



# Praktikum Architektur von Informationssystemen

## Sommersemester 2016 – Aufgabe 2

Nils Löwe , Tel. 0176 / 574 550 65

### Rahmenbedingungen für das Praktikum

- Die Bearbeitung findet in 3er-Gruppen statt.
- Die Präsentation Ihrer Lösungen erfolgt im Rahmen der Praktikumstermine.
- Jede Gruppe hat dazu 20-25 Minuten Zeit. Bereiten Sie hierzu Folien und ggf. eine Live-Demonstration Ihres Systems vor.
- Zu einigen der Praktikumsaufgaben gehört eine Implementierung. Sofern in der Aufgabe nicht explizit anders angegeben, dürfen Sie Programmiersprache, Frameworks und Entwicklungsumgebung frei wählen.
- Fangen Sie frühzeitig mit der Bearbeitung Ihrer Aufgabe an; insbesondere nicht erst am Tag oder in der Nacht zuvor.

### Aufgabe 2: Entwurf einer REST API

Hintergrund: Ihre Firma hat das Projekt gewonnen, für das sie im ersten Praktikum die offenen Fragen gestellt haben. Sie sind nun mit dem Entwurf einer der Komponenten der Software betraut und sollen einen Prototypen incl. API entwerfen.

Der Rahmen des Projekts ist die zuvor beschriebene Zeitplanungssoftware für einen Anbieter von Wartungen von Windenergieanlagen. Ihre Software hat intern eine Anbindung an die Betriebsführungszentrale, die Kundendatenbank und die Mitarbeiterdatenbank. Die konkrete Implementierung dieser Anbindungen wird von anderen Teams ausgeführt, Sie sollen die Schnittstelle zu der Web-Application erstellen, mit der die Mitarbeiter auf ihre Wochenpläne zugreifen werden.

## **Vorbereitung: Codeverwaltung, Technologieauswahl und Hosting**

*Versionieren Sie ihren Code und wählen Sie eine Technologie und einen der kostenlosen Hostingprovider aus:*

- Erstellen Sie einen kostenlosen Account bei Gitlab.com (<http://gitlab.com>) und senden Sie mir ihren Accountnamen per Mail, ich lege dann ein git Repository für ihr Team an.
- Wählen Sie einen Provider aus der u.g. Liste aus.
- Wählen Sie ein Framework oder eine Sprache mit der Sie sicher sind, bzw. die sie gern ausprobieren wollen.

## **Aufgabenbeschreibung**

Das System an dem Sie mitarbeiten soll eine REST Schnittstelle anbieten, mit der die Wochenpläne generiert, abgerufen, verändert und gelöscht werden können (CRUD Operationen). Für den ersten Prototypen sind keine speziellen Anforderungen wie Authentication etc. nötig, die Performance ist auch vorerst irrelevant. Es geht lediglich darum, eine Technologie auszuwählen und zu erproben.

Das Format und der Inhalt der Wochenpläne sind für die Aufgabe ebenfalls irrelevant, die Ressource soll lediglich aus einer ID und einem Content-String bestehen. Der Content soll einfach nur 'dummy' Daten darstellen und hat keine weitere Bedeutung für die Aufgabe.

## **Die folgenden Operationen sollen möglich sein:**

- POST mit content: Erstellen eines neuen Wochenplans
- GET mit ID: Abrufen eines Wochenplans mit gegebener ID
- POST mit ID und content: Updaten eines bestehenden Wochenplans
- DELETE mit ID: Löschen eines Wochenplans mit ID

## **Erstellen / POST mit content**

Die Http POST Operation legt einen neuen Wochenplan mit dem gegebenen Inhalt an.

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
-d '{
  "content": "this is an example plan"
}' 'http://your-api.tld/plan'
```

### Response Codes:

- 200: OK, Plan erstellt
- 405: Invalid Input (kein Content)

### Abrufen / GET mit ID

Die Http GET Operation ruft einen Wochenplan mit der gegebenen ID ab.

```
curl -X GET --header 'Accept: application/json' --header 'http://your-api.tld/plan/42'
```

### Response Codes:

- 200: OK, Response body enthält den Content als JSON
- 405: Invalid Input (keine ID)

### Updaten / POST mit ID und Content

Die Http GET Operation ruft einen Wochenplan mit der gegebenen ID ab.

```
curl -X POST --header 'Content-Type: application/json' -d 'content=HalloWelt' 'http://your-api.tld/plan/42'
```

### Response Codes:

- 200: OK
- 404: ID nicht gefunden
- 405: Invalid Input (keine ID oder kein Content)

### Löschen / DELETE mit ID

Die Http DELETE Operation löscht einen Wochenplan mit der gegebenen ID.

```
curl -X DELETE --header 'Accept: application/json' 'http://your-api.tld/plan/42'
```

### Response Codes:

- 200: OK
- 404: ID nicht gefunden
- 405: Keine/Ungültige ID gegeben

## **Bearbeitung**

Ich kann die Versionierung per git dringend empfehlen, ich habe eine Gruppe bei Gitlab angelegt in der wir den Code verwalten können. Wenn Sie sich bei Gitlab (kostenlos) registrieren und mir ihren Account nennen, füge ich Sie zu der AI2016 Gruppe hinzu.

Suchen Sie zunächst eine geeignete Technologie aus. Die späteren Nutzer werden über native Apps (iOS, Android), bzw. über eine Web-App auf die Wochenpläne zugreifen. Eine REST Schnittstelle ist daher eine vorgegebene Anforderung. Es gibt Frameworks mit denen sich sehr einfach APIs direkt erstellen lassen (z.B. Ruby on Rails 5), sowie Tools wie swagger.io, mit denen sich Server Code für diverse Sprachen generieren lässt. Auch mit C, C++ oder Assembler lassen sich Http/REST APIs implementieren, der Aufwand ist jedoch ungleich höher...

Wählen Sie einen der beschriebenen Hoster aus und erstellen Sie eine virtuelle Maschine mit einem OS Ihrer Wahl. Je nach gewählter Sprache/Framework bieten sich der eine oder der andere Provider eher an. Dies ist bereits eine Architekturentscheidung :)

Erstellen Sie die API mit den oben beschriebenen vier Operationen. Die API muss lediglich per curl (oder vergleichbarem Tool) erreichbar sein, eine Oberfläche ist nicht erforderlich.

Erstellen Sie für den Praktikumstermin eine kleine Präsentation in der Sie die getroffenen Architekturentscheidungen vorstellen, sowie einen kleinen Überblick über die verwendeten Tools und Frameworks geben. Der Inhalt der Präsentation sollte die mindestens die folgenden Punkte umfassen:

- Warum wurden genau diese Sprache/Framework ausgewählt
- Warum dieser Provider
- Was lief gut, was lief schlecht bei der Implementierung?
- Was würden Sie wieder so entscheiden, was nicht?
- Kurze Demo der entstandenen API

## **Kostenlose Webhosting Angebote**

| Hoster     | Link                                                                                                                                                               | Einschränkungen                                                                                                                                                                       |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vultr      | <a href="https://www.vultr.com/">https://www.vultr.com/</a><br>( <a href="https://www.vultr.com/">https://www.vultr.com/</a> )                                     | Virtuelle Maschinen mit diversen Betriebssystemen. Die kleinste Instanz ist für einen Monat kostenlos nutzbar.                                                                        |
| Amazon AWS | <a href="https://aws.amazon.com/de/free/">https://aws.amazon.com/de/free/</a><br>( <a href="https://aws.amazon.com/de/free/">https://aws.amazon.com/de/free/</a> ) | Ebenfalls virtuelle Maschinen mit diversen Systemen. Die kleinste Instanz ist für ein Jahr kostenlos. Wesentlich mehr Services als Vultr, dafür auch komplexer.                       |
| Heroku     | <a href="https://www.heroku.com">https://www.heroku.com</a><br>( <a href="https://www.heroku.com">https://www.heroku.com</a> )                                     | Hosting diverser Frameworks, fing mal mit Rails an. Mittlerweile ist durch JVM und Docker eigentlich alles möglich. Die kleinste Instanz ist kostenlos, kann aber nur 18h/Tag laufen. |

### Hinweise und Rahmenbedingungen

- *In der Architektur sind Prototypen in der frühen Projektphase ein wertvolles Werkzeug um Technologien zu erproben.*
- *Wenn Sie am Ende feststellen dass ihre Technologieauswahl Probleme erzeugt hat die sie mit anderen Frameworks oder Sprachen nicht gehabt hätten, dann ist das eine wertvolle Erkenntnis!*

**Die Präsentation ist bis zum Praktikumstermin per Mail an [nils@loewe.io](mailto:nils@loewe.io) abzugeben. In der Praktikumszeit sind keine Änderungen mehr vorgesehen.**