

## REST INTERFACE ALS ALTERNATIVE ZU WEB INTERFACE

### WAS IST REST

Representational State Transfer (REST) bezeichnet ein Programmierparadigma für verteilte Systeme, insbesondere für Webservices.

REST hat das Ziel, einen Architekturstil zu schaffen, der die Anforderungen des modernen Web besser darstellt. Dabei unterscheidet sich REST vor allem in der Forderung nach einer einheitlichen Schnittstelle von anderen Architekturstilen.

Der Zweck von REST liegt schwerpunktmässig auf der Maschine-zu-Maschine-Kommunikation. Anders als bei vielen verwandten Architekturen kodiert REST keine Methodeninformation in den URI, da der URI Ort und Namen der Ressource angibt, nicht aber die Funktionalität, die der Web-Dienst zu der Ressource anbietet.

Der Vorteil von REST liegt darin, dass im WWW bereits ein Grossteil der für REST nötigen Infrastruktur vorhanden ist, und viele Web-Dienste per se REST-Konform sind.

REST implementiert typisch CRUD.

### CRUD

CRUD ist ein Akronym und umfasst die folgenden Datenbankoperationen:

CRUD-Operation	SQL-92	HTTP (REST)
Create	INSERT	PUT oder POST
Read (Retrieve)	SELECT	GET
Update	UPDATE	PATCH oder PUT
Delete (Destroy)	DELETE	DELETE

### WAS SIND DTOs

Das Data Transfer Object (DTO) bündelt mehrere Daten in einem Objekt, sodass sie durch einen einzigen Programmaufruf übertragen werden können. DTOs werden in verteilten Systemen eingesetzt, um mehrere zeitintensive Fernzugriffe durch einen einzigen zu ersetzen.

DTOs abstrahieren einen Teil der Business Objects.

### WAS IST JSON

Die JavaScript Object Notation (JSON), ist ein kompaktes Datenformat in einer einfachen lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen.

JSON ist unabhängig von den Programmiersprachen.

JSON wird zur Übertragung und zum Speichern von strukturierten Daten eingesetzt; es dient als Datenformat bei der Serialisierung. Insbesondere bei Webanwendungen und mobilen Apps wird es in Verbindung mit JavaScript, Ajax oder Websockets zum Transfer von Daten zwischen dem Client und dem Server genutzt.

### POSTMAN FÜR DEN WEBZUGRIFF

Postman ist eine komplette API-Entwicklungsumgebung. Es unterstützt alle Aspekte der API-Entwicklung und basiert auf einer einzigen Ebene und gewährleistet so eine konsistente Performance und Benutzererfahrung.

## ENTWICKLUNGSZIEL

Erster Schritt je für Customer, ohne Security für REST

### DTOs FÜR (JSON)

- /CRM/src/main/java/ch/zli/m223/CRM/model/dto/CustomerDto.java
- /CRM/src/main/java/ch/zli/m223/CRM/model/dto/MemoDto.java

Optional:

- /CRM/src/main/java/ch/zli/m223/CRM/model/dto/CredentialDto.java
- /CRM/src/main/java/ch/zli/m223/CRM/model/dto/RolesDto.java
- /CRM/src/main/java/ch/zli/m223/CRM/model/dto/UserDto.java

### ERGÄNZEN VON MAVEN DEPENDENCIES

«pom.xml» ⇒ Tab «pom.xml»

```
<dependency>
  <groupId>com.jayway.jsonpath</groupId>
  <artifactId>json-path</artifactId>
  <scope>test</scope>
</dependency>
```

### REST CONTROLLER

- /CRM/src/main/java/ch/zli/m223/CRM/rest/CustomerRestController.java

Optional:

- /CRM/src/main/java/ch/zli/m223/CRM/rest/AuthenticationRestController.java
- /CRM/src/main/java/ch/zli/m223/CRM/rest/UserRestController.java

## **QUELLENVERZEICHNIS**

### **REST**

[https://de.wikipedia.org/wiki/Representational\\_State\\_Transfer](https://de.wikipedia.org/wiki/Representational_State_Transfer)

### **CRUD**

<https://de.wikipedia.org/wiki/CRUD>

### **DTOs**

<https://de.wikipedia.org/wiki/Transferobjekt>

### **JSON**

[https://de.wikipedia.org/wiki/JavaScript\\_Object\\_Notation](https://de.wikipedia.org/wiki/JavaScript_Object_Notation)

### **POSTMAN**

<https://www.getpostman.com/>