

git

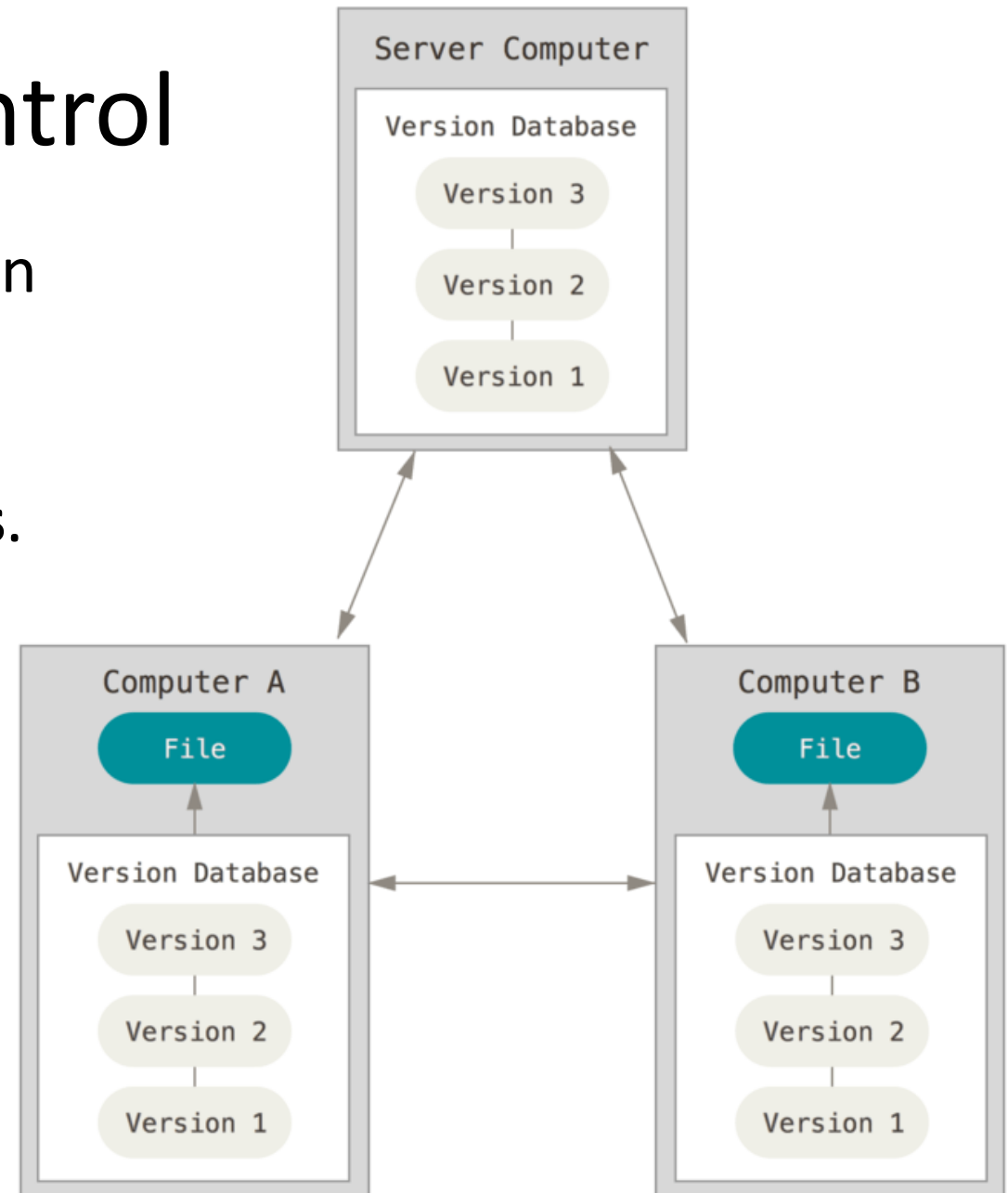
a (very) short introduction

Git

- VCS – Version Control System
 - System zur Versionsverwaltung von Projekten.
 - Möglichkeit zu alten Versionen zurückzuspringen.
- Über Kommandozeile oder (zig!) GUI-Clients bedienbar.
- Initiiert von Linus Torvalds.
- Open Source.

Distributed Version Control

Anwender erhalten nicht einfach den letzten Snapshot des Projektes von einem Server, sondern eine vollständige Kopie des Repositories.



Local repository

**Working
Directory**

**Staging Area
(Index)**

**Repository
(HEAD)**

Remote

`git add`

`git commit`

`git push`

`git checkout`

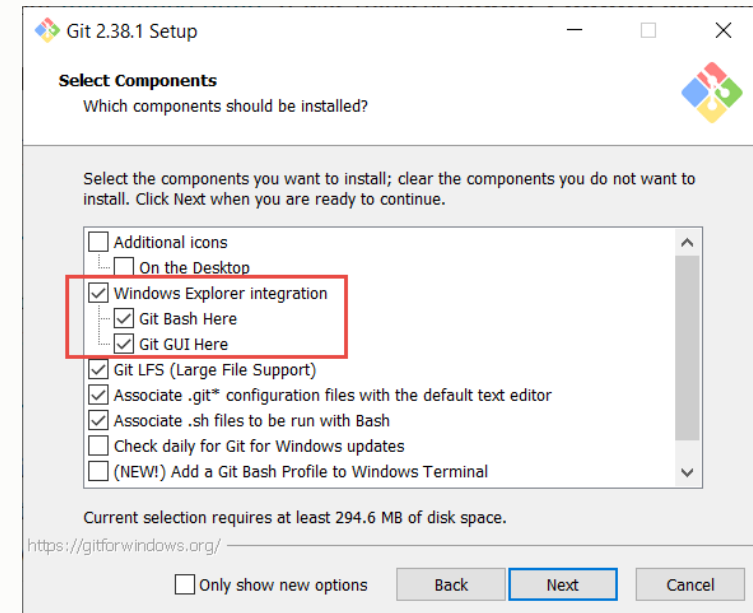
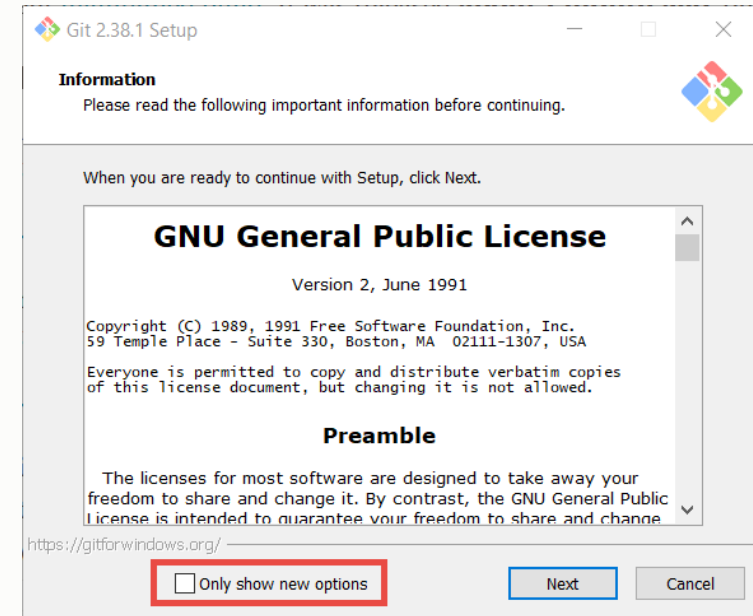
`git pull`



Let's start: Installation

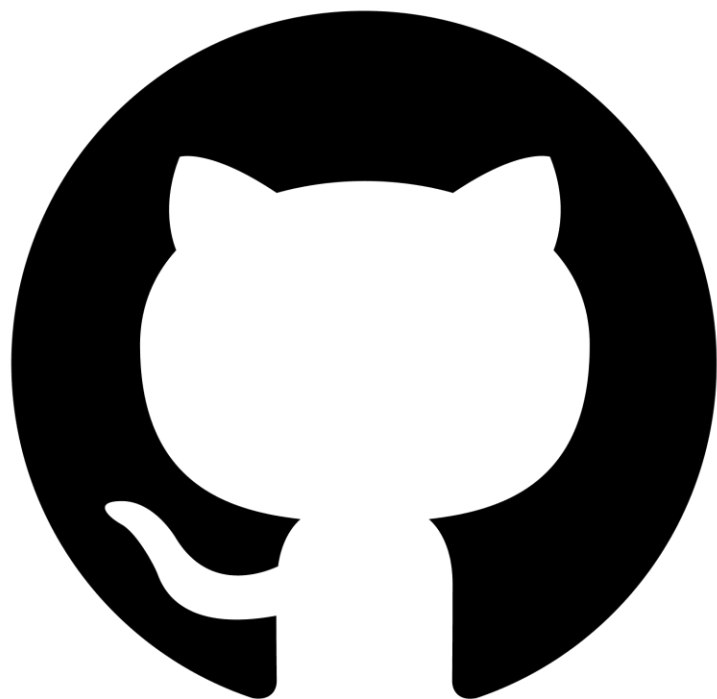


<https://www.git-scm.com/downloads>



Konfiguration

1. Starte die Git Bash
2. Konfiguriere Deinen Namen und Deine E-Mail:
 \$ git config --global user.name "John Doe"
 \$ git config --global user.email john.doe@example.com
3. Deine Einstellungen überprüfen:
 \$ git config --list

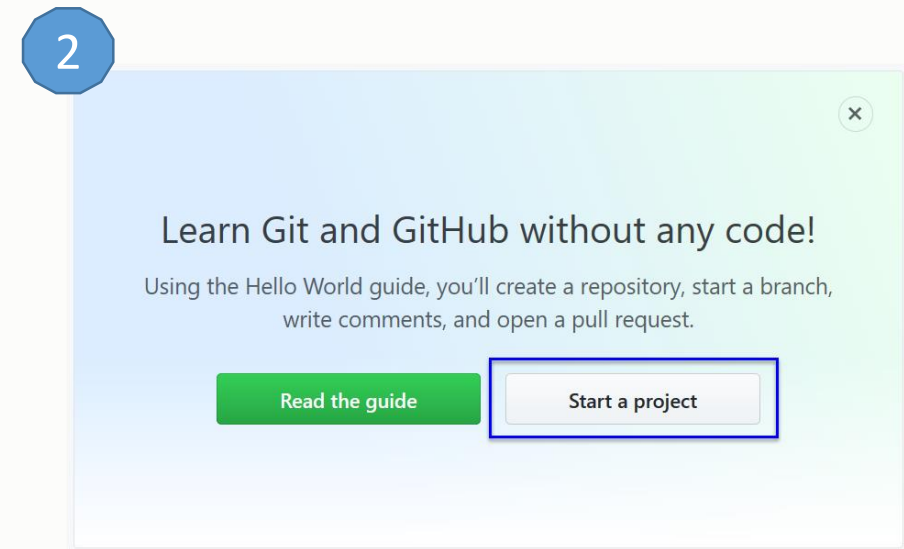
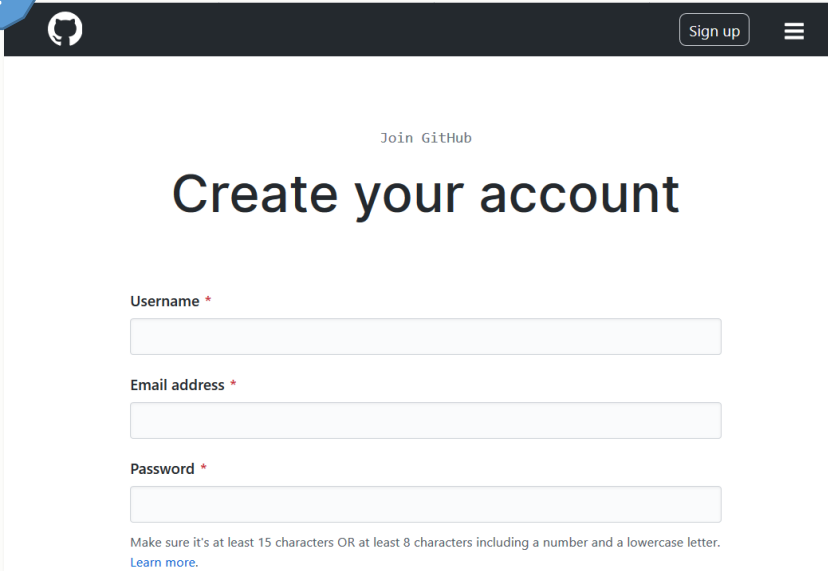


GitHub

Ein Repository auf GitHub anlegen

1. Create your GitHub Account: <https://github.com/join>
2. Create a new repository (Start a project)


1



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

 IctBerufsbildungZentralschweiz ▾

Repository name *

blj-blog ✓

Great repository names are short and memorable. Need inspiration? How about [effective-umbrella?](#)

Description (optional)

Projektarbeit: BLJ-Blog mit php



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.



Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

Add a license: **None** ▾

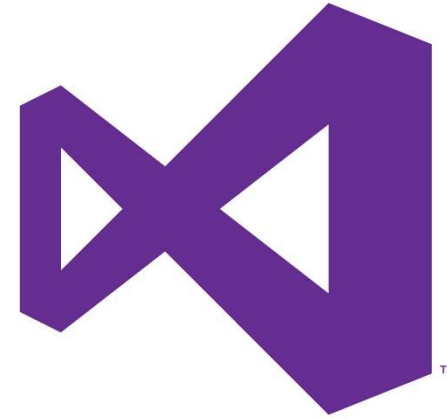


Create repository



git

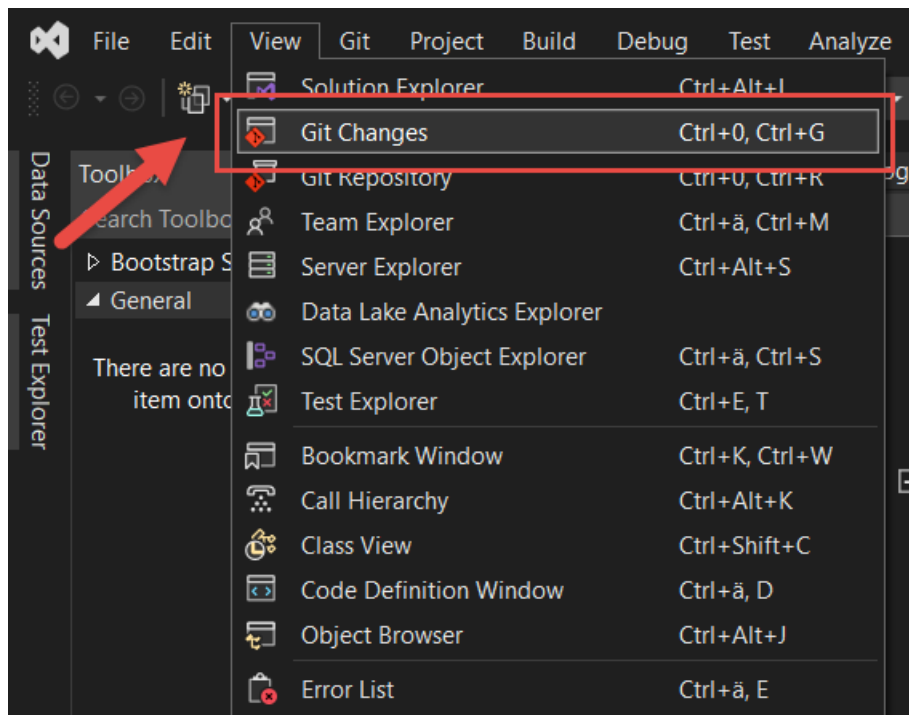
&



Visual StudioTM

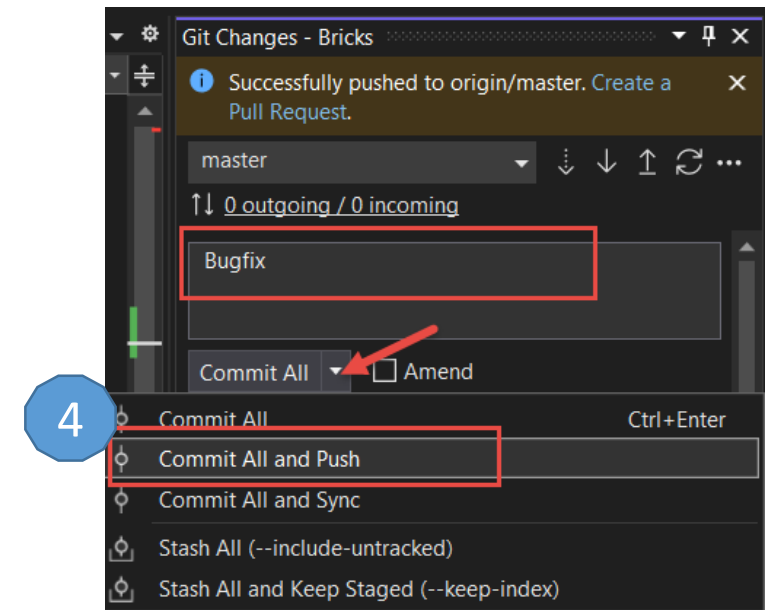
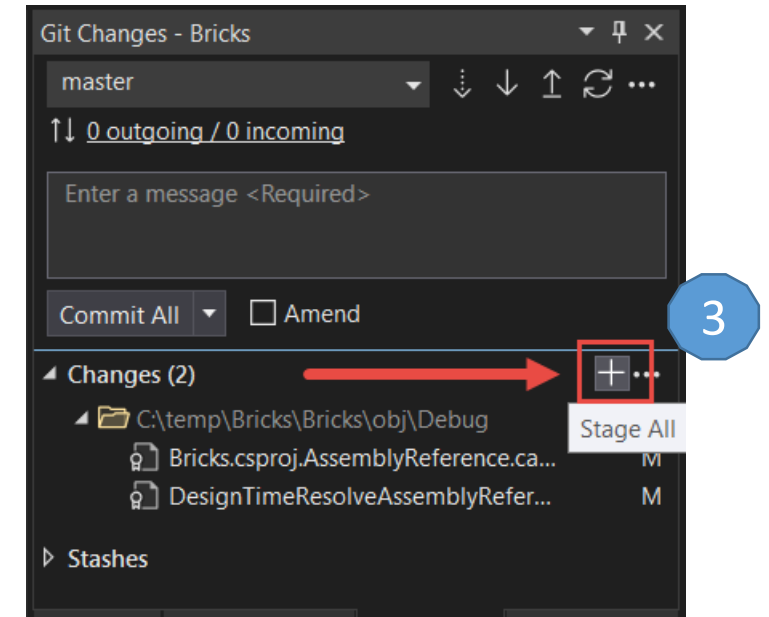
Ein Repository mit VS auf GitHub anlegen

- Menu Ansicht > Git Aenderungen > Git-Repository erstellen



Git Workflow mit VS

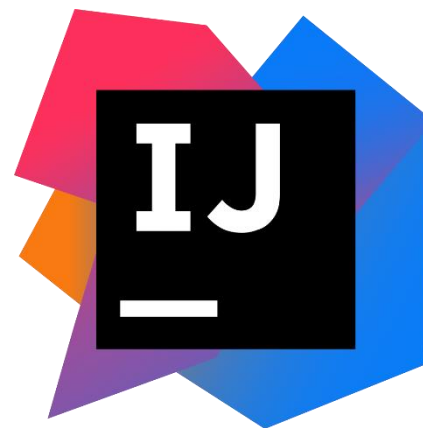
1. Sync
2. Arbeiten (Änderungen machen)
3. Stage All
4. Commit All and Push
oder
Commit All and Sync
(Message nicht vergessen)

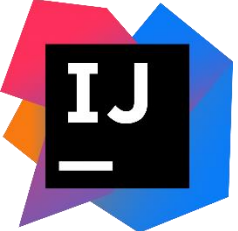




git

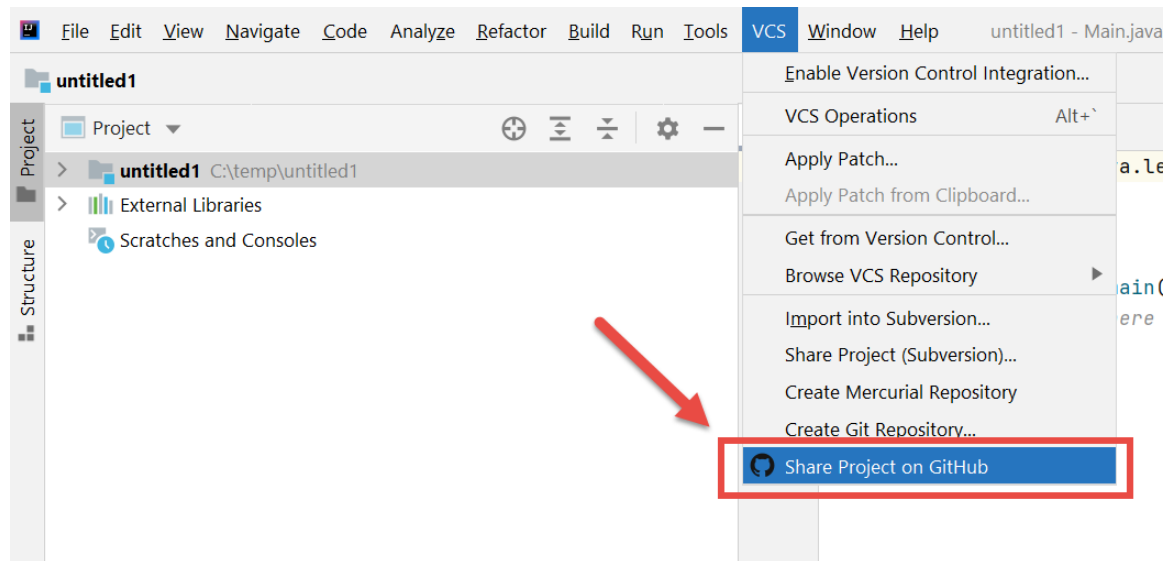
&

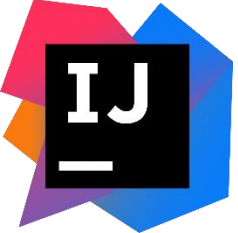




Ein Repository mit IntelliJ auf GitHub anlegen

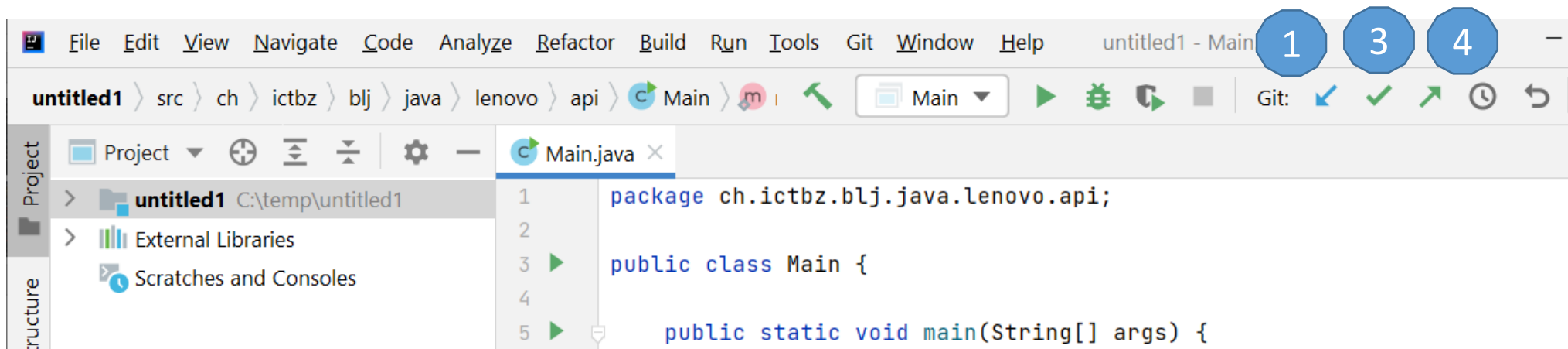
- Menu VCS > Share Project on GitHub





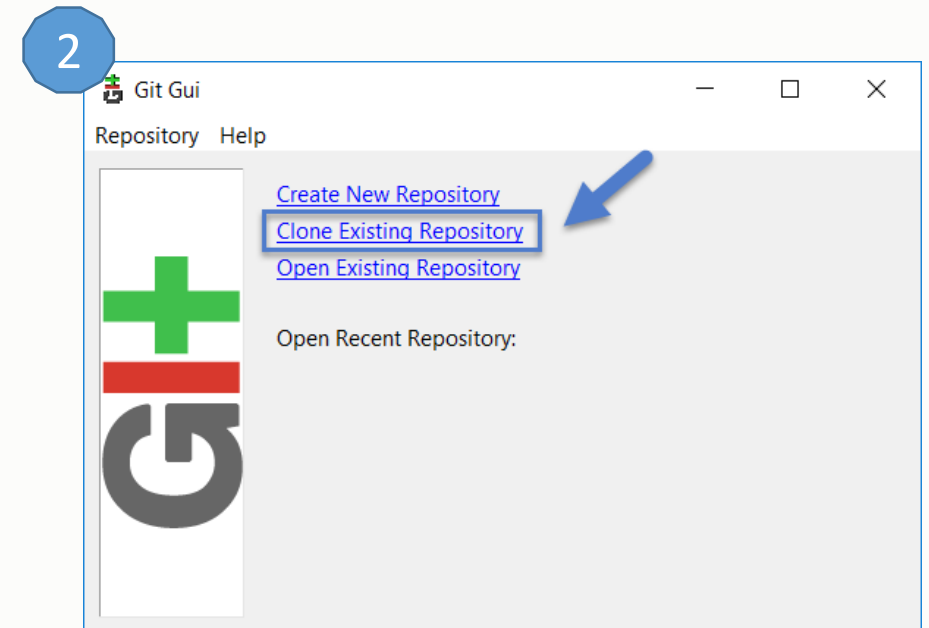
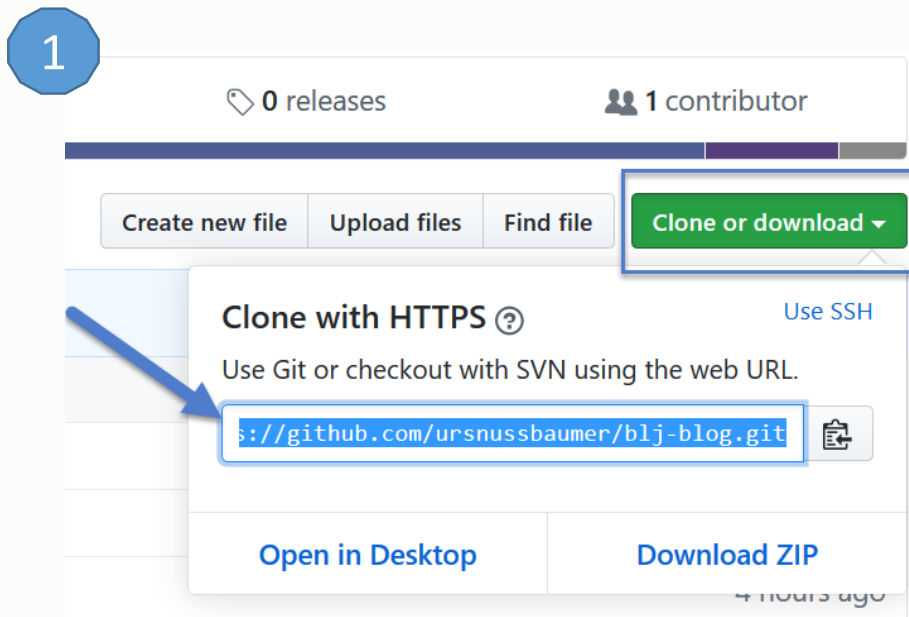
Git Workflow mit IntelliJ

1. Update Project (Merge)
2. Arbeiten (Änderungen machen)
3. Commit
4. Push

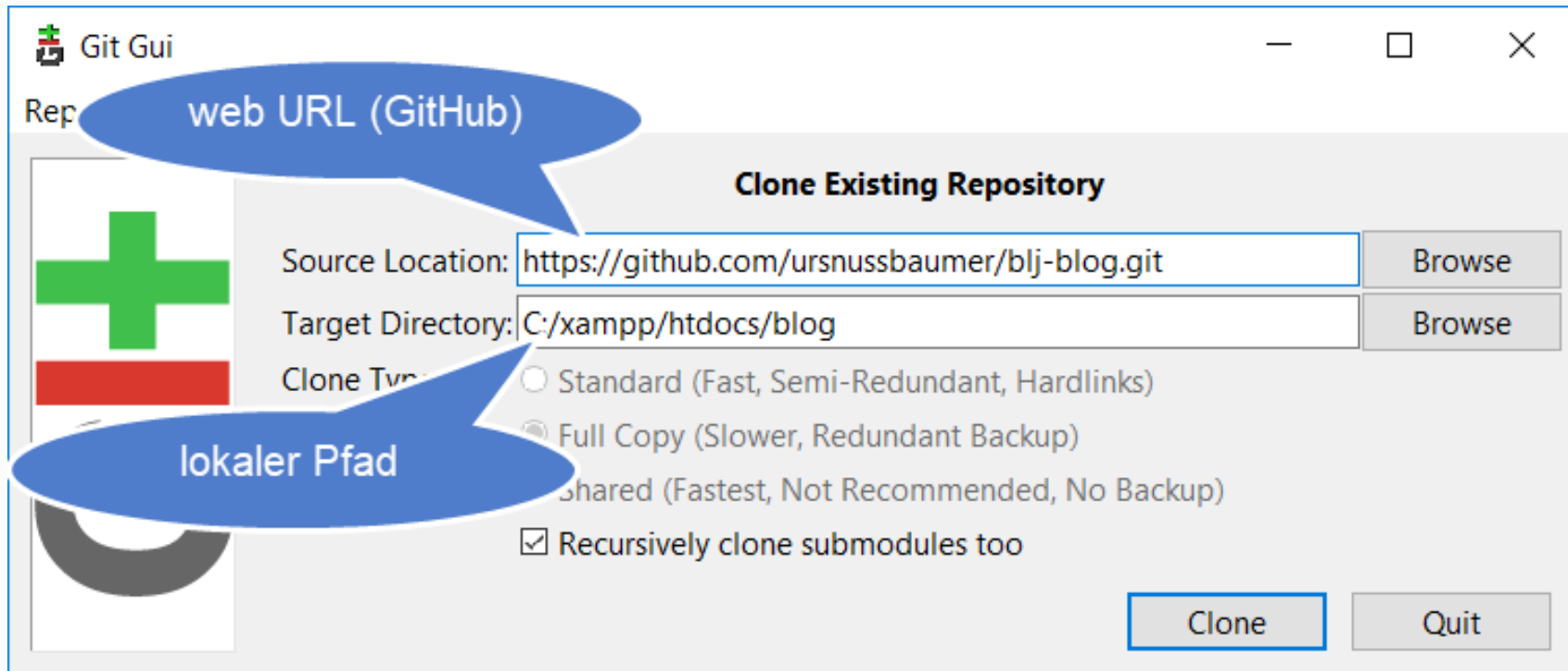


Git Gui: Ein Repository klonen (1)

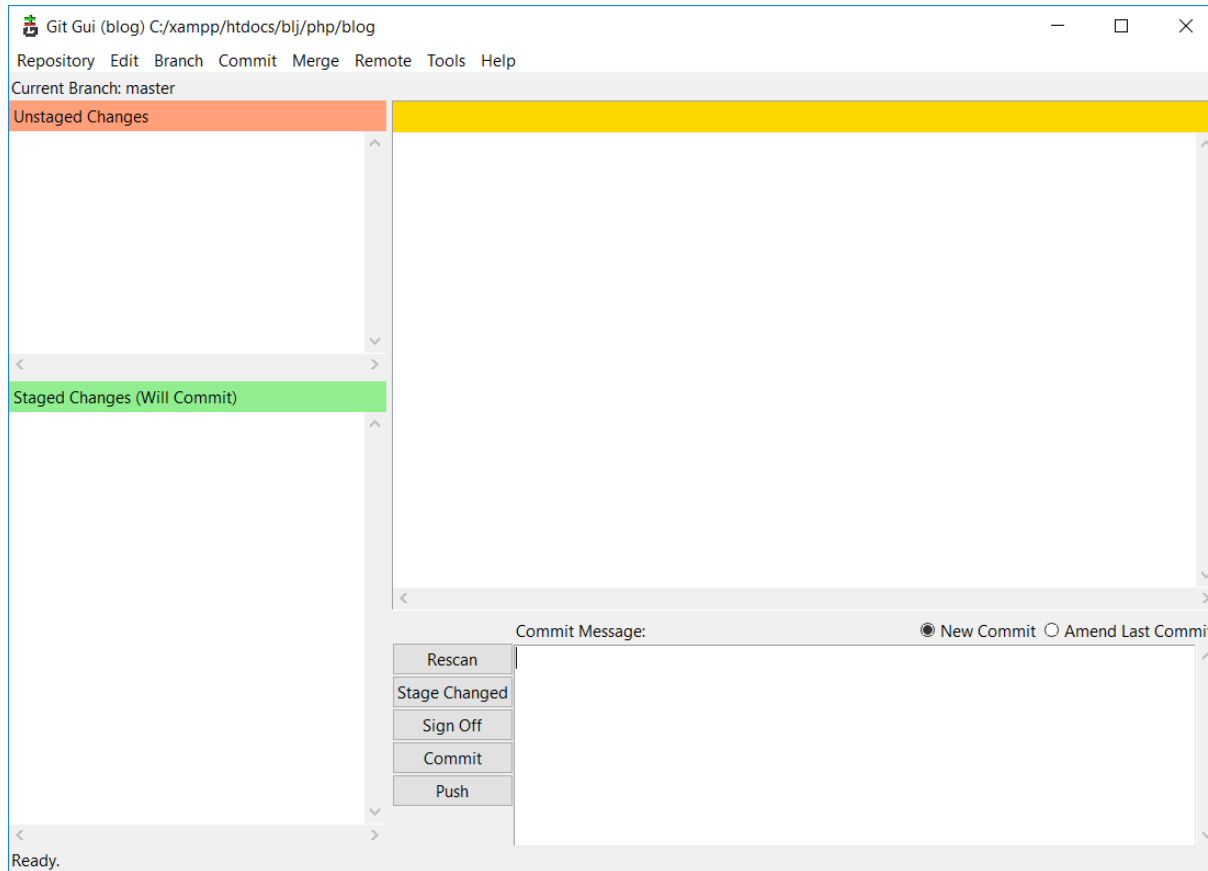
1. GitHub: Copy web URL
2. Git Gui: Clone existing Repository



Git Gui: Ein Repository klonen (2)



Git Gui: Ein Repository klonen (3)



Fertig.

Git Bash: 2 Möglichkeiten, ein Git-Repository auf dem Rechner anzulegen

1 Ein existierendes Verzeichnis als Git-Repository initialisieren

- Importieren (kopieren oder verschieben) eines existierenden Projekts/Verzeichnisses in ein neu initialisiertes Git Repository.

2 Ein existierendes Git-Repository klonen

- Klonen eines existierenden Repositories von einem anderen Rechner, der als Server fungiert.

Git Bash: Ein existierendes Verzeichnis als Git Repository initialisieren

1. Um das Verzeichnis zu initialisieren: in das betr. Verzeichnis wechseln und folgenden Befehl ausführen:

```
$ git init
```

2. Dateien hinzufügen und commiten:

```
$ git add *.*
```

```
$ git commit -m
```

3. Ein (! nicht initialisiertes !) Remote-Repository hinzufügen:

```
$ git remote add origin https://github.com/ursnussbaumer/test.git
```

4. Dateien pushen:

```
$ git push -u origin main
```

Git Bash: Ein existierendes Git-Repository klonen

1. Um das Verzeichnis zu initialisieren in das betreffende Verzeichnis wechseln und folgenden Befehl ausführen:

```
$ git clone [url] [zielverzeichnis]
```

2. Dateien hinzufügen:

```
$ git add *.*
```

```
$ git commit -m
```

Git Bash: Mit externen Repositories arbeiten

Wenn Du wissen willst, welche externen Server Du für Dein Projekt lokal konfiguriert hast:

```
$ git remote -v
```

Änderungen aus externem Repository herunterladen inklusive zusammenführen:

```
$ git pull origin master
```



Es gilt pull vor push, falls lokal etwas nicht aktuell ist.

Änderungen in ein externes Repository hochladen:

```
$ git push origin master
```

Git Bash: Zustände von Dateien

Zustand der Dateien im lokalen Repository prüfen: `$ git status`

Zustände:

- *Untracked files*
to track: `$ git add {Dateiname}`
- *Changes to be committed*
to commit: `$ git commit -m "commit message" {Dateiname}`
- *Changes not staged for commit*
to stage: `$ git add {Dateiname}`

Git Bash: Änderungen durchsehen

Wenn Du wissen willst, was sich geändert hat:

```
$ git diff
```


.gitignore: Dateien ignorieren

.gitignore

- im Root des Projektverzeichnisses
- Leere Zeilen oder Zeilen, die mit # beginnen, werden ignoriert
- Standard `glob` Muster funktionieren *
- Muster mit einem Schrägstrich (/) abschliessen, um ein Verzeichnis zu deklarieren
- Muster negieren, indem ein Ausrufezeichen (!) vorangestellt wird

Glob-Muster

- vereinfachte reguläre Ausdrücke:
 - * = kein oder mehrere Zeichen
 - [abc] = eines der angegebenen Zeichen (a, b oder c)
 - ? = beliebiges einzelnes Zeichen

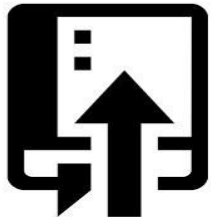
In case of fire



(C) google.com/+StephanSchmitz



1. git commit



2. git push



3. leave building