

Car Rental API Documentation

Base URL (example):

`https://demo.tourcode.online/api`

Authentication: JWT (auth:api) required for all endpoints except **register** and **login**.

Header for protected endpoints:

Authorization: Bearer {JWT_TOKEN}

Accept: application/json

1. Authentication

1.1 Register

POST /auth/register

Description: Register a new user and get JWT token.

Request Body:

```
{  
  "name": "John Doe",  
  "email": "john@example.com",  
  "password": "password123"  
}
```

Response (200 OK):

```
{  
  "user": {  
    "id": 1,  
    "name": "John Doe",  
    "email": "john@example.com"  
  },  
  "token": "JWT_TOKEN_HERE"
```

```
}
```

1.2 Login

POST /auth/login

Description: Login and get JWT token.

Request Body:

```
{  
  "email": "john@example.com",  
  "password": "password123"  
}
```

Response (200 OK):

```
{  
  "user": {  
    "id": 1,  
    "name": "John Doe",  
    "email": "john@example.com"  
  },  
  "token": "JWT_TOKEN_HERE"  
}
```

Error Response (401 Unauthorized):

```
{  
  "error": "Unauthorized"  
}
```

1.3 Logout

POST /auth/logout 

Description: Logout and invalidate JWT token.


Headers:

Authorization: Bearer {JWT_TOKEN}

Response (200 OK):

```
{  
  "message": "Logged out"  
}
```

1.4 Get Current User

GET /me 

Description: Get authenticated user information.

Headers:

Authorization: Bearer {JWT_TOKEN}

Response (200 OK):

```
{  
  "id": 1,  
  "name": "John Doe",  
  "email": "john@example.com"  
}
```

2. Cars

Method Endpoint Description

GET /cars List all cars

GET /cars/{id} Get single car details

POST /cars Create new car

Method Endpoint Description

PUT /cars/{id} Update car

DELETE /cars/{id} Delete car

Example Car Response:

```
{  
  "id": 2,  
  "name": "Toyota Corolla",  
  "brand": "Toyota",  
  "model": "2023",  
  "price_per_day": 60,  
  "description": "Comfortable car for city drives",  
  "image": "https://example.com/car.jpg"  
}
```

3. Bookings

Method	Endpoint	Description
GET	/bookings	List all bookings
GET	/bookings/{id}	Get booking details
POST	/bookings	Create a booking
PUT	/bookings/{id}	Update booking
DELETE	/bookings/{id}	Delete booking

Example Booking Request:

```
{  
  "user_id": 1,  
  "car_id": 2,
```

```
"start_date": "2026-01-10",  
"end_date": "2026-01-15"  
}
```

Example Booking Response:

```
{  
  "id": 5,  
  "user_id": 1,  
  "car_id": 2,  
  "start_date": "2026-01-10",  
  "end_date": "2026-01-15",  
  "total_price": 300,  
  "status": "pending"  
}
```

4. Maintenances

Method	Endpoint	Description
GET	/maintenances	List all maintenances
GET	/maintenances/{id}	Get maintenance details
POST	/maintenances	Create maintenance
PUT	/maintenances/{id}	Update maintenance
DELETE	/maintenances/{id}	Delete maintenance

Example Response:

```
{  
  "id": 1,  
  "car_id": 2,
```

```
"maintenance_date": "2026-01-05",  
"notes": "Oil change",  
"status": "completed"  
}
```

★ 5. Reviews

Method	Endpoint	Description
GET	/reviews	List all reviews
GET	/reviews/{id}	Get review details
POST	/reviews	Create review
PUT	/reviews/{id}	Update review
DELETE	/reviews/{id}	Delete review

Example Review Request:

```
{  
  "user_id": 1,  
  "car_id": 2,  
  "rating": 5,  
  "comment": "Excellent car!"  
}
```

🏠 6. Payments

Method	Endpoint	Description
GET	/payments	List all payments
GET	/payments/{id}	Get payment details

Method	Endpoint	Description
POST	/payments	Create payment
PUT	/payments/{id}	Update payment
DELETE	/payments/{id}	Delete payment

Example Payment Request:

```
{  
  "booking_id": 5,  
  "amount": 300,  
  "method": "credit_card",  
  "status": "completed"  
}
```

7. Users

Method	Endpoint	Description
GET	/users	List all users
GET	/users/{id}	Get user details
PUT	/users/{id}	Update user
DELETE	/users/{id}	Delete user

Example Update Request:

```
{  
  "name": "Jane Doe",  
  "phone": "+201234567890"  
}
```

Notes

1. **Authentication:** JWT required for all endpoints except /auth/register and /auth/login.
2. **Authorization:** Consider implementing **role-based access** (admin vs customer).
3. **Error Handling:**

```
{  
  "error": "Resource not found"  
}
```

4. **Protected Resources:** Cars, Users, Bookings, Reviews, Payments, Maintenance — all should be protected.
5. **Business Rules:**
 - Booking total price = $(\text{end_date} - \text{start_date} + 1) \times \text{price_per_day}$.
 - Reviews can only be made after a completed booking.
 - Maintenance may block car availability.