

# 2020\_projektarbeit\_ws2020\_by\_go

---

## Test Accounts / Profile

---

### Twitter

Projektarbeit WS2020 by GO (@ByWs2020)

### Facebook

Projektarbeitwszwanzigzwanzig bygo

### Instagram

Projektarbeitws2020 bygo (@projektarbeitws2020bygo)

### Repo - github

[ozwoldFH/2020\\_projektarbeit\\_ws2020\\_by\\_go](#)

## Was?

---

Die Applikation soll auf alle (bzw. einige) bekannte Social Media Plattformen über API Schnittstellen zugreifen. Es sollen Nachrichten/Beiträge auf eine Seite als Liste/Raster angezeigt werden und zeigt von welcher Plattform dieser Beitrag erfasst wurde. Geplant ist auch noch Beiträge zu verfassen und diese über bestimmte/alle Social Media Plattformen zu veröffentlichen. Besonderheiten wie die verschiedenen Like Arten von Facebook oder das Retweeten sollte möglich sein. Filterfunktion ist auch geplant, um beispielsweise gewisse Beiträge von nur einer bestimmten Social Plattform zu filtern und anzuzeigen.

Folgende Punkte sind bereits erledigt -> einige Schnittstellen auf Facebook werden bereits angesprochen. Ein simples Login Verfahren mit MySQL und OAuth ebenso.

`app.post("/user/register", (req, res))`: Hier kann sich ein User registrieren. Es wird ganz simpel der Username und das Passwort aufgenommen und in die MySQL Datenbank gespeichert.

`app.post("/user/login", (req, res))`: Ein simples Login, der überprüft, ob Username und Passwort in der Datenbank eingetragen ist. Der soll später auf ein API Key von Facebook und/oder Instagram mit zurückgeben.

`app.get("/", (req, res))`: Ein simples "Hello World".

`app.post("/fb/login", (req, res))`: Dieser REST API Endpoint gibt uns einen gültigen API Token von Facebook zurück. Hier ist zu beachten, dass ein App Access Token und kein User Access Token zurückgegeben wird.

`app.post("/fb/token", (req, res))`: Ein REST API Endpoint Versuch, um einen User Access Token zu erhalten. Funktioniert leider nicht.

`app.get("/fb/get/fb_id", (req, res):` Ein simples REST API Endpoint, um seinen Facebook ID zu erhalten.

`app.get("/fb/get/posts", (req, res):` Ein REST API Endoint, um alle Posts von seinem Facebook Account anzuzeigen. Dieser wird im JSON Format zurückgegeben.

Nächste Schritte wäre eine Oberfläche zu programmieren. Gedacht wäre, die Technologie React zu nutzen, um Posts von Facebook anzuzeigen sortiert nach Datum. Die Schnittstellen zu Instagram ist auch eingeplant. Eine andere Schnittstelle wäre Twitter API gewesen, aber auf Grund Komplikationen, wurde mir der Developer Account nicht gewährt.

## Warum?

---

Hier wird gezeigt, ob die Erfahrungen von der FH Joanneum ausreichen, um das oben genannte Ziel erreichen zu können. Derweil war dennoch viel Recherche notwendig gewesen.

## Wie?

---

Hier werden folgende Technologien verwendet:

**graph api:** von Facebook. Alle Objekte werden in Facebook "nodes" genannt. Das können z.B. Personen sein oder Fotos, Beiträge etc. Die Verbindungen zwischen Personen wird "edges" genannt. Das kann z.B. die Freundschaft zwischen zwei Personen anzeigen bzw. verbinden. Es gibt zusätzliche Attribute die den Nodes angehören, die "fields" genannt werden. Damit kann man Informationen von einem Objekt lesen. Hraph API basiert auf das "Social Graph" Konzept.

**node.js:** NodeJS wird für Netzwerkbasierde Anwendungen genutzt. Hier ist ein großes Stichwort "Event-basierend". Erhalten wir irgendeine Anfrage, dann wird sofort ein Callback-Funktion ausgeführt. Mit NodeJS kann man sehr einfach einen Server erstellen und man braucht sich um Threads oder ähnliches nicht kümmern.

**react:** React ist eine JavaScript Bibliothek, um sehr einfach und schnell Oberflächen zu erstellen. Im Prinzip wird hier auf "Komponente" aufgebaut. Ein Komponet kann ein Komponet erhalten usw. Jedes Komponent kann seinen eigenen Zustand speichern und falls sich etwas an den Daten ändert, dann wird nur die dementsprechende Komponente aktualisiert.

**visual studio code REST Client:** REST Client ist ein Visual Studio Code Extension um ganz einfache HTTP request Nachrichten zu erstellen und zu senden. Während man in VS Code entwickelt, muss man nicht extra eine andere Anwendung starten, um seine REST API Endpunkte zu testen. Da reicht es einfach aus ein HTTP GET Message zu schreiben und zu senden.

**node.js module fb:** Ein Modul für NodeJS. Der basiert auf das Facebook Javascript SDK, welches im Prinzip für den Client ausgelegt ist. Dieses Modul ist nicht offiziell von Facebook entwickelt worden, sollte aber für diese Projektarbeit ausreichend sein.

**node.js module express:** Ist ein Modul für NodeJS. Express wird genutzt, um einfache Webanwendungen zu erschaffen.

`node.js module mysql`: Ein Modul für NodeJS. Komplette in JavaScript geschrieben und ist im Prinzip ein NodeJS Treiber für MySQL.

`xampp mit mysql`: Wird benötigt, um Login Daten abzuspeichern.