ToEnder

Todo / Kalender Web

Schneeberger Patrick

Inhaltsverzeichnis

[1.0 Projektbeschreibung 3](#_Toc56668819)

[1.1 Vorwort 3](#_Toc56668820)

[1.2 Mein Projekt 3](#_Toc56668821)

[2.0 Teschnische Notizen 3](#_Toc56668822)

[2.1 Speicherung der Todos 3](#_Toc56668823)

[3.0 Planung 3](#_Toc56668824)

[3.1 Zeitplanung 3](#_Toc56668825)

[3.2 User Stories 3](#_Toc56668826)

[4.0 Mockup 4](#_Toc56668827)

1. Projektbeschreibung
   1. Vorwort

Im Modul-150 haben wir den Auftrag bekommen eine Projekt umzusetzen das wie eine praktische Arbeit im Betrieb gewertet wird. Dies Arbeit dient zur Vorbereitung auf unsere praktische Arbeit im Betrieb.

* 1. Mein Projekt

Ich habe mich entschieden eine Webbasierte ToDo Applikation zu erstellen. Die ToDo's sollen einem Datum zugewiesen werden können. Dazu sollen die nachher übersichtlich angezeigt werden.

Github: <https://github.com/sInt3g3r/ToEnder>

1. Teschnische Notizen
   1. Speicherung der Todos

* Als JSON
* Status wird im JSON gespeichert.
* Deno verwenden

|  |  |
| --- | --- |
| Was | Beschreibung |
| Continious Integration |  |
| Continious Deployment |  |
| Node.js |  |
| Jest |  |
| Heroku |  |
| npm |  |
| Docker |  |

1. Planung
   1. Zeitplanung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Datum | Arbeit | Geschätzt | Ist | Bemerkung |
| 19.11.2020 | Dokumentation vorbereiten,  Planung, Beginn mit dem Projekt, Mokup erstellen, User-Stories erstellen | 3h |  |  |
| 26.11.2020 | ToDo erfassen fertig | 3h |  |  |
| 03.12.2020 | ToDo bearbeiten, löschen, Status fertig | 3h |  |  |
| 10.12.2020 | ToDo's einem Datum zuweisen und anzeigen | 3h |  |  |
| 17.12.2020 | Projekt beenden. Beginn mit dem Manuellen Test | 3h |  |  |
|  | Testen des Projekts. Bugfixing | 3h |  |  |
|  |  |  |  |  |

* 1. Arbeitsjournal

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Arbeit | Dauer |  |
| 19.11.2020 | Continuous Integrateion  Continuous Deployment grob kennen gelernt.  Angefangen dieses einzurichten. | 3h |  |
| 26.11.2020 | Continuous Deployment fertig eingerichtet.  Beginn mit der Webseite | 3h |  |
| 03.12.2020 | add.html erstellt für das Erfassen von ToDo's | 3h |  |
| 10.12.2020 | add.html funktion fertig in Local Storage. Mit der Darstellung des Kalenders begonnen | 3h |  |
| 17.12.2020 | Leere Cards werden in der Monatsansicht dargestellt  -> Mehrere Tasks am selben Tag ? |  |  |
| 21.12.2020 | Mehrere Tasks am selben Tag darstellen mit ID, nicht fertig | 1.5h |  |
| 23.12.2020 | Mehrere Tasks am selben Tag darstellen mit ID, Fertig  Löschen eines Task funktioniert.  Umbennenen von Card / Cards zu task / tasks | 1.5h |  |
| 26.12.2020 | Editieren Funktion angefangen | 45min |  |
| 01.01.2021 | Editieren Funktion beendet | 4h |  |
| 02.01.2021 | Dialoge beim löschen hinzugefügt. Aktualisieren des Monats nach dem löschen hinzugefügt. Style änderungen an Buttons vorgenommen | 3h |  |

* 1. User Stories

Beispiel: **"Als [Kundentyp] [möchte] ich, [damit]."**

*Als Max möchte ich meine Freunde einladen, damit wir diesen Service gemeinsam nutzen können.*

*Als Sascha möchte ich meine Arbeit organisieren, damit ich mehr Kontrolle darüber habe.*

Die User Stories werden im Github repository als Issues erfasst. Im Projekt werden danach die User Stories abgearbeitet.

<https://github.com/sInt3g3r/ToEnder>

1. Continious Integration
   1. npn
   2. Node.js

1) Windows Installer LTS

* 1. Heroku.com

Project: toender

Download and install the [Heroku CLI](https://devcenter.heroku.com/articles/heroku-command-line).

$ heroku login -i

$ cd my-project/

$ git init

$ heroku git:remote -a toender

$ git add .

$ git commit -am "make it better"

$ git push heroku master

1. Continious Deployment
   1. Github

1) Projekt Repository erstellen -> Code pushen

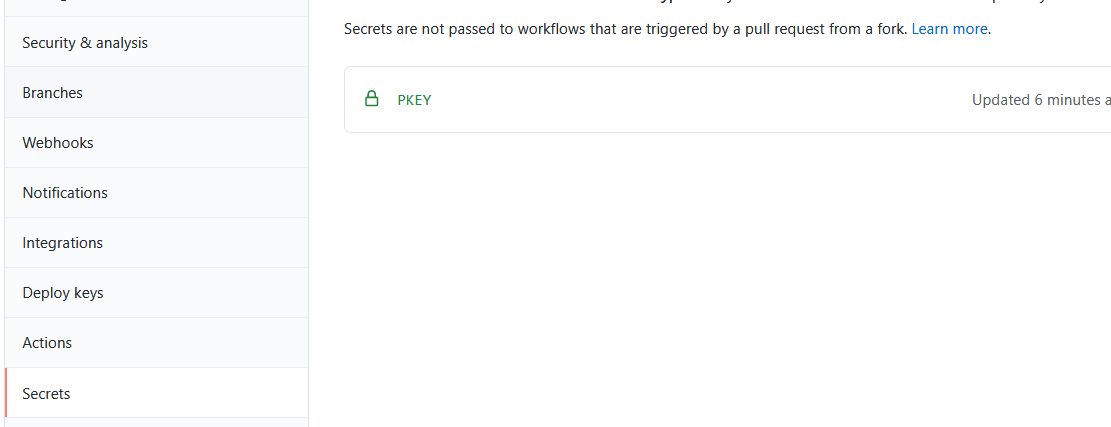
2) Deployment Repository erstellen  
 **Achtung beim Repository: "<username>.github.io"**

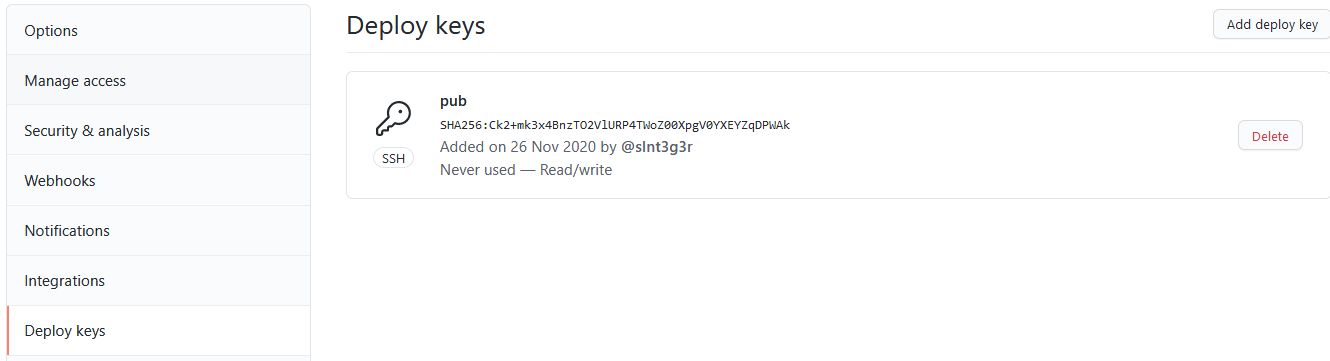
muss verwendet werden sonst funktioniert es nicht!

z.b sint3g3r.github.io

3) mit cmd "ssh-keygen -t rsa" neuen public/private key erstellen

4) Im Projekt Repository unter Settings->Secrets den Private key eintragen.



5) Im Deployment Repository unter Settings->Deploy keys den Public key eintragen. 

* 1. Deploy.yml

name: CI only

on:

  push:

    branches:

      - main

jobs:

  deploy-test-pages:

    runs-on: ubuntu-latest

    steps:

    - uses: actions/checkout@v2

    - name: Deploy

      uses: peaceiris/actions-gh-pages@v3

      with:

        deploy\_key: ${{ secrets.PKEY }}

        external\_repository: sInt3g3r/sInt3g3r.github.io

        publish\_branch: main

        publish\_dir: ./frontend/src

<https://github.com/actions/checkout>

<https://github.com/peaceiris/actions-gh-pages>

1. Mockup

