

- 1. Github Repository erstellen**
- 2. Inhalt ins Repository laden**
- 3. Netlify Account erstellen**
- 4. Netlify mit Github verbinden**
- 5. Site URL anpassen**
- 6. Readme editieren**

## GIT, GITHUB, VISUAL STUDIO CODE


# 1. Github Repository erstellen

Logge dich über den Browser ein in deinen Github Account. Erstelle auf Github ein neues Repository: GridModules. Setze es auf Public.

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

---

**Owner \*** **Repository name \***

 hzuellig / kickstartTest ✓

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

**Description (optional)**

kickstartTest

---

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

---

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

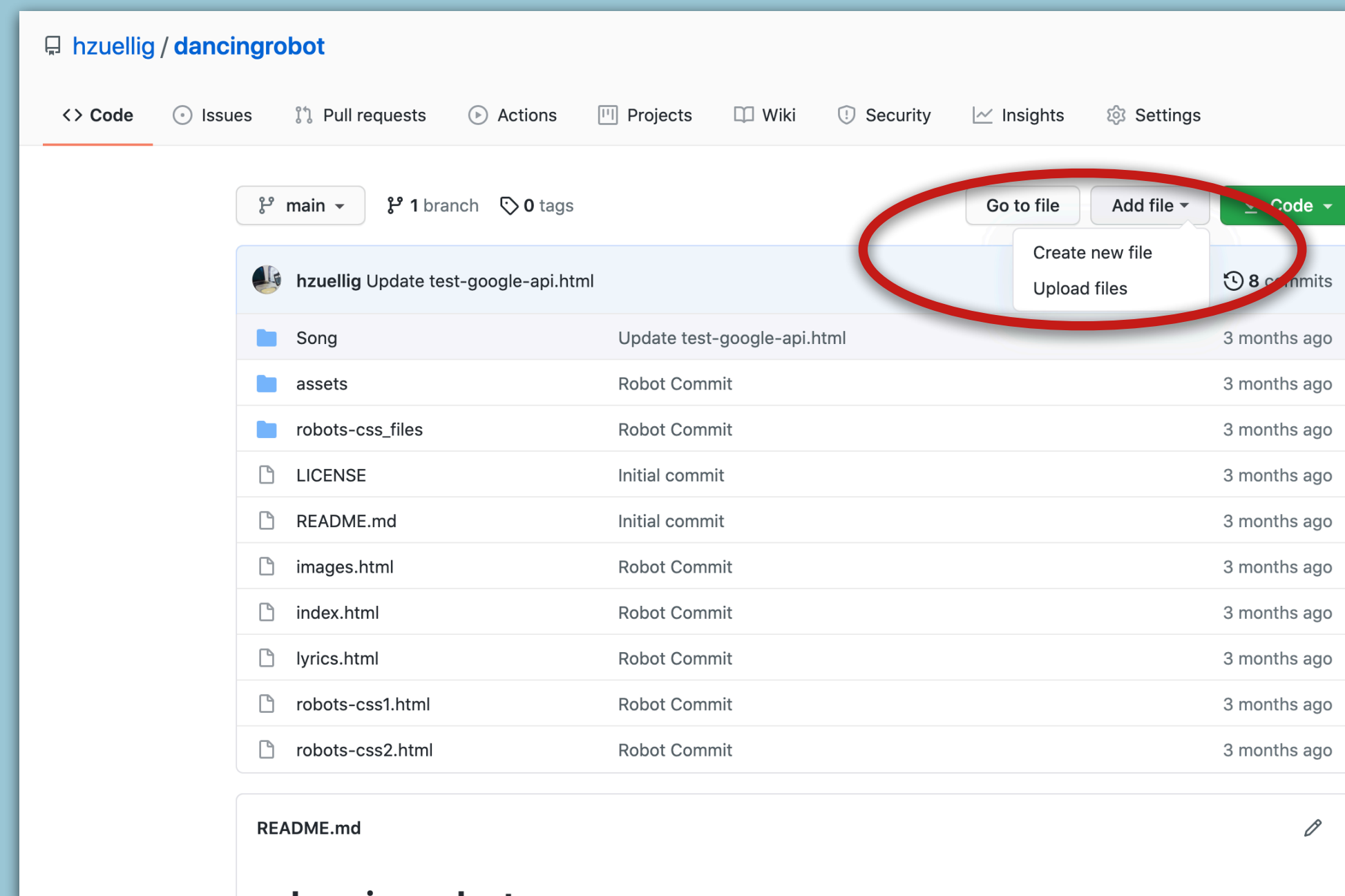
This will set `main` as the default branch. Change the default name in your [settings](#).

---

**Create repository**

GIT, GITHUB, VISUAL STUDIO CODE

## 2. Inhalt ins Repository laden



# 3. Netlify Account erstellen

Über den Dienst von Netlify kannst du dein Github Repo verknüpfen und öffentlich sichtbar machen, wie die Applikation aussieht. Nach jedem push können die Anpassungen präsentiert werden.

Geht auf <https://www.netlify.com/> und loggt euch ein mit euren Github Daten.

## GITHUB UND NETLIFY

# 4. Netlify mit Github verbinden

In der Team Overview Seite klickt ihr auf ‚New Site from Git‘ und wählt Github aus. Netlify greift dann auf eure Repos zu. Wählt euer Repo aus und sagt <Deploy Site>.

**Create a new site**  
From zero to hero, three easy steps to get your site on Netlify.

1. Connect to Git provider   2. Pick a repository   3. Site settings, and deploy!

**Site settings for hzuellig/dancingrobot**  
Get more control over how Netlify builds and deploys your site with these settings.

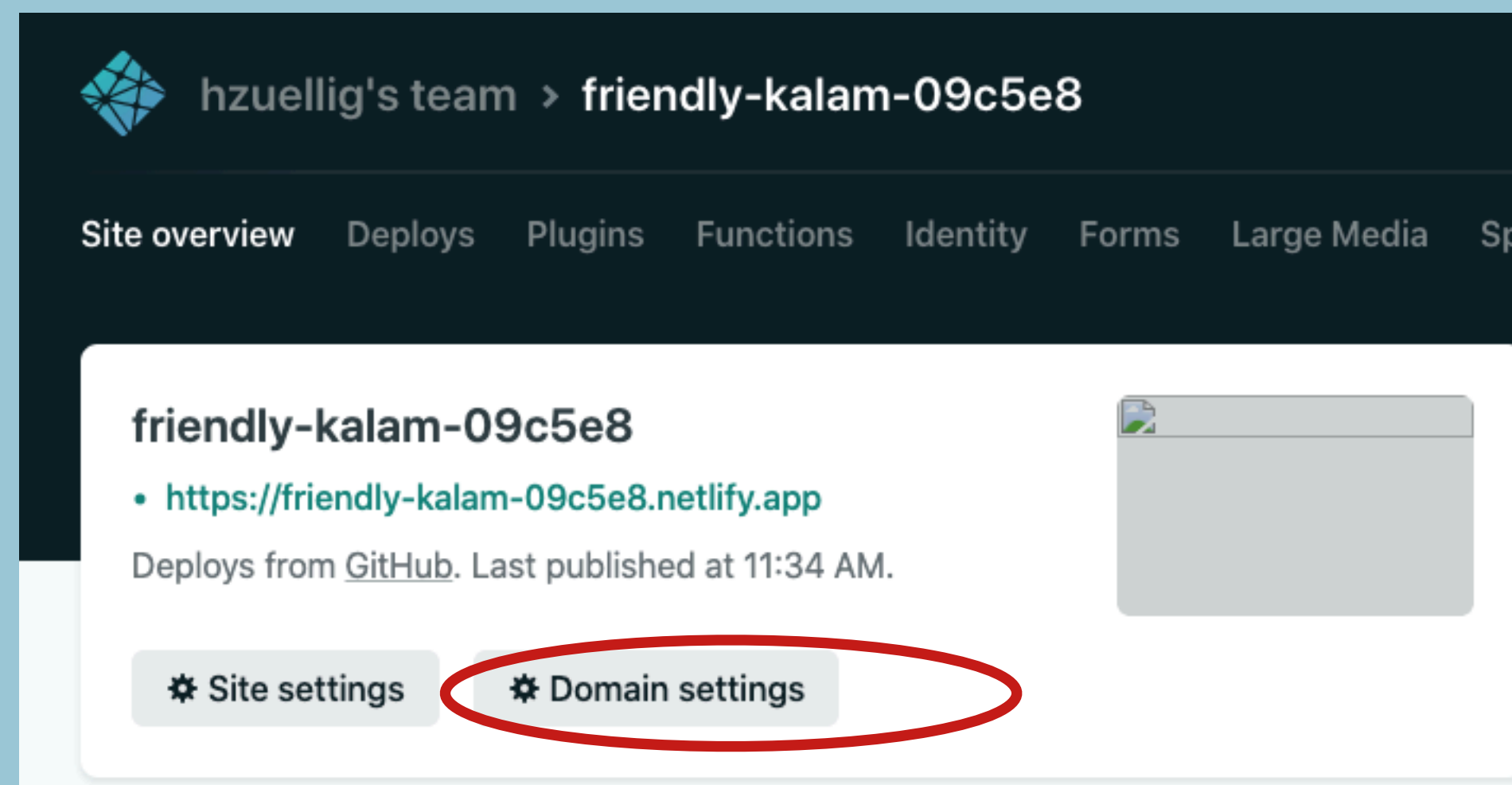
Owner  
hzuellig's team

Branch to deploy  
main

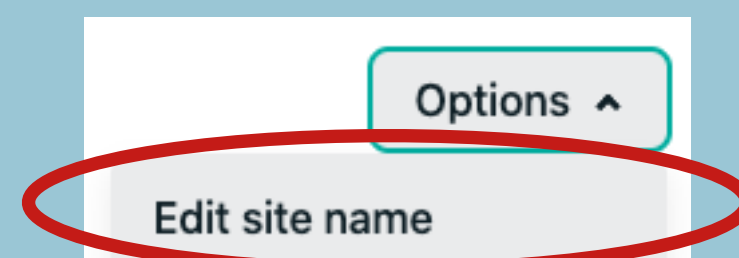
Show advanced

Deploy site

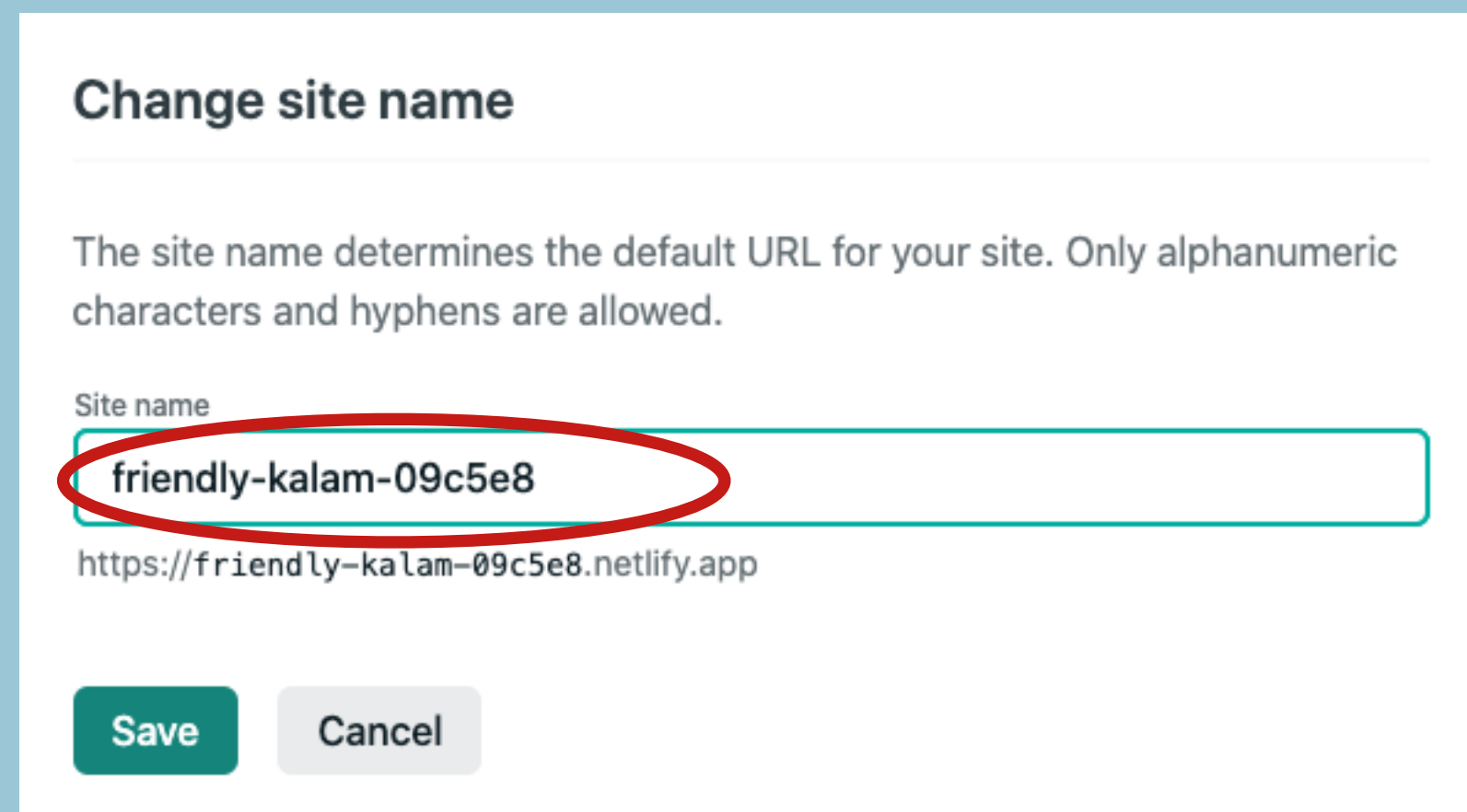
## 5. Site URL anpassen



In der Site overview klickst du auf Domain Settings.  
In der Folgeseite klickst du auf Options und wählst Edit site name.



## 5. Site URL anpassen



**Change site name**

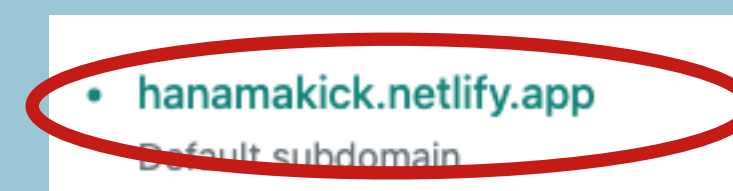
The site name determines the default URL for your site. Only alphanumeric characters and hyphens are allowed.

Site name

friendly-kalam-09c5e8

https://friendly-kalam-09c5e8.netlify.app

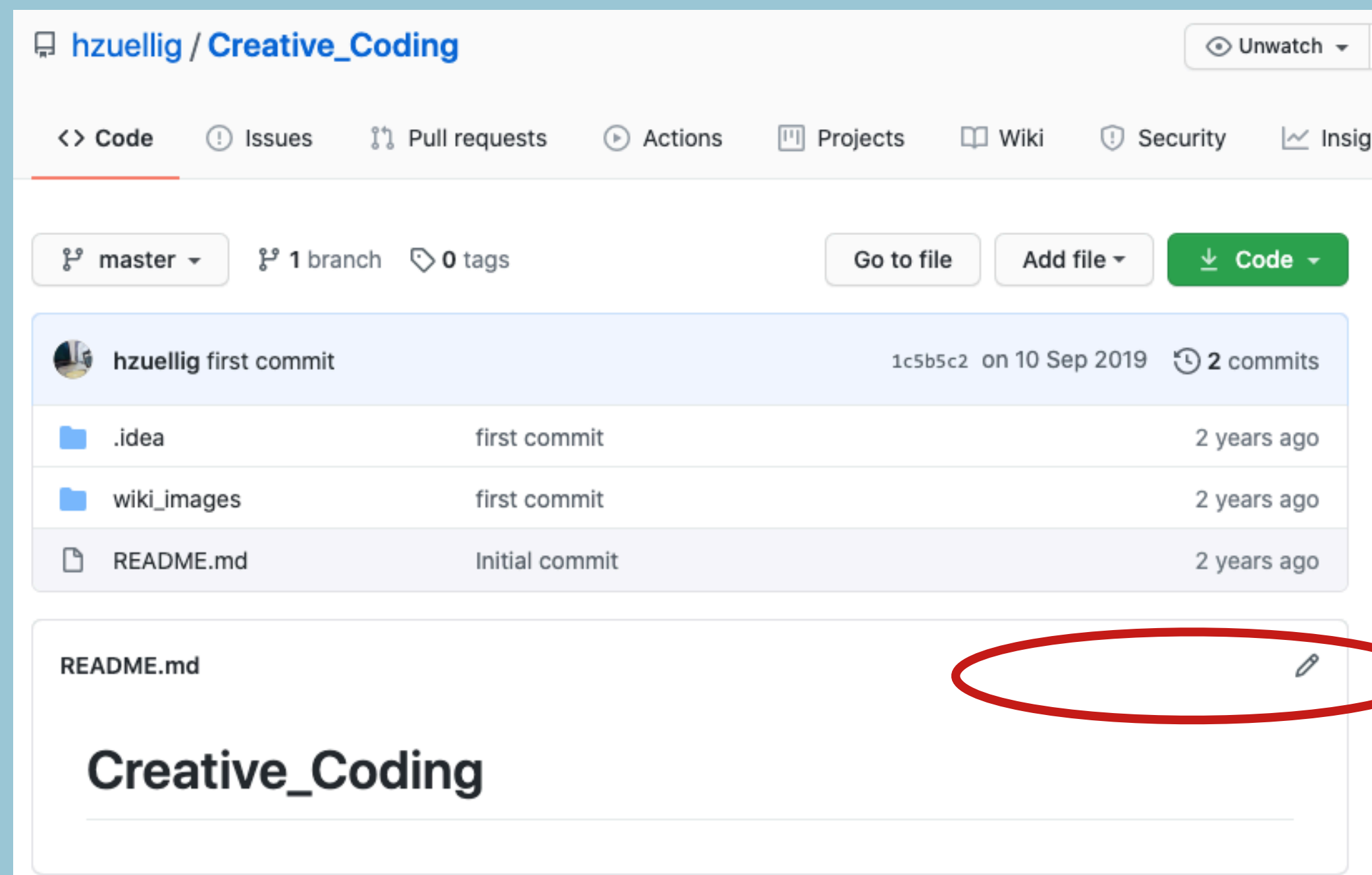
Save Cancel

- 
- hanamakick.netlify.app
- Default subdomain

Hier kannst du einen selbstgewählten Namen eingeben. Über diese URL kannst du deine Arbeiten präsentieren. Jede Änderung, die du auf Github pushst, wird hier automatisch sichtbar.

Bravo :-)

## 6. Readme editieren



Zurück auf Github kannst du das Readme File anpassen und deine Netlify URL einfügen.



- 1. Git installieren**
- 2. Github Repository erstellen**
- 3. Repository lokal klonen**
- 4. Inhalt ins Repository laden**
- 5. Inhalt in die Git Verwaltung aufnehmen**
- 6. Authentifizieren (einmalig) und pushen**
- 7. Commits prüfen**
- 8. Workflow**

GIT, GITHUB, VISUAL STUDIO CODE

# 1. Git installieren

Überprüft zuerst, ob Git bereits auf eurem Rechner installiert ist. Gebt in einem Terminal den Befehl

```
git --version
```

ein. Die Antwort könnte lauten

```
git version 2.24.3 (Apple Git-128)
```

dann ist git installiert.

Falls Git noch nicht installiert ist, kannst du hier einen Installer laden:

<https://sourceforge.net/projects/git-osx-installer/files/>

GIT, GITHUB, VISUAL STUDIO CODE

## 2. Github Repository erstellen

Logge dich über den Browser ein in deinen Github Account. Erstelle auf Github ein neues Repository: GridModules. Setze es auf Public.

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

**Owner \*** **Repository name \***  
hzuellig / kickstartTest ✓

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

**Description (optional)**  
kickstartTest

☐ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☒ **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more](#).

☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more](#).

☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more](#).

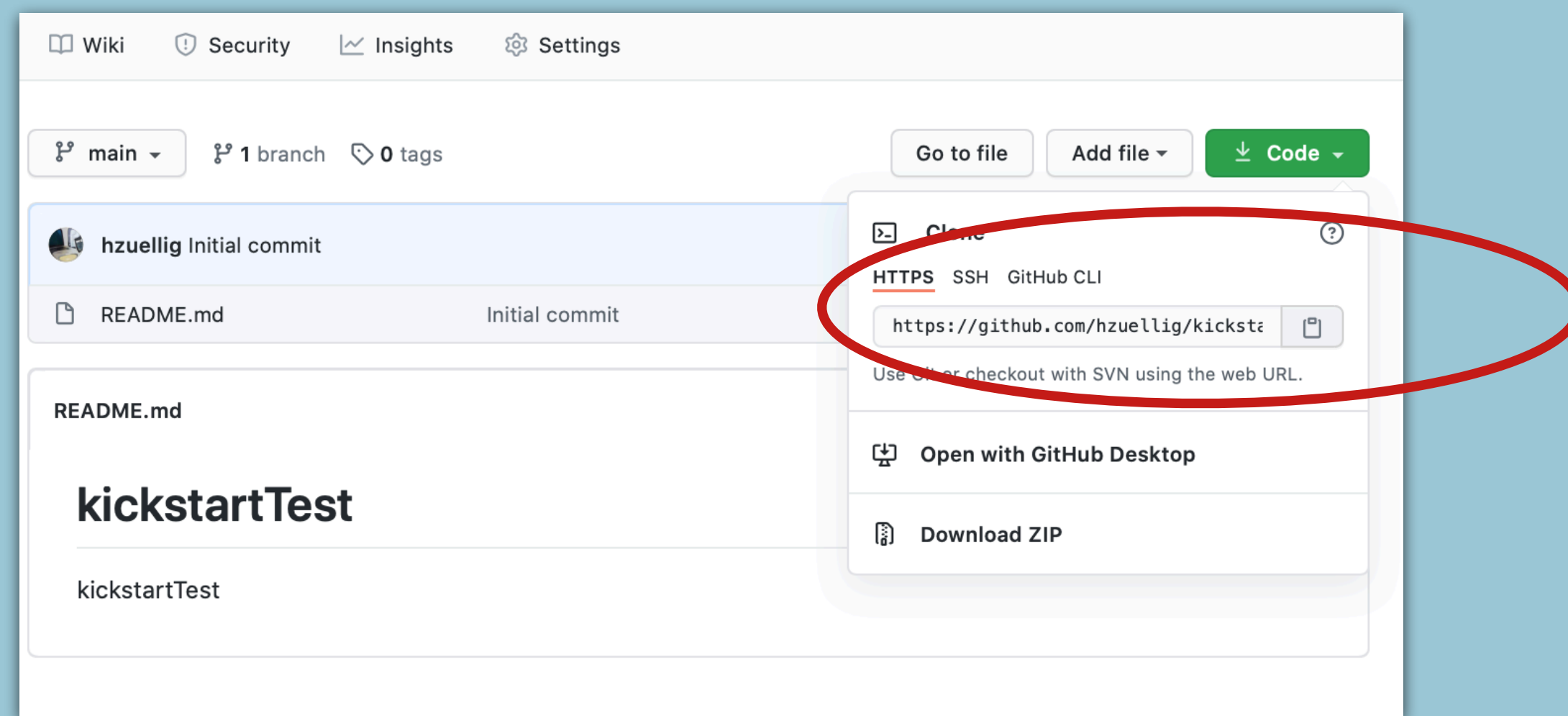
This will set `main` as the default branch. Change the default name in your [settings](#).

**Create repository**

GIT, GITHUB, VISUAL STUDIO CODE

# 3. Klonen

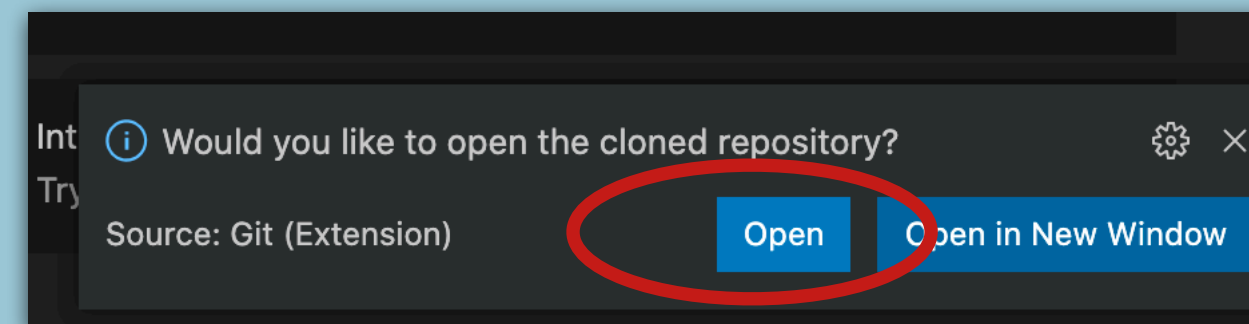
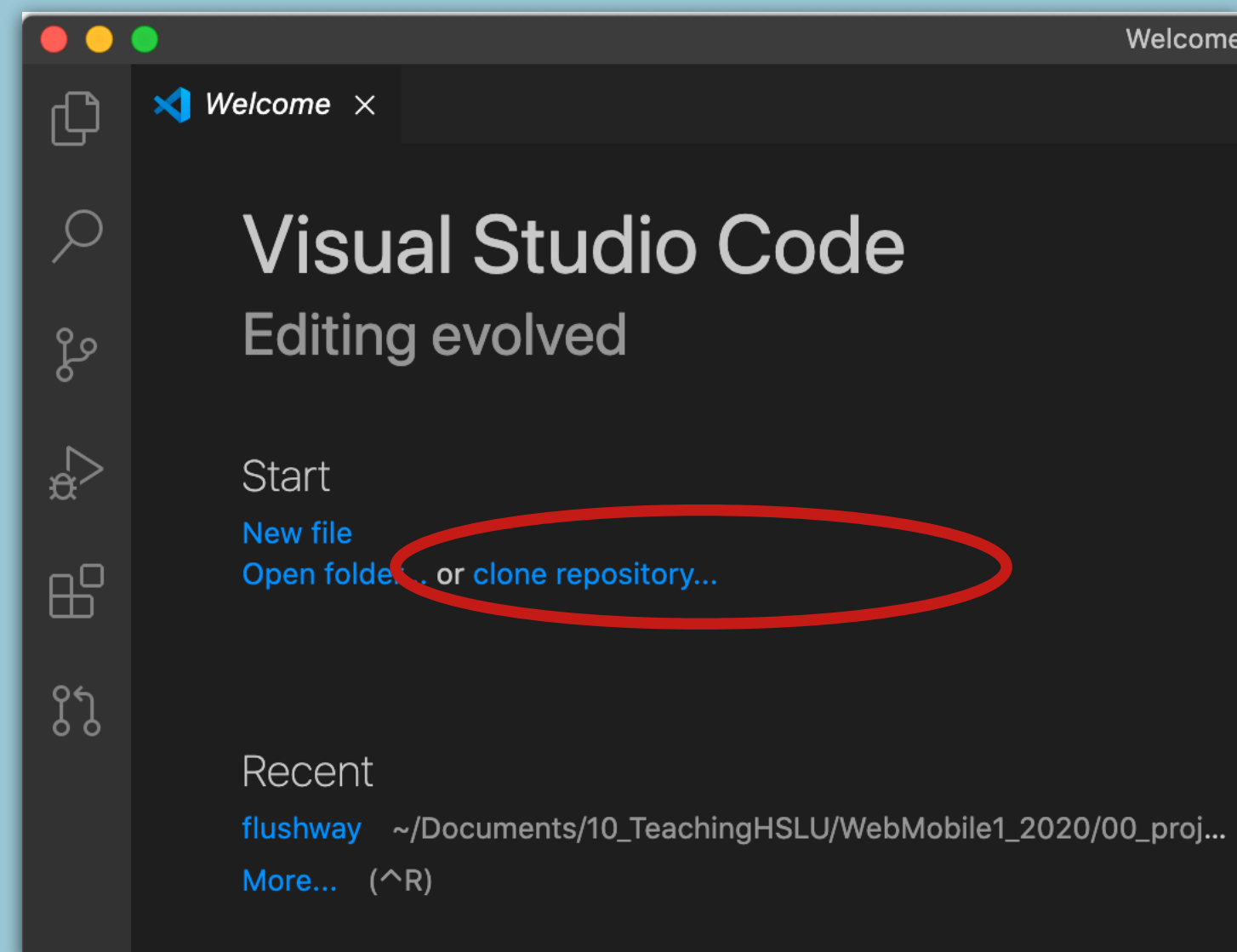
Kopiere die URL neben dem Repo:



GIT, GITHUB, VISUAL STUDIO CODE

# 3. Klonen

Klone das Repo in Visual Studio Code und öffne es.  
Das Repo ist noch leer!



GIT, GITHUB, VISUAL STUDIO CODE

## 4. Inhalt erstellen

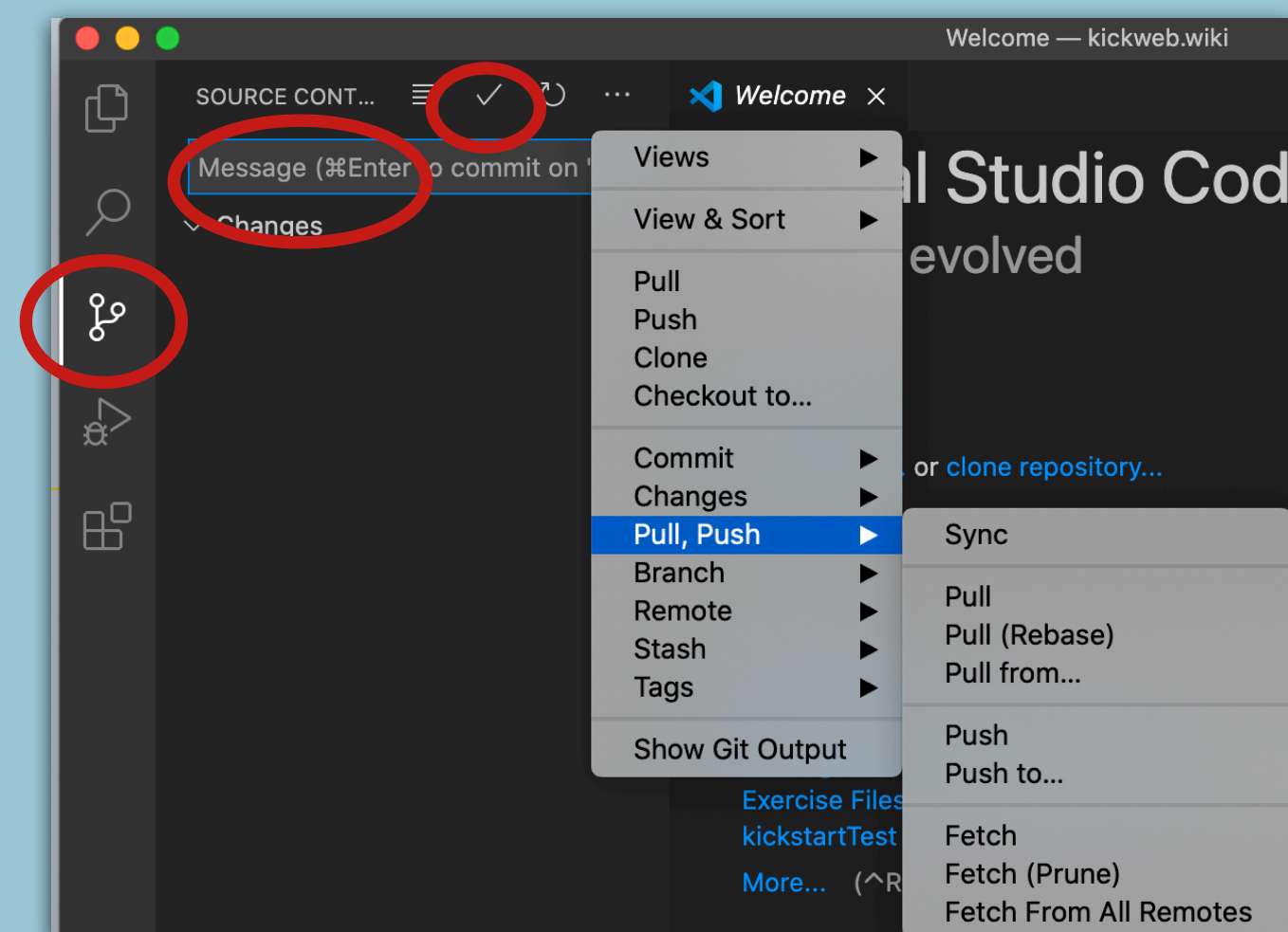
Du kannst über den Finder ein bestehendes Projekt in das neue geklonte Repository ziehen – dieses ist bereits mit Git initialisiert.

Oder du kannst im leeren Repository ein neues Projekt aufbauen.

GIT, GITHUB, VISUAL STUDIO CODE

# 5. Inhalt in die Git Verwaltung aufnehmen

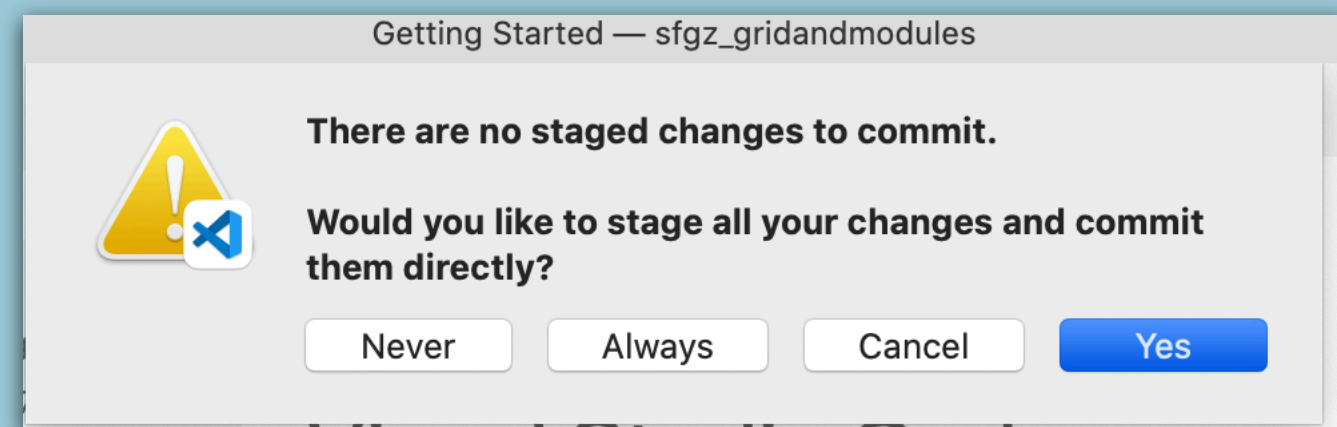
Dann kannst du die neuen Dateien in die Versionsverwaltung einspielen. Wenn du bei Visual Studio Code commit wählst, übernimmt das Programm das. Gib eine Message ein, z.Bsp. ‚mein erster push‘. Mit dem Häkchen bestätigst du den commit.



GIT, GITHUB, VISUAL STUDIO CODE

# 5. Inhalt in die Git Verwaltung aufnehmen

Visual Studio Code fragt dich dann, ob neu hinzugefügte Dateien jeweils automatisch in die Versionsverwaltung aufgenommen werden sollen. Du kannst das bestätigen.

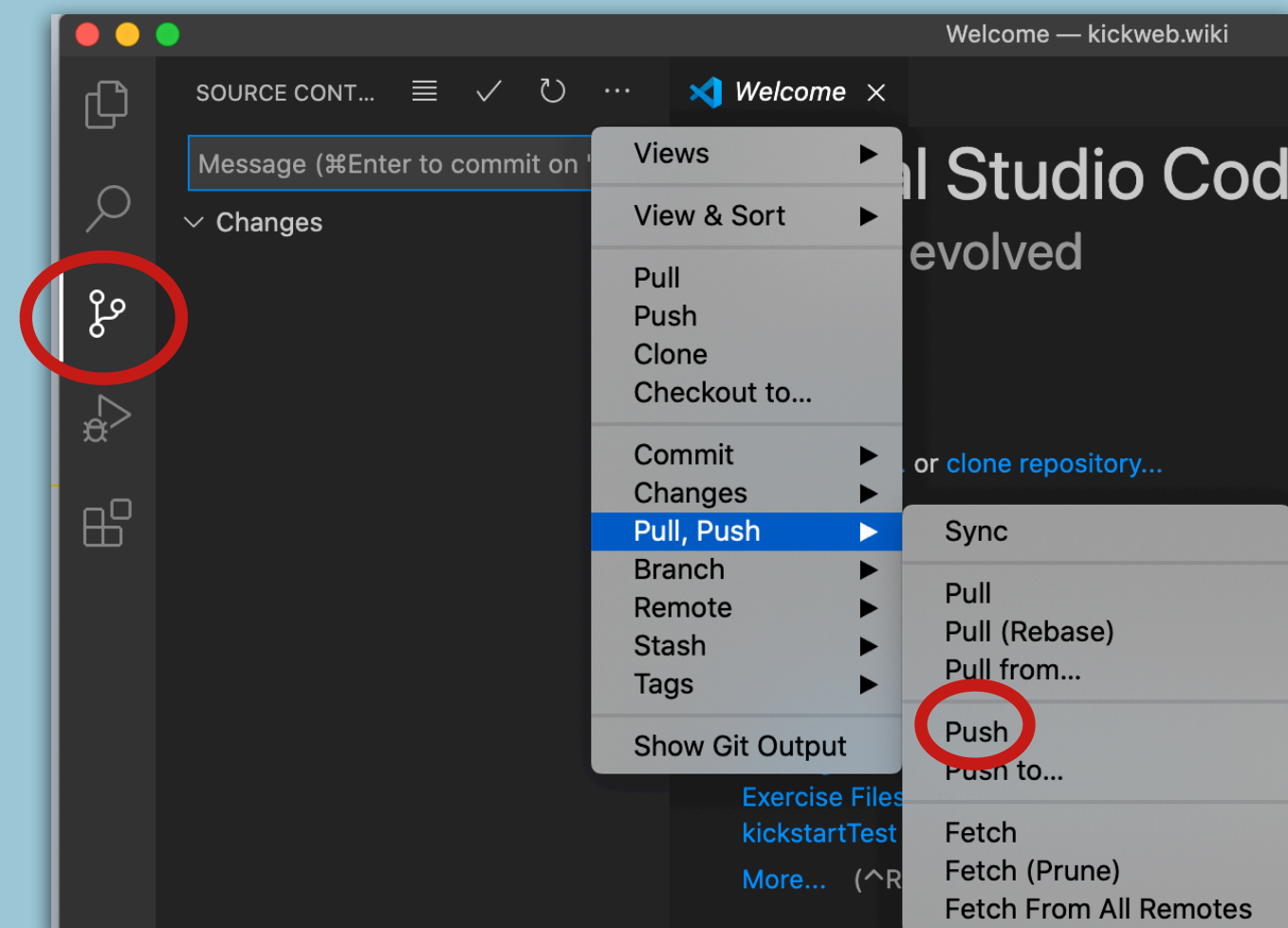




GIT, GITHUB, VISUAL STUDIO CODE

## 6. Authentifizieren

Nun bist du bereit, deinen ersten Push auf den Github Server zu machen. Das machst du über Das Menü:



GIT, GITHUB, VISUAL STUDIO CODE

## 6. Authentifizieren (einmalig) und pushen

Weil du das erste Mal Git benutzt, musst du dich authentifizieren und ins Terminal deinen Github Namen und dein Github E-Mail eingeben. **Die Angaben in Anführungszeichen mit deinen Github Angaben ersetzen!**

Über das Menü Terminal kannst du im Visual Studio Code ein Terminalfenster öffnen.

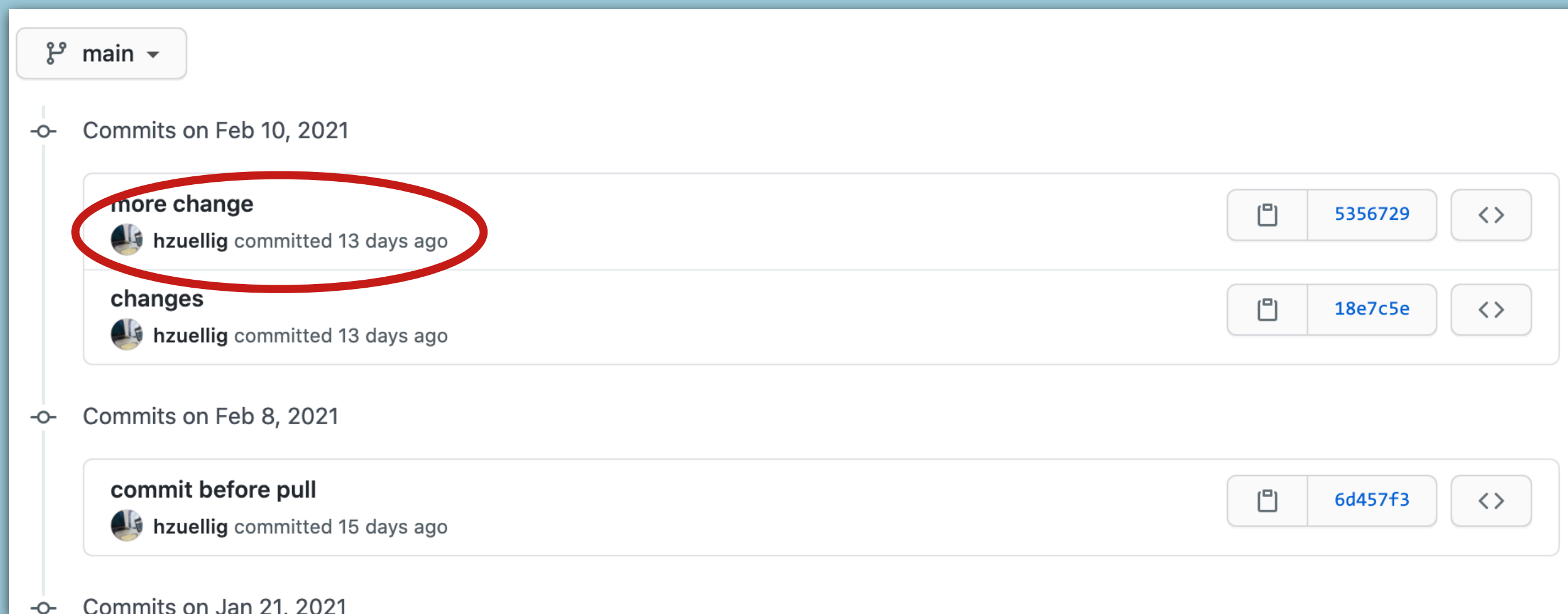
Nach der Eingabe drückst du Enter. Du wirst nun auf ein Browserfenster geführt, auf dem du den Zugriff von Visual Studio Code auf das Github Repo bestätigst. Nachher kannst du den Push fertig machen.

```
$ git config --global user.name "John Doe"  
$ git config --global user.email "johndoe@example.com"
```

GIT, GITHUB, VISUAL STUDIO CODE

# 7. Commits prüfen

Nach dem commit kannst du die Dateien über push auf den Github Server laden. Über commits kannst du auf Github prüfen, ob den push erfolgreich war.



GIT, GITHUB, VISUAL STUDIO CODE

## 8. Workflow

Wenn du mir anderen arbeitest, solltest du dir folgenden Workflow angewöhnen:

```
add / commit  
pull  
push
```

Mit pull holst du dir den aktuellen Stand vom Server.  
So verhinderst du merge Konflikte mit Anpassungen von anderen.