

## **Projektbericht REST-API**

vorgelegt von

Arne Kreuz,

Joshua Nestler,

Moritz Schönenberger

betreut und begutachtet von

Prof. Dr. Reinhard Brocks

Saarbrücken, 10. Mai 2024



# Inhaltsverzeichnis

<b>1</b>	<b>Projekt</b>	<b>1</b>
1.1	Zielsetzung . . . . .	1
1.2	Projektbeiträge . . . . .	1
1.3	Verwendete Technologien . . . . .	1
<b>2</b>	<b>Spezifikation</b>	<b>3</b>
2.1	OpenAPI . . . . .	3
<b>3</b>	<b>Implementierung Webserver</b>	<b>5</b>
3.1	Jakarta EE . . . . .	5
3.1.1	Probleme . . . . .	5
3.2	Spring Boot . . . . .	5
3.3	ASP.NET Core . . . . .	5
<b>4</b>	<b>Tests</b>	<b>7</b>
<b>5</b>	<b>Workflow</b>	<b>9</b>
<b>6</b>	<b>Fazit</b>	<b>11</b>
6.1	Zusammenfassung . . . . .	11
6.2	Lessons Learned . . . . .	11
6.3	Ausblick . . . . .	11
	<b>Abbildungsverzeichnis</b>	<b>13</b>
	<b>Tabellenverzeichnis</b>	<b>13</b>
	<b>Listings</b>	<b>13</b>
	<b>Abkürzungsverzeichnis</b>	<b>15</b>
<b>A</b>	<b>Representational State Transfer (REST)-Endpunkte</b>	<b>19</b>
A.1	User . . . . .	19



# 1 Projekt

## 1.1 Zielsetzung

Write

## 1.2 Projektbeiträge

Write

## 1.3 Verwendete Technologien

Technologie	Version	Link
Java	17	<a href="https://www.java.com/">https://www.java.com/</a>
Maven	3.8.5	<a href="https://maven.apache.org/">https://maven.apache.org/</a>
Jakarta EE	11.0.0	<a href="https://jakarta.ee/">https://jakarta.ee/</a>
Wildfly	32.0.0	<a href="https://www.wildfly.org/">https://www.wildfly.org/</a>
Spring Boot	3.2.4	<a href="https://spring.io/projects/spring-boot">https://spring.io/projects/spring-boot</a>
.NET	8.0	<a href="https://dotnet.microsoft.com/">https://dotnet.microsoft.com/</a>
ASP.NET Core	8.0	<a href="https://dotnet.microsoft.com/apps/aspnet">https://dotnet.microsoft.com/apps/aspnet</a>
H2	2.2.224	<a href="https://www.h2database.com/">https://www.h2database.com/</a>
MariaDB	11.3	<a href="https://mariadb.org/">https://mariadb.org/</a>
JMeter	5.4.1	<a href="https://jmeter.apache.org/">https://jmeter.apache.org/</a>
Postman	10.23.4	<a href="https://www.postman.com/">https://www.postman.com/</a>
Docker	26.1.2	<a href="https://www.docker.com/">https://www.docker.com/</a>
IntelliJ IDEA	2024.1.2	<a href="https://www.jetbrains.com/idea/">https://www.jetbrains.com/idea/</a>
Rider	2024.1.2	<a href="https://www.jetbrains.com/rider/">https://www.jetbrains.com/rider/</a>
Git	2.45.0	<a href="https://git-scm.com/">https://git-scm.com/</a>
GitHub	-	<a href="https://github.com/">https://github.com/</a>
GitHub Actions	-	<a href="https://github.com/features/actions">https://github.com/features/actions</a>
OpenAPI	3.0.3	<a href="https://www.openapis.org/">https://www.openapis.org/</a>
Mermaid	10.9.0	<a href="https://mermaid-js.github.io/mermaid/">https://mermaid-js.github.io/mermaid/</a>
LaTeX	2023	<a href="https://www.latex-project.org/">https://www.latex-project.org/</a>

Tabelle 1.1: Verwendete Technologien



## 2 Spezifikation

### 2.1 OpenAPI

Write





## 3 Implementierung Webserver

### 3.1 Jakarta EE

Als Referenzimplementierung für dieses Projekt wurde Jakarta EE verwendet. Jakarta EE ist eine Sammlung von Spezifikationen für Enterprise-Java-Anwendungen. Es ist eine Weiterentwicklung von Java EE, welches von Oracle entwickelt wurde. Jakarta EE wird von der Eclipse Foundation entwickelt und ist Open Source.

Write

#### 3.1.1 Probleme

##### Abhängigkeiten vom Webserver

Jakarta EE benötigt einen separaten Webserver, um die Anwendung auszuführen. Die Konfiguration und das Setup sind stark abhängig von dem verwendeten Webserver. In diesem Fall wurde Wildfly<sup>1</sup> verwendet.

Die meisten dieser Webserver sind nicht dazu ausgelegt, eine einzelne Anwendung zu hosten. Dies macht es schwieriger, die Anwendung in einer Microservice-Architektur zu betreiben.

Um dieses Projekt mit diesem Webserver zu betreiben, ist es notwendig, ein Konfigurations-Skript zu schreiben, welches in den Bauprozess des Docker-Abbilds integriert wird.

- Konfiguration und Setup ist stark abhängig von dem verwendeten Webserver (in diesem Fall Wildfly)

### 3.2 Spring Boot

Write

### 3.3 ASP.NET Core

Write

---

<sup>1</sup><https://www.wildfly.org/> | Abgerufen: 2024-05-10, 12:44 Uhr



## 4 Tests

Write



## 5 Workflow

Write



## 6 Fazit

Write

### 6.1 Zusammenfassung

Write

### 6.2 Lessons Learned

Write

### 6.3 Ausblick

Write





# Abbildungsverzeichnis

## Tabellenverzeichnis

1.1	Verwendete Technologien . . . . .	1
A.1	User-Endpunkte . . . . .	19

## Listings



# Abkürzungsverzeichnis

REST    Representational State Transfer



# Anhang



# A REST-Endpunkte

## A.1 User

Methode	Endpunkt	Parameter	Beschreibung
GET	/users	-	Gibt alle Benutzer zurück
GET	/users/{id}	id	Gibt den Benutzer mit der ID zurück
POST	/users	-	Erstellt einen neuen Benutzer
PUT	/users/{id}	id	Aktualisiert den Benutzer mit der ID
DELETE	/users/{id}	id	Löscht den Benutzer mit der ID

Tabelle A.1: User-Endpunkte





## **Kolophon**

Dieses Dokument wurde mit der L<sup>A</sup>T<sub>E</sub>X-Vorlage für Abschlussarbeiten an der htw saar im Bereich Informatik/Mechatronik-Sensortechnik erstellt (Version 2.23, März 2022). Die Vorlage wurde von Yves Hary und André Miede entwickelt (mit freundlicher Unterstützung von Thomas Kretschmer, Helmut G. Folz und Martina Lehser). Daten: (F)10.95 – (B)426.79135pt – (H)688.5567pt