

TU DRESDEN

SERVICE AND CLOUD COMPUTING

PRAKTIKUM

TimeTracker

Eine Microservice-Anwendung für das bessere Zeitmanagement

Autoren:

Sinthujan THANABALASINGAM

Wieland STRAUSS

Dozentin:

Dr.-Ing. Iris BRAUN

19. Januar 2019



Inhaltsverzeichnis

1	Über TimeTracker	2
1.1	Inhalt der Anwendung	2
1.2	Einordnung der Arbeit	2
2	Arbeitsablauf	2
2.1	Team	2
2.2	Vorgehensweise	2
3	Technologien	3
3.1	Architektur	3
3.2	Übersicht verwendeter Technologien	4
3.3	Umsetzung	4
3.4	Schnittstellenbeschreibung	4
3.5	Sicherheit	4
4	Bedienungsanleitung des Clients	4
5	Feedback und Kritik am Praktikum	5

1 Über TimeTracker

1.1 Inhalt der Anwendung

TimeTracker ist eine Anwendung um die Zeit im Überblick zu behalten. Wenn der Nutzer registriert ist, kann er sich jederzeit auf den Server einloggen und die Aufnahme einer Aktivität starten. Nach beenden seiner Aufgabe wird die Aufnahme gestoppt.

Dem Benutzer wird eine Übersicht über seine aufgenommene Zeit angeboten. Dadurch bekommt er einen besseren Überblick über seine aufgebrauchte Zeit und kann sich in Zukunft besser einteilen.

Zusätzlich bietet die Plattform eine globale Statistik über alle Benutzer.

1.2 Einordnung der Arbeit

Die Anwendung wurde als Prüfungsvorleistung im Rahmen einer praktischen Übung der Lehrveranstaltung *Service and Cloud Computing* des Lehrstuhls Rechnernetze der TU Dresden entwickelt. Dozentin und Betreuerin ist Frau Dr.-Ing. Iris Braun.

2 Arbeitsablauf

2.1 Team

Das Entwicklerteam besteht aus den Studenten Sinthujanan Thanabalasingam (Master Informatik, 4. Semester)und Wieland Strauß (Diplom Informatik, 7. Semester).

2.2 Vorgehensweise

Nach der Ideenfindung wurde zuerst die Architektur entworfen. Die Entwicklung wurde nahezu vollständig nach dem Prinzip des TDD (Test Driven Development) betrieben. Die API wurde parallel mit *Swagger* erstellt. Auch die Verwendung von Docker und die Entwicklung des Frontends wurden zeitgleich umgesetzt. Zuletzt wurde die Anwendung auf einen Server von DigitalOcean ausgerollt

Der Arbeitsablauf organisierte sich aus mehreren Treffen. Bei diesem wurden die Planung durchgeführt, der Großteil der Anwendung geschrieben und Aufgaben für die selbstständige Durchführung festgelegt. Aufgrund der verschiedenen Expertise wurde zum Teil Pair Programming durchgeführt.

3 Technologien

3.1 Architektur

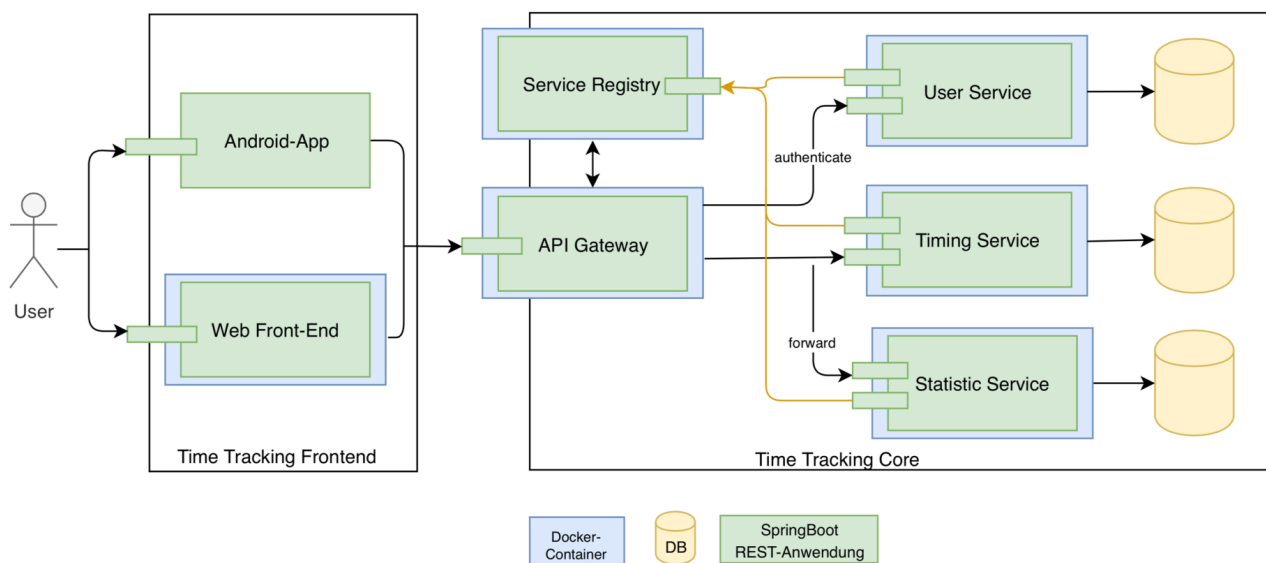


Abbildung 1: Architektur von TimeTracker

Anmerkung: Die Andoid-App wurde nicht im Rahmen der Lehrveranstaltung fertig gestellt und ist somit noch ausstehend.

3.2 Übersicht verwendeter Technologien

Tabelle 1: Technologien

Name	Version	Verwendung
Java	bla	bla
Mavent	3.1.0	bla
Spring boot	bla	bla
MySQL	bla	bla
Netflix Zuul	bla	bla
Netflix Eureka	bla	bla
JSON	bla	bla
REST	bla	bla
React	bla	bla
single.spa	bla	bla
OpenAPI + Swagger	bla	bla
Postman	bla	bla
Docker	bla	bla

3.3 Umsetzung

3.4 Schnittstellenbeschreibung

- von swagger?

3.5 Sicherheit

JSON Web Token

HTTPS

4 Bedienungsanleitung des Clients

TimeTracker bietet folgende Funktionen:

- Time Tracking

- Anlegen von Aktivitäten
- Tracking für verschiedene Aktivitäten
- Abruf von Statistiken (Benutzer/Global)

5 Feedback und Kritik am Praktikum