**Diskussion REST API**

Die Konzepte für REST Schnittstellen werden anhand der Inhalte der Vorlesungsfolien für die erstellte API diskutiert und vorgestellt. Die vollständige Dokumentation der API erfolgt aber in der Swagger Dokumentation.

**Ressource**

Die API basiert auf zahlreichen Ressourcen, die mit Hilfe von Swagger erstellt wurden. Diese sind in der untenstehenden Anwendung abgebildet. Beispiele aus der Anwendung sind:

* /events
* /user
* /events/id
* …

**Adressierbarkeit**

Die Adressierbarkeit der Ressourcen ist mit Hilfe von URIs sichergestellt, die auf der durch Swagger erstellten Dokumentation basiert. Auch die Verwendung physischer Ressourcen wurde in der API vermieden. Beispiele sind die Verwendung von /events. Als Rückgabewert werden Repräsentationen im json Format verwendet. Für eine Server der lokal über den Port 3000 erreicht werden kann, sind beispielsweise folgende Adressen erreichbar:

* <http://localhost:3000/events>
* <http://localhost:3000/user>
* …

**Repräsentation**

Wie in den Konzepten von REST festgehalten werden alle Operationen mit Hilfe von Repräsentationen durchgeführt. Der Einfachheit halber wird allerdings nur mit Repräsentationen in json gearbeitet. Allerdings wäre auch eine Verwendung von weiteren Repräsentationen wie beispielsweise XML leicht zu ergänzen. Eine beispielhafte Repräsentation ist ein json Objekt eines Events, das wie folgt aussehen kann:

{

"name": "Software Conference",

"description": "Conference on Software-Quality",

"location": "Linz",

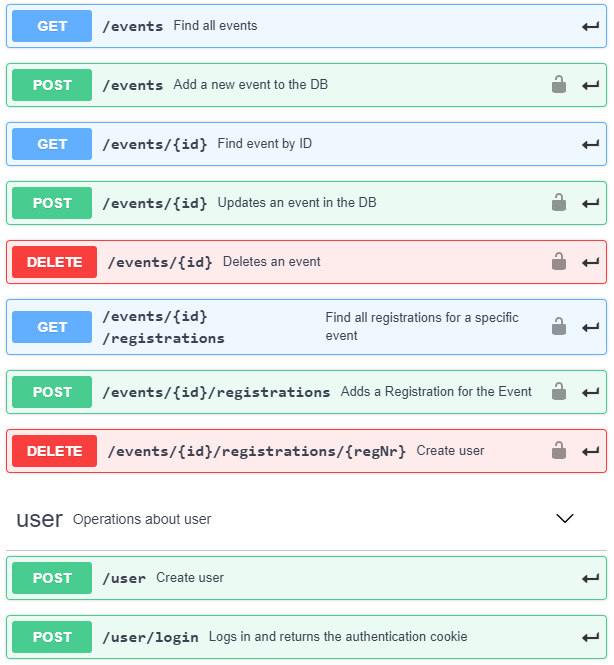
"date": "10.7.2019",

"maxParticipants": "100",

"host": "Moritz"

}

**HTTP Schnittstelle**

Zur Kommunikation mit dem REST Service dienen die bekannten HTTP Methoden, wobei vielseitige Operationen unterstützt bzw. verwendet werden. Konkret verwendet wurden dabei GET, POST und DELETE.

Auch die Antworten erfolgen mittels HTTP üblicher Status Codes um so Erfolgreiche Abarbeitung, Fehler in der Eingabe oder auch der Durchführung der Anfragen signalisieren zu können.

**Zustandslosigkeit**

Das komplette REST Backend ist zustandslos und hält keinerlei Daten bezüglich einer Session am Server bereit. Auch für die Autorisierung der Benutzer ist konnte es durch die Verwendung von Json Web Tokens erreicht werden, dass keine Daten zum Benutzer am Server gespeichert sein müssen. Um Zugriff zu geschützten Operationen zu erhalten braucht der Client einen gültigen Token, der am Server verifiziert werden kann. Dadurch kann auch eine entsprechende Skalierbarkeit der Anwendung sichergestellt werden.

Zusätzlich werden alle Daten wie beispielsweise die ID eines bestimmten Events bei jedem Aufruf mitgegeben. Die Übertragung erfolgt dabei entweder als Teil der URI oder in Form eines Json Objektes im Body des Aufrufes. Lediglich der Token wird (wie bei Authorizations dieser Art üblich) im Header übertragen.