

RESTful API Resources

Alle Daten werden in JSON übermittelt.

Alle Fehler Antworten haben mindestens ein "msg" Attribut. Häufig auch ein "error" Attribut mit genaueren Informationen.

/TESTINGONLY/user/:id		Löschtung eines vorhandenen Benutzers (benötigt für UnitTests)
DELETE		
result:	Status: 200 oder 500	

/register		Registrierung eines neuen Benutzers
POST		
params:	name:	Benutzername
	email:	Benutzer E-Mail
	pwd:	Benutzerpasswort
result:	Status:	201, 400, 409, 500
		201: Attribut "user" liefert zusätzlich noch weitere Benutzerinformationen zurück.

/login		Einloggen eines vorhandenen Benutzers
POST		
params:	name:	Benutzername
	pwd:	Benutzerpasswort
result:	Status:	201, 400, 409, 500
		201: Attribut "user" liefert zusätzlich noch "id" und "name" des Benutzers. Attribut "tocken" liefert den aktuellen Session Tocken.

/auto		Autos holen. (Filterung möglich)
POST		
params:	{ "<filter-id>": "<filter-data>", "<filter2-id>": "<filter2-data>", ... }	
	filter-id:	muss mit Datei übereinstimmen. (src/server/src/filter/_base.js)
	filter-data:	dokumentation, wie die Daten auszusehen haben steht in den einzelnen Filter Dateien. (src/server/src/filter/<filter-name>.js)
result:	Status:	200, 500
		200: Attribut "data" enthält die Autos.

/auto/best	Nur die Besten Autos holen.	
GET		
params:	"limit": Optionaler Parameter zum Begrenzen der geholten Autos.	
	filter-id:	muss mit Datei übereinstimmen. (src/server/src/filter/_base.js)
result:	dokumentation, wie die Daten auszusehen haben steht in den einzelnen Filter Dateien. (src/server/src/filter/<filter-name>.js)	
	Status: 200, 500	
200: Attribut "data" enthält die Autos.		

/auto/filter	Alle Filter holen.	
GET		
result:	Status: 200 oder 500	
	200: Attribut "data" enthält alle Filter	

/auto/mainfilter	Nur die Filter der Hauptseite holen.	
GET		
result:	Status: 200 oder 500	
	200: Attribut "data" enthält alle Filter der Hauptseite.	

/rating/user/:id	Alle Benutzer Bewertungen holen	
GET		
result:	Status: 200 oder 500	
	200: Attribut "data" enthält alle Benutzer Bewertungen	

/rating/auto/:id	Alle Auto Bewertungen holen	
GET		
result:	Status: 200 oder 500	
	200: Attribut "data" enthält alle Auto Bewertungen	

/rating/auto/:id	Bewertung von einen Benutzer für ein Auto setzen	
POST		
params:	"userId"	Benutzer ID
	"autoId"	Auto ID
	"rating"	1-5 Sterne Bewertung
result:	Status: 200 oder 500	
	200: Attribut "data" enthält alle Auto Bewertungen	