Erkenntnisse



|  |  |
| --- | --- |
| Autor: | Alexander Schmid |
| Datum: | 19.11.2021 |
| Klasse: | API20 |
| Abgabetermin: |  |
| Dozent: | Fabian Dennler |

Inhaltsverzeichnis

[1 Einleitung 3](#_Toc74247191)

# Einleitung

Wir müssen mittels einer agilen Methode eine Software entwickeln. Das Ziel dabei ist eine funktionsfähige Software zu haben die mittels einer API funktioniert. Wir haben uns entschieden, eine Software zu machen, wo alle Followers/Freunde etc. zusammenzählt und aufzeigt.

# Tag 1

Der erste Tag haben wir mehr oder weniger drauf los Programmiert nachdem alles besprochen wurde. Ich habe das Gefühl, dass wir eigentlich relativ wenig Aufwand haben werden, da wir Python verwenden. Ich habe ¾ vom tag an der API von Facebook/Instagram verwendet. Diese ist nur über einen Authentifizierungstoken der Software verfügbar, leider funktioniert dies nicht so, wie es Dokumentiert ist.

Nach langer suche und ausprobieren mit Herr Dennler selber habe ich einen weg bei Instagram gefunden, durch eine Hintertür diese Informationen raus zu lesen. Leider ist es so gesagt nicht wirklich eine API die dabei verwendet wird, jedoch kommen die Daten trotzdem in JSON.

Gleichzeitig haben wir uns entschieden, dass wir eine eigene API erstellen, mit den Informationen die wir fürs anzeigen der Software brauchen. Dafür hat uns Herr Dennler <https://replit.com/> gezeigt, wo wir so gesagt wie als Server verwenden kann.

## Erkenntnisse

Ich habe gesehen, dass mittels Python sich sehr vieles auch sehr einfach machen lässt… Jedoch eignet sich nicht alles um mit Python zu machen, da im Frontend und Serverseitig nicht so geeignet ist wie PHP zum Beispiel.

# Tag 2

Wir mussten uns heute eingestehen, dass wir mit Python an sich nicht genug weit kommen würden und ebenfalls sehr viele verschiedene Programmiersprachen verwenden müssten. Dazu überredet Node zu verwenden und schlussendlich eine Electron-Application zu erstellen hat uns Herr Dennler. Ich war zuerst ein bisschen skeptisch, ob die Zeit überhaupt dafür ausreichen würde, aber am Ende vom Tag sah es eigentlich gut aus.

Am Anfang des Tages hatten wir keinen Strom an den Arbeitstischen der Schule… Ich habe die Zeit ausgenutzt um Gedanken über die Applikation zu machen. Nebenbei habe ich auf einen Block Papier schriftlich festgehalten, was wir noch genau tun müssen:

* PHP-Form für das Python Script
* Python Übermittlung der Daten an den Node-Server
* Readme schreiben
* Logo erstellen
* Frontend Programmieren
* Functions auf Parameter umschreiben und einen Return-Wert geben

Anschliessend haben wir eine kleine Besprechung durchgeführt und die Arbeit unter uns verteilt und waren am arbeiten. Schlussendlich hat uns Herr Dennler wie erwähnt aufgezeigt, dass sich für so ein Projekt Node viel besser eignet.

Ich habe somit ein Electron Projekt aufgesetzt und ein neues Git-Repository erstellt, während Claudio mit Herr Dennler die Funktionen von Python auf Node umgeschrieben haben. Yassine war zwischenzeitlich mit mir dran und hat noch das Logo mit der Userstory fertig gestellt. Anschliessend fügten wir alles zusammen und ich habe die Applikation so erweitert, dass wir die Daten die wir brauchen Lokal auf den PC abspeichern können und ein Formular dafür gemacht (Herr Dennler musste mir dabei helfen).

## Erkenntnisse

Wie am ersten Tag schon gemerkt gab es viele Probleme mit dem Python-Script die wir am einfachsten mit der Programmiersprache Node umgehen konnten.

Was ich sicherlich gelernt habe ist, dass wirklich viele Sachen einfach mittels Packages gelöst werden können. Persönlich habe ich noch ein Problem mit den Async Funktionen und deren «Promises». Momentan habe ich den Fehler in der Applikation, dass ich die gespeicherten Daten zwar auslese, aber diese zu schnell wieder brauche und somit nur eine undefinierte Variable habe. Aus diesem Grund habe ich das Auslesen der Daten als eine Asynchrone Funktion in unser «function.js» geschrieben und diese mit einem Rückgabewert verziert. Nun bekomme ich ein «promise { <pending> }» zurück… Anschliessend habe ich an jene möglichen Stellen versucht, ein «await» einzubauen. Leider hat Node nie dieses Await akzeptiert und gab entweder die Meldung «'await' has no effect on the type of this expression.» oder «SyntaxError: await is only valid in async functions and the top level bodies of modules»… Ich habe leider wirklich keine Ahnung, wieso ich diese Fehler bekomme, aber ich bin mir sicher, dass ich diese Fehler in Zukunft sobald ich das gelernt habe nicht mehr haben werde da ich eine grosse Verzweiflung daran hatte und es Schmerzvoll war.