Trip Exchange System

# RESTful API Overview

The RESTful API is exposing web services to work with the server. The API has the following web service endpoints:

|  |  |  |
| --- | --- | --- |
| **HTTP Method** | **Web service endpoint** | **Description** |
| POST (public) | api/users/register | Registers a new user in the system. |
| POST (public) | api/users/login | Logs in a user in the system. |
| GET | api/users/userInfo | Gets the information about the current logged user |
| POST | api/users/logout | Logs out a user from the system. |
| GET (public) | api/stats | Gets statistics for the system – total trips, total users, total drivers and total finished trips. |
| POST | api/trips | Creates a new trip with current user as driver. |
| GET (public) | api/trips | Gets top 10 trips, no matter the free seats. Sorted by their date of creation. |
| GET | api/trips?page=P | Gets the trips at positions from page P. All trips are sorted by date of creation and are at most 10. |
| GET | api/trips?page=P&sort=S&order=O | Gets the trips at positions from page P. All trips are sorted by “driver”, “date”, “seats” and are at most 10. Order is “asc” or “desc”. If order is missing, it is assumed “asc” as default. |
| GET | api/trips?page=P&from=F&to=T | Gets the trips at positions from page P.  Trips are filtered by from town (F) and to town (T) (one of which may be missing). All trips are sorted by date of creation and are at most 10. |
| GET | api/trips?page=P&finished=true | Gets the trips at positions from page P, includes all the already finished trips (with time of travel earlier than the current time). By default, if finished is missing, the server returns only the upcoming trips. |
| GET | api/trips?page=P&isMine=true | Gets the trips at positions from page P. Filters the trips by only the current user’s created or joined trips. |
| GET | api/trips/{tripId} | Gets trip with id = tripId providing all the passengers usernames. |
| PUT | api/trips/{tripId} | Join a trip with id = tripId with available seats – assigns the current user as passenger. |
| GET (public) | api/drivers | Get top 10 drivers, sorted by username. |
| GET | api/drivers?page=P | Gets the drivers at positions from page P, sorted by username. |
| GET | api/drivers?page=P&username=U | Gets the drivers at positions from page P, sorted by username and filtered by username (filter is “*contains*” and case-insensitive). |
| GET | api/drivers/{driverId} | Gets driver with id = driverId, along with his/her 50 most recent upcoming trips. |

**\*NOTE: Parameters *sort*, *order*, *from*, *to*, *finished* and *isMine* on the *trips* API point are optional and can be combined with the *page* parameter. For example *api/trips?page=5&sort=seats&to=Sofia* is valid request.**

**\*NOTE: Paging numeration always starts from 1.**

# RESTful API Details

## Users

### **Register**

|  |  |  |  |
| --- | --- | --- | --- |
| api/users/register | | | POST |
| Request | { "email": "test@test.com",  "password": "12345678",  "confirmPassword": "12345678",  "isDriver": "true",  "car": "Mazda 6 2014" } | | |
| Response | Null | 200 OK | |

### **Login**

|  |  |  |
| --- | --- | --- |
| api/users/login | | POST |
| Request | { "grant\_type": "password"  "username": " test@test.com",  "password": "12345678" } | |
| Response | {  "access\_token": "deYtpuVHa4Ba-1t-…..\_hdTI-1Pa-bNiASNQz",  "token\_type": "bearer",  "expires\_in": 1209599,  "userName": "test@test.com",  ".issued": "Fri, 26 Sep 2014 10:39:16 GMT",  ".expires": "Fri, 10 Oct 2014 10:39:16 GMT"  } | |

### **User info**

|  |  |  |
| --- | --- | --- |
| api/users/userInfo | | GET |
| Headers | "Authorization" = "Bearer deYtpuVHa4Ba-1t-…..\_hdTI-1Pa-bNiASNQz" | |
| Response | {  "email": "test@test.com",  "isDriver": true,  "car": "Mazda 6 2014"  } | |

### **Logout**

|  |  |  |  |
| --- | --- | --- | --- |
| api/users/logout | | | POST |
| Headers | "Authorization" = "Bearer deYtpuVHa4Ba-1t-…..\_hdTI-1Pa-bNiASNQz" | | |
| Response | Null | 200 OK | |

## Statistics

|  |  |  |
| --- | --- | --- |
| api/stats | | GET |
| Response | {  "trips": 1337,  "finishedTrips": 919,  "users": 654,  "drivers": 321  } | |

## Trips

### **Create trip**

|  |  |  |
| --- | --- | --- |
| api/trips | | POST |
| Headers | { "Authorization": "Bearer deYtpuVHa4Ba-1t-….. \_hdTI-1Pa-bNiASNQz "} | |
| Request | From, to, dateOfLeaving, numberOfFreeSeats | |
| Response | Id, driverId, driverName, from, to, dateOfCreating, dateOfLeaving, numberOfFreeSeats | |

### Get trips

|  |  |  |
| --- | --- | --- |
| api/trips | | GET |
| Headers | { "Authorization": "Bearer deYtpuVHa4Ba-1t-….. \_hdTI-1Pa-bNiASNQz "} | |
| Response | Collection of { Id, driverId, driverName, from, to, dateOfCreating, dateOfLeaving, numberOfFreeSeats, isMine } | |

\*NOTE: All other GET requests to trips return the same data but filtered, sorted and paged.

### **Get trip detail**

|  |  |  |
| --- | --- | --- |
| api/trips/{tripId} | | GET |
| Headers | { "Authorization": "Bearer deYtpuVHa4Ba-1t-….. \_hdTI-1Pa-bNiASNQz "} | |
| Response | Id, driverId, driverName, from, to, dateOfCreating, dateOfLeaving, numberOfFreeSeats, isMine, Collection of { passengerNames as string } | |

### **Join trip**

|  |  |  |
| --- | --- | --- |
| api/trips/{tripId} | | PUT |
| Headers | { "Authorization": "Bearer deYtpuVHa4Ba-1t-….. \_hdTI-1Pa-bNiASNQz "} | |
| Response | {} | |

## Drivers

### **Get drivers**

|  |  |  |
| --- | --- | --- |
| api/drivers | | GET |
| Headers | { "Authorization": "Bearer deYtpuVHa4Ba-1t-….. \_hdTI-1Pa-bNiASNQz "} | |
| Response | Collection of { Id, name, numberOfUpcomingTrips, numberOfTotalTrips } | |

\*NOTE: All other GET requests to drivers return the same data but filtered, sorted and paged.

### **Get driver detail**

|  |  |  |
| --- | --- | --- |
| api/drivers/{driverId} | | GET |
| Headers | { "Authorization": "Bearer deYtpuVHa4Ba-1t-….. \_hdTI-1Pa-bNiASNQz "} | |
| Response | Id, name, numberOfUpcomingTrips, numberOfTotalTrips, Collection of { Id, from, to, dateOfCreating, dateOfLeaving, numberOfFreeSeats, isMine } | |