

# REST-Spezifikation zu "Interne Mitfahrgelegenheit"

Moritz Basel

2. Januar 2018

## 1 Einführung

### 1.1 Zweck des Dokuments

Dieses Dokument dient als Handbuch für Frontenddeveloper zur Software "Interne Mitfahrgelegenheit". Es wird versucht, die REST-API vollständig aufzuzählen und alle möglichen Anwendungsfälle zu erfassen. Dennoch besteht die Möglichkeit, dass einzelne Spezialfälle (noch) nicht abgedeckt werden. Bei Auffinden undokumentierten Verhaltens, bei Featureanfragen und Sonstigen bitten wir um Rückmeldung unter <incomplete>.

### 1.2 Terminologie

**Endpoints** bezeichnen eine spezifische (z.B. /api/check\_api oder gruppierte (z.B. /api/users/<uid>) URIs

### 1.3 Allgemeines

Alle Endpunkte lesen und schreiben ausschließlich JSON. Der Content-Type in jedem Http-Request-Header muss als 'application/json' vorliegen.

Für Endpunkte, bei denen Autorisierung nötig ist, wird diese erbracht, indem der Http-Header 'Authorization' gesetzt ist und den Access-Token der von /api/auth erhalten wurde, in folgender Form enthält

'Authorization' : 'Bearer ey.....', z.B.:

'Authorization' : 'Bearer eyJhbGciOiJI.3u4zudusua.asudfuu23'

## Inhaltsverzeichnis

1	Einführung	1
1.1	Zweck des Dokuments	1
1.2	Terminologie	1
1.3	Allgemeines	1

<b>2</b>	<b>JSON-Objekte</b>	<b>2</b>
2.1	User . . . . .	2
2.2	Appointment . . . . .	2
<b>3</b>	<b>User-Management</b>	<b>3</b>
3.1	Überblick . . . . .	3
3.2	POST /api/users . . . . .	3
3.3	POST /api/users . . . . .	3
3.4	PUT,PATCH /api/users . . . . .	3
3.5	GET /api/users/<UID> . . . . .	3
3.6	POST /api/auth . . . . .	3
<b>4</b>	<b>Appointment-Funktionalitäten</b>	<b>4</b>
4.1	Überblick . . . . .	4
4.2	GET /api/appointments/<appointmentID> . . . . .	4
<b>5</b>	<b>Entwicklungs-API</b>	<b>4</b>
5.1	Überblick . . . . .	4
5.2	DELETE /api/dev/removeUser/<username> . . . . .	4

## 2 JSON-Objekte

### 2.1 User

Die JSON-Repräsentation eines Users enthält folgende Felder:

1. username
2. email
3. phoneNumber
4. globalAdminStatus
5. id

### 2.2 Appointment

Die JSON-Repräsentation eines Appointments enthält folgende Felder:

1. id
2. startLocation
3. startTime
4. repeatTime

Zur Semantik des Felds "repeatTime":  
Folgende Werte sind erlaubt

"none"	Das Appointment wird nicht wiederholt
"daily"	Am nächsten Tag findet das Appointment erneut statt
"weekly"	Nach einer Woche findet das Appointment erneut statt

## 3 User-Management

### 3.1 Überblick

<b>POST</b> /api/users	Erstellt einen neuen Benutzer
<b>GET</b> /api/users/<UID>	User-Profil
<b>GET</b> /api/users/<Username>	User-Profil
<b>POST</b> /api/auth	Login-Token Erstellung
<b>PUT, PATCH</b> /api/users/uid	User-Update

### 3.2 POST /api/users

Erwartet eine vollständige JSON-Repräsentation eines Benutzers und zusätzlich das password - Feld.

### 3.3 POST /api/users

Erwartet eine JSON-Repräsentation eines Benutzers und erstellt diesen nach Validierung der Daten. HTTP-Status 201 bei Erfolg.

### 3.4 PUT,PATCH /api/users

Erwartet eine JSON-Repräsentation eines Benutzers, sowie optional ein neues Password im Feld password und überschreibt die alten Nutzerdaten.

### 3.5 GET /api/users/<UID>

Agiert wie erwartet. Nur der Benutzer selbst sowie Globale Administratoren haben Zugriffsrechte.

### 3.6 POST /api/auth

Gibt den Login-Token im JSON-Eintrag 'access\_token' zurück. Dieser muss bei Zugriffen auf geschützte http-Endpunkte im Header-Feld 'Authorization' in allen Requests mitgegeben werden.

Außerdem werden alle Nutzerdaten in den Feldern 'username', 'email', 'phoneNumber', 'globalAdminStatus' versendet. Dies wird nur zur Unterstützung getan, da im JWT Body alle Felder in Base64 vorhanden sind.

## 4 Appointment-Funktionalitäten

### 4.1 Überblick

<b>GET</b> /api/appointments/<appointmentID>	Daten zum gewähltem Appointment
<b>GET</b> /api/users/<userID>/appointments	Liste der Appointments zum Nutzer
<b>GET</b> /api/appointments	Liste aller Appointments
<b>POST</b> /api/appointments	Erstellt ein neues Appointment

### 4.2 GET /api/appointments/<appointmentID>

Gibt eine JSON-Repräsentation des Appointments zurück.

## 5 Entwicklungs-API

### 5.1 Überblick

<b>DELETE</b> /api/dev/removeUser/<uname>	Löscht bestimmten Nutzer
<b>GET</b> /api/dev/check_token	User-ID des Nutzers
<b>GET</b> /api/dev/check-api	Simpler Test
<b>GET</b> /api/dev/log	Log

### 5.2 DELETE /api/dev/removeUser/<username>

Löscht den gewählten Nutzer vollständig. Kann nur vom Nutzer selbst und von globalen Administratoren durchgeführt werden.