

1 Präambel

Dieses Dokument richtet sich primär an mich selbst, und soll mir helfen, meinen Fortschritt zu dokumentieren.

2 Woche 1

Eigentlich habe ich in dieser Woche nicht viel gemacht. Da ich allerdings zum 16 (Mittwoch angefangen habe, habe ich mir vorgenommen, die Woche 1 von Mittwoch bis Freitag zu zählen). Besonders spannend war für mich die Verwendung von Github Copilot kennen zu lernen. Ansonsten war für mich in das Projekt Surge-Ahead Ablenkung genug, nebst all den neuen Dingen wie Mitarbeiter-Karte und Belehrung zu Verpflichtung dem Grundgesetz durch Hans.

3 Woche 2

3.1 2024-10-21

Der Erkenntnisgewinn des Tages: Promotion wird kein Selbstläufer und ich muss mich noch etwas gedulden. Allerdings werde ich dann wohl eher über IGradU ans Ziel kommen. Morgen gibt es dann noch ein Kennenlernen mit den Leuten von Surge-Ahead. Hierbei habe ich bisher noch nicht besonders viel gutes von denen gehört und mir wurde geraten, dass ich besser resolut und realistisch mit ihren Wünschen umgehen soll.

Jetzt probiere ich noch Mutt zu installieren und zu konfigurieren.

Nachtrag: Habe Mutt unmittelbar danach wieder deinstalliert, da es Grütze ist.

3.2 2024-10-23

Gestern war alles recht interessant und ich habe eine Menge Einblicke bekommen. Abgesehen vom Feuersalarm war ich auch bei einer Bachelor-Prüfung dabei, habe die Surge-Ahead Leute kennengelernt und habe Tobias in eine Übung begleitet.

3.3 2024-10-24

Nach viel rätseln konnte ich heute Phillip kennen lernen und viel über surge-ahead und vieles andere erfahren. Nach Phillips Aussage ist es wichtig, dass ich mich auf die Promotion konzentriere und nicht zu viel Zeit in Surge-Ahead investiere. Er selbst hat sich, wie er selbst sagt auch mit dem Ziel der Promotion bei Hans beworben, konnte sich aber nicht von surge-ahead abheben und war dementsprechend nicht in der Lage seinen Plan durchzusetzen. Außerdem, sollte ich darauf achten, meine work-life-balance zu wahren.

Ich denke, dass ich noch etwas länger die Füße still halten sollte, bevor ich Hans auf mein Promotionswille anspreche.

Scheinbar bin ich auch nicht für surge-ahead angestellt sondern CALM-QE und OMI-TP angestellt.

Im Moment kann ich den vollen Umfang und die Konsequenzen noch nicht abschätzen und warte dementsprechend noch etwas ab.

3.4 2024-10-28

Wochenende war nett, hoffentlich bleibt es nett. Für diese Woche gilt es zu klären:

- Dart basis lernen
- Flutter Basics zu lernen
- Pflichtenheft und Lastenheft abzuklären
- Dashboard basics mit Flutter abzuklären

Stand heute habe ich nun einen guten einblick in Flutter aber muss morgen noch weiter machen: <https://codelabs.developers.google.com/codelabs/flutter-codelab-first6>

4 2024-11-04

Letzte Woche lief ganz knackig und verhältnismäßig gut. Für diese Woche habe ich mir vorgenommen, die einen Redis-Container mit meinem Flutter-Container zu couplen und sobald das funktioniert, das Dashboard zu implementieren.

Allerdings würde das dann in einem Projekt from scratch erfolgen.

5 2024-11-06

Am Tag einer absehbaren 2. Präsidentschaft von Donald Trump werde ich heute das vergleichbar unschöne Thema middleware für die Kommunikation zwischen Redis-Container und Flutter-Web-App (Auch im Container) angehen. Die letzten Tage hat sich herauskristallisiert, dass sich da noch einiges tun muss. Community-Chat der sich diesem Problem widmet: <https://github.com/stablekernel/postgresql-dart/issues/142>

Julian sagt Rest-API kann helfen:

User

Was ist eine rest api

Copilot

Eine REST API (Representational State Transfer Application Programming Interface) ist

eine Schnittstelle, die es ermöglicht, über das HTTP-Protokoll auf Webressourcen zuzugreifen und diese zu manipulieren. REST APIs folgen bestimmten Prinzipien und Konventionen, die sie leicht verständlich und benutzbar machen. Hier sind einige der Hauptmerkmale:

1. ****Ressourcenbasierte Architektur****: Jede Ressource (z.B. Benutzer, Artikel) wird durch eine eindeutige URL identifiziert.
2. ****HTTP-Methoden****: REST APIs verwenden HTTP-Methoden, um Operationen auf Ressourcen durchzuführen:
 - 'GET': Abrufen von Ressourcen
 - 'POST': Erstellen neuer Ressourcen
 - 'PUT': Aktualisieren bestehender Ressourcen
 - 'DELETE': Löschen von Ressourcen
3. ****Stateless****: Jede Anfrage an die API muss alle Informationen enthalten, die notwendig sind, um sie zu verstehen und zu verarbeiten. Der Server speichert keine Sitzungsinformationen zwischen den Anfragen.
4. ****Repräsentationen****: Ressourcen können in verschiedenen Formaten dargestellt werden, z.B. JSON oder XML.
5. ****Hypermedia as the Engine of Application State (HATEOAS)****: Clients erhalten Informationen darüber, wie sie durch die API navigieren können, durch Links in den Antworten.

Ein einfaches Beispiel für eine REST API könnte so aussehen:

- 'GET /users': Liste aller Benutzer abrufen
- 'POST /users': Einen neuen Benutzer erstellen
- 'GET /users/{id}': Details eines bestimmten Benutzers abrufen
- 'PUT /users/{id}': Einen bestimmten Benutzer aktualisieren
- 'DELETE /users/{id}': Einen bestimmten Benutzer löschen

6 2024-11-07

Nach einer einem Interessanten Treffen mit den Bethesda-Boys werden diese nun klar ihre Eingabedaten definieren. In the meantime werde ich ein Dateneingabetool mit Flutter konzipieren und implementieren. Hierbei ist zu beachten, dass wirklich nur geschrieben werden kann (POST).