

Praktikum Architektur von Informationssystemen

Sommersemester 2018 – 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.

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 Sicherheitsplattform aus der ersten Praktikumsaufgabe. Die Plattform verbindet Menschen in Notsituationen mit privaten Sicherheitsleuten in der Nähe. Die Plattform bietet eine REST API für unterschiedliche Nutzer, u.a. Anwender, Administratoren, Wachleute oder Controller. Die konkrete Implementierung dieser API Entpunkte wird von unterschiedlichen Teams ausgeführt, Sie sollen die API zu der mobile App der Guards erstellen, mit der Status- und Location Updates verwaltet werden.

Vorbereitung: Codeverwaltung, Technologieauswahl und Hosting

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

- Erstellen Sie einen kostenlosen Account bei github.com (<http://github.com>) oder gitlab.com (<http://gitlab.com>) und legen 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 Guard Accounts generiert, genutzt, 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.

Ein Guard soll einen Namen, eine Email und ein Passwort als Attribute haben.

Die folgenden Operationen sollen möglich sein:

- Registrieren - Einen neuen Guard mit Email / Passwort anlegen
- Login - Als Guard mit Email/Passwort einloggen
- Logout - Eine bestehende Session beenden
- Profil sehen - Als eingeloggter Guard darf ich mein eigenes Profil (Name) sehen

Bearbeitung

Ich kann die Versionierung per git dringend empfehlen. Legen Sie sich einen Account bei gitlab oder github an und senden Sie mit den Link zum Repository per Mail.

Suchen Sie zunächst eine geeignete Technologie aus. Die späteren Nutzer werden über native Apps (iOS, Android), bzw. über eine Web-App 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...

Die meisten Frameworks bieten ebenfalls Bibliotheken oder Plugins, mit denen die Standard "User" Operationen einfach erstellt werden können. Gerade Themen wie Authentication werden schnell sehr komplex. Die Frage Make or Buy stellt sich hier schnell - beide Alternativen haben ihre Vor- und Nachteile.

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. Auch 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. Je nach Framework bekommen Sie die Oberfläche ggf. dazu, das ist natürlich auch in Ordnung.

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

Webhosting Angebote

Hoster	Link	Einschränkungen
Vultr	https://www.vultr.com/ (https://www.vultr.com/)	Virtuelle Maschinen mit diversen Betriebssystemen. Die kleinste Instanz ist für einen Monat kostenlos nutzbar.
Amazon AWS	https://aws.amazon.com/de/free/ (https://aws.amazon.com/de/free/)	Ebenfalls virtuelle Maschinen mit diversen Systemen. Die kleinste Instanz ist für ein Jahr kostenlos. Wesentlich mehr Services als Vultr, dafür auch komplexer.
Heroku	https://www.heroku.com (https://www.heroku.com)	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.
...		Es gibt unzählige weitere Provider mit kostenlosen oder Test-Accounts.

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 abzugeben. In der Praktikumszeit sind keine Änderungen mehr vorgesehen.

Impressum

Datenschutzerklärung

Kontakt