

RideApp-JS API Documentation

Table of Contents

Base URL	2
Authentication & User Management.....	2
Register a New User	2
User Login.....	2
Get User Profile.....	3
Update User Profile	3
Change User Password	3
Get User Ride History	4
Rider Actions	5
Book a Ride	5
Cancel a Ride.....	5
Make a Payment for a Ride	5
Rate a Driver	6
Driver Actions	7
Register a Vehicle.....	7
Update Vehicle Information	7
Accept a Ride.....	7
Update Ride Status	8
Rate a Rider	8

Base URL

http://localhost:3000

Authentication & User Management

These endpoints handle user creation, login, and profile management for all user types. **Base**

Path: /app/users

Register a New User

Creates a new user account for either a rider or a driver.

- **Method:** POST
- **Endpoint:** /app/users/register
- **Authentication:** Public Endpoint

JSON Body (for a Rider):

```
{
  "fullName": "",
  "email": "",
  "phone": "",
  "password": "",
  "role": "rider"
}
```

JSON Body (for a Driver):

```
{
  "fullName": "",
  "email": "",
  "phone": "",
  "password": "",
  "role": "driver",
  "licenseNumber": ""
}
```

- **Additional Points:** A `licenseNumber` is mandatory when the `role` is "driver".

User Login

Authenticates a user and sets a secure, httpOnly cookie named `authToken`.

- **Method:** POST
- **Endpoint:** /app/users/login
- **Authentication:** Public Endpoint

JSON Body:

```
{
  "email": "",
  "password": ""
}
```

- **Additional Points:** On successful login, an `authToken` cookie is set. API clients like Thunder Client will automatically include this cookie in all subsequent requests to protected endpoints.

Get User Profile

Fetches the profile information of the currently logged-in user.

- **Method:** GET
- **Endpoint:** `/app/users/profile`
- **Authentication: Protected Endpoint** (Requires login)
- **JSON Body:** None

Update User Profile

Updates the name and phone number of the logged-in user.

- **Method:** PUT
- **Endpoint:** `/app/users/profile`
- **Authentication: Protected Endpoint**

JSON Body:

```
{
  "fullName": "",
  "phone": ""
}
```

Change User Password

Updates the password for the logged-in user.

- **Method:** PATCH
- **Endpoint:** `/app/users/password`
- **Authentication: Protected Endpoint**

JSON Body:

```
{
  "oldPassword": "",

```

```
"newPassword": ""  
}
```

Get User Ride History

Fetches a complete ride history for the logged-in user, whether they are a rider or a driver.

- **Method:** GET
- **Endpoint:** /app/users/history
- **Authentication:** Protected Endpoint
- **JSON Body:** None

Rider Actions

These endpoints are for actions performed by users with the "rider" role.

Base Path: /app/riders

Book a Ride

Creates a new ride request.

- **Method:** POST
- **Endpoint:** /app/riders/book
- **Authentication:** Protected Endpoint

JSON Body:

```
{
  "pickup_location": "",
  "drop_location": ""
}
```

- **Additional Points:** The response will contain the `ride_id` which you will need for other actions like canceling or paying.

Cancel a Ride

Cancels a ride that is in a "requested" or "accepted" state.

- **Method:** POST
- **Endpoint:** /app/riders/cancel/:ride_id
- **Authentication:** Protected Endpoint
- **JSON Body:** None
- **Additional Points:** Replace `:ride_id` in the URL with the actual ID of the ride you want to cancel (e.g., /app/riders/cancel/1).

Make a Payment for a Ride

Submits payment information for a "completed" ride.

- **Method:** POST
- **Endpoint:** /app/riders/payment/:ride_id
- **Authentication:** Protected Endpoint

JSON Body:

```
{
  "payment_id": ,
  "fare": ,
  "mode": ""
}
```

Rate a Driver

Submits a rating for a driver after a ride is "completed".

- **Method:** POST
- **Endpoint:** /app/riders/rate/:ride_id
- **Authentication:** Protected Endpoint

JSON Body:

```
{
  "rate": ,
  "comment": ""
}
```

Driver Actions

These endpoints are for actions performed by users with the "driver" role.

Base Path: /app/driver

Register a Vehicle

Registers a new vehicle for the logged-in driver. A driver can only have one vehicle.

- **Method:** POST
- **Endpoint:** /app/driver/vehicle/register
- **Authentication:** Protected Endpoint

JSON Body:

```
{
  "model": "",
  "year": ,
  "reg_number": "",
  "plate_number": "",
  "color": ""
}
```

Update Vehicle Information

Updates the details of the driver's currently registered vehicle.

- **Method:** PATCH
- **Endpoint:** /app/driver/vehicle/update
- **Authentication:** Protected Endpoint

JSON Body:

```
{
  "model": "",
  "color": ""
}
```

Accept a Ride

Allows a driver to accept a pending ride request.

- **Method:** POST
- **Endpoint:** /app/driver/accept
- **Authentication:** Protected Endpoint

JSON Body:

```
{
  "ride_id": ""
}
```

- **Additional Points:**

- The `ride_id` is the ID of the ride requested by a rider. In a real-world scenario, this would be pushed to the driver. For testing, you can get this from the "Book a Ride" response.
- **SHOWING SAME ID.** The app correctly identifies the user based on the `authToken` cookie they have after logging in. When a ride is booked, the `rider_id` is set to the `userId` from the currently logged-in user's token. When a ride is accepted, the `driver_id` is set to the `userId` from the currently logged-in user's token.

Update Ride Status

Updates the status of an active ride.

- **Method:** PATCH
- **Endpoint:** `/app/driver/status`
- **Authentication:** Protected Endpoint

JSON Body:

```
{
  "ride_id": "",
  "status": ""
}
```

- **Additional Points:** Valid statuses for transition are: `ongoing`, `completed`.

Rate a Rider

Submits a rating for a rider after a ride is "completed".

- **Method:** POST
- **Endpoint:** `/app/driver/rate/rider`
- **Authentication:** Protected Endpoint

JSON Body:

```
{
  "ride_id": "",
  "rating": ,
  "comment": ""
}
```