

Online Booking System for Car Rental



Team Members: Aliev Khadis, Wang Haitao, Ritu Arora

2025-02-26



RentNDrive

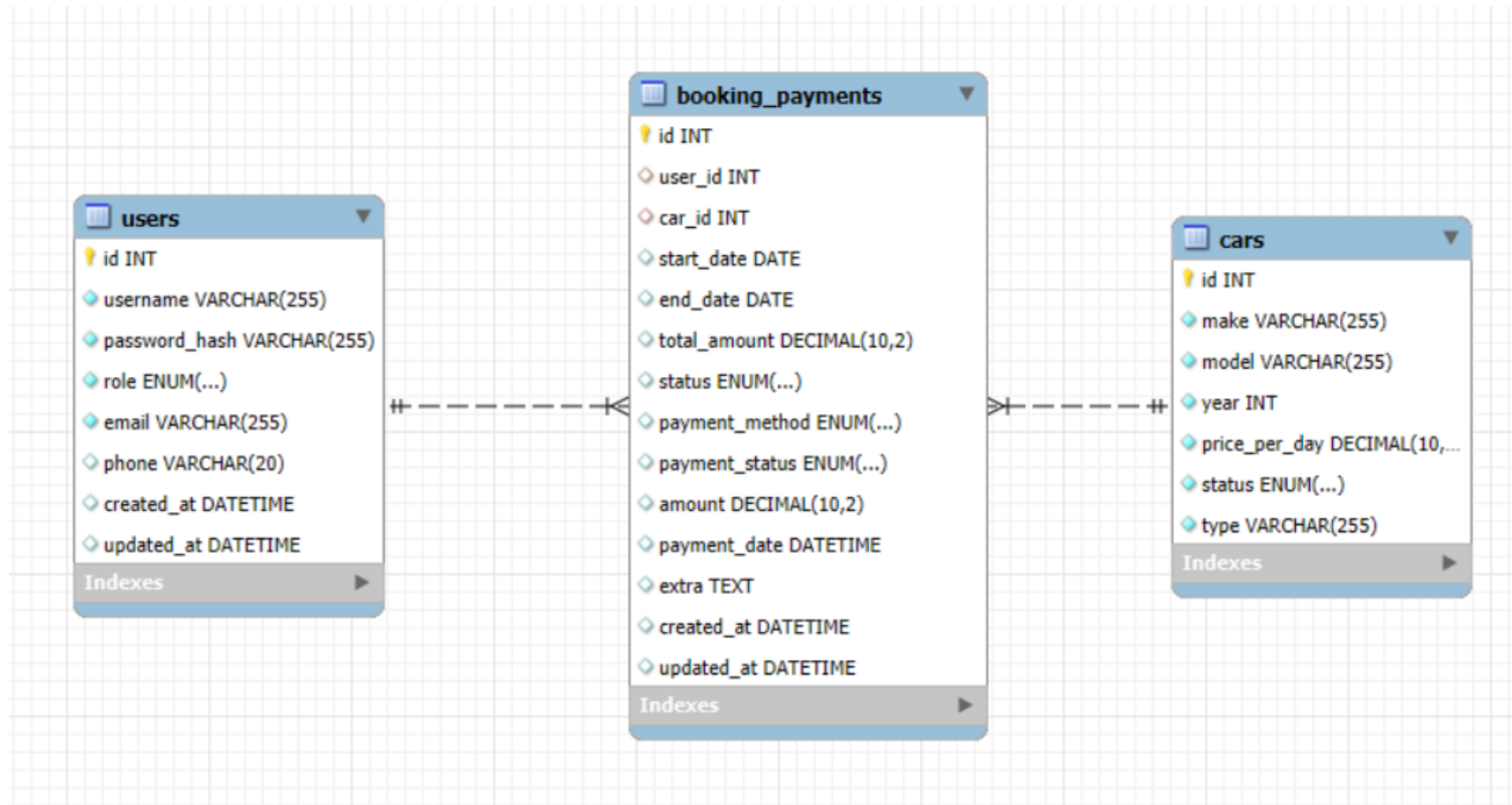
- Inefficient car rental booking and management processes.
- An online booking system that simplifies car rentals for customers and streamlines management for admins.
- Ease of use for customers.
- Efficient fleet management for admins.
- Real-time booking and availability tracking.

RentNDrive: Key Features

- Booking Wizard (step-by-step process)
- Car Availability Calendar (FullCalendar integration)
- User Authentication (role-based access)
- Admin Dashboard (fleet and booking management)



Entity-Relationship Diagrams



API Design Flow

- User Management: /api/login, /api/register, /api/users/{userId}
- Car Management: /api/cars (GET, POST, PUT, DELETE)
- Booking Management: /api/cars/available, /api/bookings
- Payments: /api/payments

Customer Logs In

Request: POST /api/login

Response: Receives JWT token for authentication.

Customer Searches for Cars

Request: GET /api/cars/available?start_date=YYYY-MM-DD&end_date=YYYY-MM-DD

Description: Searches for available cars for specific dates (you can add optional filters such as car type, location, etc., depending on your requirements).

Customer Makes a Payment

Request: POST /api/payments

Description: Customer completes payment for the booking.

Customer Makes a Booking

Request: POST /api/bookings

Description: Customer makes a booking for a car by providing booking details like the car, rental dates, etc.

API Design Flow



API Design Flow (contd..)



Admin Adds a New Car

Request: POST /api/cars

Description: Admin adds a new car to the fleet with details such as make, model, price, and availability.

Admin Reviews the Booking

Request: GET /api/bookings/{bookingId}

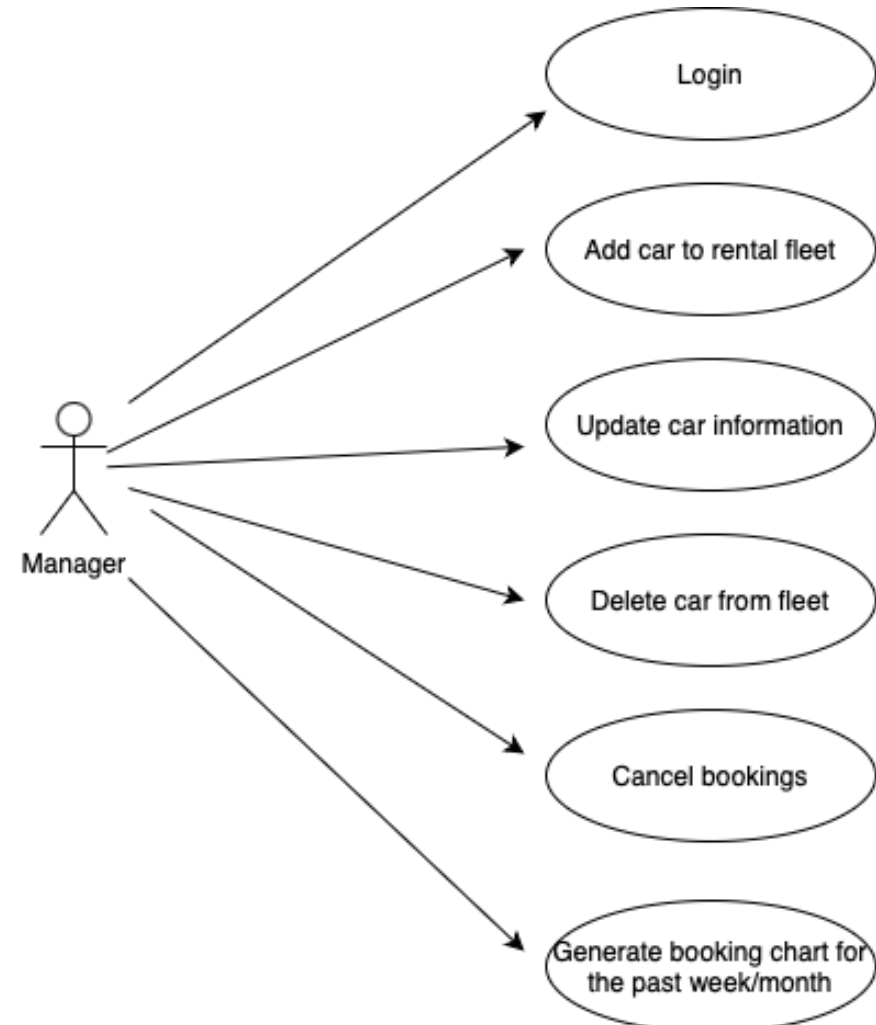
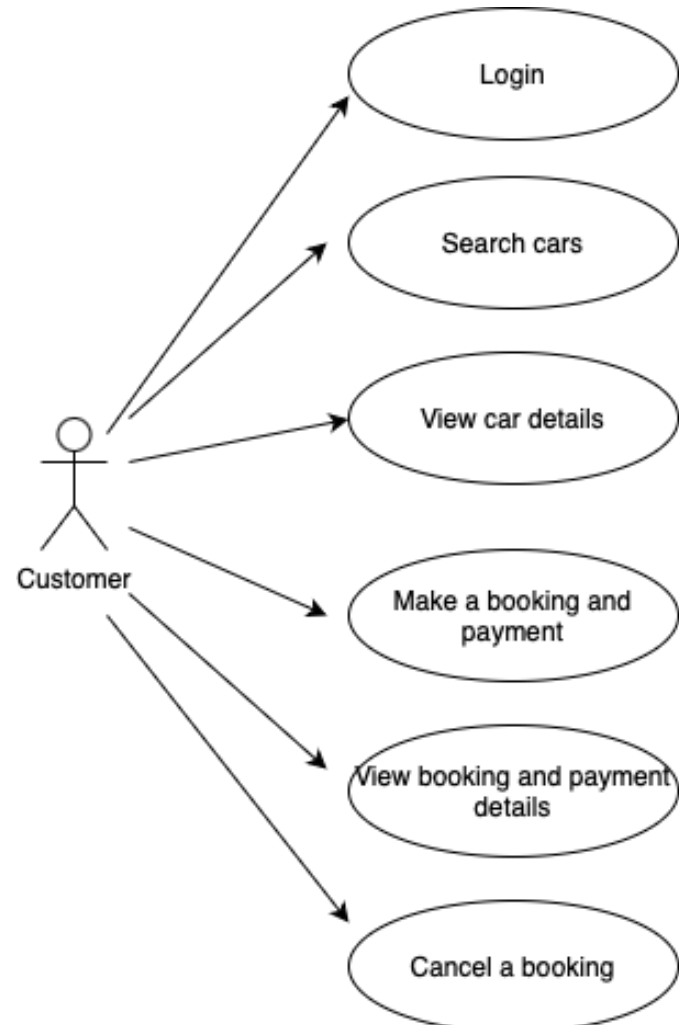
Description: Admin can review details for a specific booking.

Admin Tracks Car Availability

Request: GET /api/cars/available?start_date=YYYY-MM-DD&end_date=YYYY-MM-DD

Description: Admin tracks availability of cars (This could be part of the same /available API).

Case Diagram

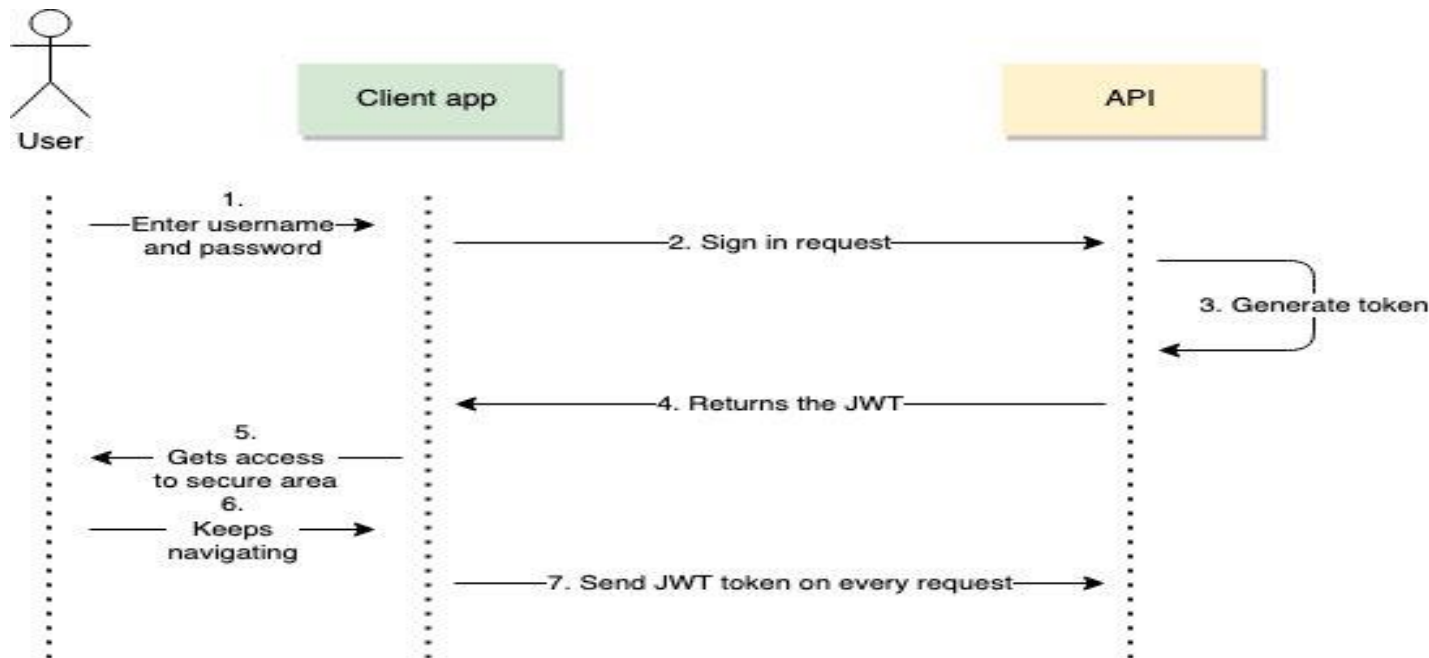


Authentication & Registration



JSON Web Tokens (JWT)

- A standard for securely transmitting information between parties as a JSON object.
- Used for authentication and authorization.
- Self-contained, carrying all necessary user information.



- Used for secure admin and user access.
- Role based access is included in the token.

Registering a New User

Endpoint: /api/register (POST)

Request:

- Client sends username, email, password, phone, and role (if applicable).
- Server validates input using express-validator.
- Checks for existing users (username or email).
- Hashes the password using bcryptjs.
- Stores user data in the users table.

Response:

- Success: "User registered successfully."
- Error: Validation errors or "Username/email already exists."



POST

http://localhost:3000/api/register

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cook

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

☒ JSON

Beauti

```
1 {
2   "username": "testuser",
3   "email": "testuser@example.com",
4   "password": "password123",
5   "phone": "123-456-7890",
6   "role": "customer"
7 }
```

Body

Cookies

Headers (8)

Test Results (0/1)

🕒

201 Created

327 ms

315 B

🌐

📄

Save Response

{ } JSON

▶ Preview

🔗 Visualize

🔍

📄

🔍

```
1 {
2   "message": "User registered successfully."
3 }
```

User Login and Token Generation

Endpoint: /api/login (POST)

Request:

- Client sends username and password.
- Server validates input.
- Checks if the user exists in the users table.
- Compares the provided password with the stored hash using bcryptjs.

Response:

- Success: JWT token, user role, username, and id.
- Error: "Invalid username or password."

```
// Example from auth.controller.js (login function)
```

```
const jwt = require("jsonwebtoken");  
const jwtConfig = require("../config/jwt");
```

```
// ... (rest of login function) ...
```

```
const token = jwt.sign(  
  { id: user.id, role: user.role }, // Payload: user ID and role  
  process.env.JWT_SECRET, // Secret key from environment variables  
  {  
    expiresIn: jwtConfig.expiresIn, // Token expiration time (e.g., '1h', '30m')  
    algorithm: jwtConfig.algorithm, // Algorithm used for signing (e.g., 'HS256')  
  }  
);
```

```
console.log("Generated Token:", token); // Log the token for debugging
```

```
// ... (send the token in the response) ...
```

```
res.json({ token, role: user.role, username: user.username, id: user.id });
```

JWT Generation:

- Uses jsonwebtoken to create the JWT.
- Includes user ID and role in the token payload.
- Uses a secret key (JWT_SECRET) for signing.
- Sets an expiration time for the token.

Protecting Routes with JWT Verification

Middleware: authMiddleware.verifyToken

- Extracts the JWT from the Authorization header.
- Verifies the token's signature and expiration.
- Attaches user information to req.user.

Protected Routes:

- Admin routes (e.g., /api/cars) require admin role.
- User routes (e.g., /api/users/{userId}) require a valid token.
- Admin routes use router.use(protectAdminRoute) to protect all admin routes.
- The login and register routes are not protected.

// Example from middleware/auth.js

```
const jwt = require("jsonwebtoken");
const log = require("../logger");
```

Tabnine | Edit | Explain

```
const verifyToken = (req, res, next) => {
  const token = req.header("Authorization");

  if (!token) {
    return res.status(401).json({ message: "Access denied. No token provided." });
  }

  try {
    const decoded = jwt.verify(
      token.replace("Bearer ", ""),
      process.env.JWT_SECRET
    );
    req.user = decoded; // Attach user info to the request
    next(); // Proceed to the next middleware or route handler
  } catch (error) {
    log.warn(`Token verification failed: ${error.message}`);
    res.status(400).json({ message: "Invalid token." });
  }
};
```

```
module.exports = { verifyToken };
```


POST

http://localhost:3000/api/login

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Beautify

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "username": "testuser",
3   "password": "password123"
4 }
```

Body

Cookies

Headers (8)

Test Results (1/1)

200 OK

262 ms

491 B

Save Response

JSON

Preview

Visualize

1 {

2 "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTYsInVubGU1OiJkdXN0b211ciIsIm1hdCI6MTc0M0M2N0E1NiwiZXBwIjoxNzQwMzY3NzU2fQ.22i2o45wR5ed9z_ZpIDsSaM1jXC9usKG1x2sXJCEb64",

3 "role": "customer",

4 "username": "testuser",

5 "id": 16

6 }

Register

Register

Already have an account? [Login](#)

Start Date

2025/02/24

End Date

2025/02/24



Cars



SUVs



Minivans



Trucks



Vans



Boxtrucks

Car rental for any occasion

Browse an incredible selection of cars, from the everyday to the extraordinary.

[Explore cars](#)

- Dashboard
- Manage Cars
- Manage Bookings
- Manage Customers

15

16

15

Select	ID	User ID	Car ID	Start Date	End Date	Total Amount	Status	Payment Status	Created At	Updated At
<input type="radio"/>	16	1	1	2024-01-10	2024-01-15	\$250.00	confirmed	completed	2025-02-24	2025-02-24
<input type="radio"/>	17	2	2	2024-01-20	2024-01-25	\$225.00	confirmed	completed	2025-02-24	2025-02-24
<input type="radio"/>	18	3	3	2024-02-01	2024-02-07	\$480.00	confirmed	completed	2025-02-24	2025-02-24
<input type="radio"/>	19	4	4	2024-02-10	2024-02-15	\$350.00	pending	pending	2025-02-24	2025-02-24
<input type="radio"/>	20	5	5	2024-02-20	2024-02-28	\$720.00	confirmed	completed	2025-02-24	2025-02-24
<input type="radio"/>	21	6	6	2024-03-01	2024-03-05	\$340.00	confirmed	completed	2025-02-24	2025-02-24
<input type="radio"/>	22	7	7	2024-03-10	2024-03-17	\$525.00	pending	pending	2025-02-24	2025-02-24
<input type="radio"/>	23	8	8	2024-03-20	2024-03-25	\$275.00	confirmed	completed	2025-02-24	2025-02-24
<input type="radio"/>	24	9	9	2024-04-01	2024-04-08	\$420.00	confirmed	completed	2025-02-24	2025-02-24

Make ▼

Year ▼

Type ▼

Price ▼

Apply



Tesla Model 3 2022



BMW X5 2022



Mercedes C-Class 2023



Ford Mustang 2023



POST



http://localhost:3000/api/bookings

Send

Params

Authorization

Headers (9)

Body ●

Scripts ●

Settings

Coo

☐ none☐ form-data☐ x-www-form-urlencoded☒ raw☐ binary☐ GraphQL

JSON ▾

Beau

```
1  {
2    "carId": 1, // Replace with actual car ID
3    "startDate": "2024-01-10",
4    "endDate": "2024-01-15",
5    "paymentMethod": "credit_card",
6    "amount": 330, // Calculate amount based on price_per_day and days
7    "stripeToken": "tok_visa" // Replace with a test token from stripe.
8  }
```

Body

Cookies

Headers (8)

Test Results (0/1)



201 Created

• 1.28 s

• 669 B



Save Response

{ } JSON ▾

▶ Preview



Visualize



```
2    "id": 1,
3    "user_id": 16,
4    "car_id": 1,
5    "start_date": "2024-01-10",
6    "end_date": "2024-01-15",
7    "total_amount": 330,
8    "status": "confirmed",
9    "payment_method": "credit_card",
10   "payment_status": "completed",
11   "amount": 330,
12   "created_at": "2025-02-24T04:48:27.026Z",
13   "updated_at": "2025-02-24T04:48:27.026Z",
14   "updatedAt": "2025-02-24T04:48:27.078Z",
15   "createdAt": "2025-02-24T04:48:27.029Z",
16   "payment_date": "2025-02-24T04:48:27.078Z"
17 }
```

Postbot

▶ Runner

🔗 Start Proxy


🍪 Cookies

🏠 Vault



Booking and payment Details

Car Information

 Car Image

Make:

Year:

Type:

Price per Day:

Start Date:

End Date:

Extra:

Total Price: \$0

Payment Information

Cardholder's Name:

Card Number:

Expiry Date:

CVV:

Make a Booking & Pay

Challenges

- **Test Fragility & Maintenance**

Tests break easily with small code modifications, creating additional maintenance overhead.

- **Endless Bug Fixing & Retesting**

Frequent bug generation after every code change leads to constant retesting and fixes.

Contribution from each Team Member

Khadis Aliev

- Client-side - Bookings

Haitao Wang

- Server-side – Admin dashboard
- Client-side – Admin and customer dashboards, and Payments

Ritu Arora

- Cloud Database management
- Server-side – User Registration and Authentication, Bookings and payments
- Client-side - User Registration and Authentication
- Guided team members and managed ScrumBoard and Git repository



Thank you

POST

▼

http://localhost:8080/api/register

Send

▼

Params Authorization Headers (9) Body Scripts ● Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼

```
1 {
2   "username": "inuser",
3   "email": "inuser@example.com",
4   "password": "password123",
5   "phone": "123-456-7890",
6   "role": "customer"
7 }
```

Body Cookies Headers (8) Test Results (0/1) ↺

201 Created • 446 ms • 315 B • 🌐 | e.g. Save Response

{ } JSON ▼ ▶ Preview 🔄 Visualize ▼

☰ 📄 🔍 🔗

```
1 {
2   "message": "User registered successfully."
3 }
```

▼

Send

▼

Cookies

8 hidden

	Key	Value	Description	⋮ Bulk Edit Presets ▾
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI...		
	Key	Value	Description	

200 OK • 234 ms • 485 B •  |  Save Response 



```
1 {
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MjIsInJvbmUiOiJhZG1pbiIsIm1hdCI6MTc0MDU5NDg5MywiZXhwIjoxNzQwNTk4NDkzfQ.
   J42DbZv_Zz2VANcuqErmSQv7la-xvbpU26-TxaPzzQE",
3   "role": "admin",
4   "username": "testadmin",
5   "id": 22
6 }
```

GET

▼

http://localhost:8080/api/users/22

Send

▼

Params

Authorization

Headers (9)

Body ●

Scripts ●

Settings

Cookies

Headers

8 hidden

	Key	Value	Description	...	Bulk Edit	Presets ▼
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MjI...				
	Key	Value	Description			

Body

Cookies

Headers (8)

Test Results (1/1)

200 OK

151 ms

450 B

Save Response

{}

JSON ▼

▶ Preview

🔗 Visualize ▼

```
1 {
2   "id": 22,
3   "username": "testadmin",
4   "role": "admin",
5   "email": "testadmin@example.com",
6   "phone": "123-000-7890",
7   "created_at": "2025-02-25T22:04:00.000Z",
8   "updated_at": "2025-02-25T22:04:00.000Z"
9 }
```

GET http://localhost:8080/api/admin/cars

Parameters Authorization Headers (9) **Body** Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

```
1 {  
2   "username": "testadmin",  
3   "password": "password123"  
4 }
```

Body Cookies Headers (8) Test Results (1/1) ↺

200 OK

{ } **JSON** ▶ Preview 📄 Visualize

```
1 [  
2   {  
3     "id": 1,  
4     "make": "Toyota",  
5     "model": "Camry",  
6     "year": 2022,  
7     "price_per_day": "50.00",  
8     "status": "available",  
9     "type": "Sedan"  
10  },  
11  {  
12    "id": 2,  
13    "make": "Honda",  
14    "model": "Civic",  
15    "year": 2021,  
16    "price_per_day": "45.00",  
17    "status": "booked",  
18    "type": "Sedan"  
19  },  
20 ]
```


POST



http://localhost:8080/api/admin/cars

Send

Params Authorization Headers (9) **Body** Scripts Settings

Cc

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

Bea

```
1 {
2   "make": "Toyota",
3   "model": "Camry",
4   "year": "2022",
5   "price_per_day": 50.00,
6   "status": "available",
```

Body Cookies Headers (8) Test Results (0/1)

201 Created • 161 ms • 427 B • Save Response

{ } JSON Preview Visualize

```
1 {
2   "message": "Car added successfully.",
3   "car": {
4     "id": 17,
5     "make": "Toyota",
6     "model": "Camry",
7     "year": "2022",
8     "price_per_day": 50,
9     "status": "available",
10    "type": "Sedan"
11  }
12 }
```

PUT



http://localhost:8080/api/admin/cars/17

Send

Params Authorization Headers (9) **Body** Scripts Settings☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

B

```
1  {}
2  |  "price_per_day": 55.00
3  |  {}
4
5
```

Body Cookies Headers (8) Test Results (1/1)

200 OK • 277 ms • 422 B • Save Respo

{} JSON Preview Visualize 

```
1  {}
2  |  "message": "Car updated successfully.",
3  |  "car": {
4  |    "id": 17,
5  |    "make": "Toyota",
6  |    "model": "Camry",
7  |    "year": 2022,
8  |    "price_per_day": 55,
9  |    "status": "available",
10 |    "type": "Sedan"
11 |  }
12 }
```

DELETE

⌵

http://localhost:8080/api/admin/cars/17

Send

⌵

Params Authorization Headers (9) Body ● Scripts ● Settings

Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ⌵

Beautify

```
1 {
2   | "price_per_day": 55.00
3   | }
4
5
```

Body Cookies Headers (8) Test Results (1/1) ↺

200 OK • 192 ms • 306 B • 🌐 | e.g. Save Response ⋮

{ } JSON ⌵ ▶ Preview 🔄 Visualize ⌵

☰ 📄 🔍 🔗

```
1 {
2   | "message": "Car deleted successfully."
3   | }
```

POST



http://localhost:8080/api/booking/bookings

Send

Params Authorization Headers (9) Body ● Scripts ● Settings

Cooki

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▾

Beautif

```
1  {
2    "carId": 15, // Replace with actual car ID
3    "startDate": "2024-01-10",
4    "endDate": "2024-01-15",
5    "paymentMethod": "credit_card",
6    "amount": 300, // Calculate amount based on price_per_day and days
7    "stripeToken": "tok_visa" // Replace with a test token from stripe.
8  }
```

Body Cookies Headers (8) Test Results (0/1) 400 Bad Request • 217 ms • 339 B • | Save Response

{ } JSON ▾ ▶ Preview Visualize ▾

```
1  {
2    "message": "Calculated amount does not match provided amount."
3  }
```


GET

http://localhost:8080/api/booking/cars/available?startDate=2024-01-10&endDate=2024-01-15

Send

Params

Authorization

Headers (9)

Body

Scripts

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

1

{

2

"carId": 1, // Replace with actual car ID

3

"startDate": "2025-07-10",

4

"endDate": "2025-07-15",

5

"paymentMethod": "credit_card",

6

"amount": 300, // Calculate amount based on price_per_day and days

7

"stripeToken": "tok_visa" // Replace with a test token from stripe.

8

}

Body

Cookies

Headers (8)

Test Results (1/1)

200 OK • 163 ms • 1.59 KB • Save Response

{ } JSON

Preview

Visualize

1

[

2

{

3

"id": 4,

4

"make": "Chevrolet",

5

"model": "Suburban",

6

"year": 2020,

7

"price_per_day": "70.00",

8

"status": "available",

9

"type": "SUV"

10

},

11

{

12

"id": 5,

13

"make": "BMW",

14

"model": "X5"

POST

http://localhost:8080/api/booking/bookings


Send

Params Authorization Headers (9) Body Scripts Settings

Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1  {
2    "carId": 1, // Replace with actual car ID
3    "startDate": "2025-07-10",
4    "endDate": "2025-07-15",
5    "paymentMethod": "credit_card",
6    "amount": 300, // Calculate amount based on price_per_day and days
7    "stripeToken": "tok_visa" // Replace with a test token from stripe.
8  }
```

Body Cookies Headers (8) Test Results (0/1) 201 Created • 363 ms • 592 B •  { } JSON

```
1  {
2    "id": 33,
3    "user_id": 16,
4    "car_id": 1,
5    "start_date": "2025-07-10",
6    "end_date": "2025-07-15",
7    "total_amount": 300,
8    "status": "confirmed",
9    "payment_method": "credit_card",
10   "payment_status": "completed",
```