**Car Management API Documentation**

**Base URL**

* Local Development: http://localhost:5001/api
* Production: [https://carapp-backend-czb0.onrender.com](https://carapp-backend-czb0.onrender.com/)/api

**Authentication**

This API uses JWT for authentication. Include the token in the Authorization header for requests that require authentication.

Authorization: Bearer <your\_token>

**Endpoints**

**User Routes**

**Register a New User**

* **URL:** /api/users/register
* **Method:** POST
* **Description:** Register a new user with an email and password.
* **Body Parameters:**
  + name (string): User's name.
  + email (string): User's email.
  + password (string): User's password.
* **Response:**
  + 201 Created: Returns the created user.
  + 400 Bad Request: Invalid input data.

**Login**

* **URL:** /api/users/login
* **Method:** POST
* **Description:** Logs in a user and returns a JWT token.
* **Body Parameters:**
  + email (string): User's email.
  + password (string): User's password.
* **Response:**
  + 200 OK: Returns an authentication token.
  + 401 Unauthorized: Invalid credentials.

**Car Routes**

**Get All Cars for Logged-in User**

* **URL:** /api/cars
* **Method:** GET
* **Description:** Retrieve all cars associated with the authenticated user.
* **Headers:** Requires Authorization header with Bearer token.
* **Response:**
  + 200 OK: List of cars owned by the user.
  + 401 Unauthorized: Invalid or missing token.

**Create a New Car**

* **URL:** /api/cars
* **Method:** POST
* **Description:** Creates a new car with multiple images.
* **Headers:** Requires Authorization header with Bearer token.
* **Body Parameters:**
  + title (string): Title of the car.
  + description (string): Description of the car.
  + tags (string): Comma-separated tags.
  + images (file[]): Array of images to upload (up to 10 files).
* **Response:**
  + 201 Created: Returns the created car object.
  + 400 Bad Request: Missing images or required data.
  + 500 Internal Server Error: Server error.

**Get a Car by ID**

* **URL:** /api/cars/{id}
* **Method:** GET
* **Description:** Retrieve a car by its ID.
* **Headers:** Requires Authorization header with Bearer token.
* **Path Parameters:**
  + id (string): The car ID.
* **Response:**
  + 200 OK: Car details.
  + 404 Not Found: Car not found.
  + 401 Unauthorized: Invalid or missing token.

**Search for Cars by Title**

* **URL:** /api/cars/search
* **Method:** GET
* **Description:** Search cars by title for the authenticated user.
* **Headers:** Requires Authorization header with Bearer token.
* **Query Parameters:**
  + keyword (string): Search keyword for the car title.
* **Response:**
  + 200 OK: List of cars matching the search.
  + 400 Bad Request: Missing search keyword.
  + 500 Internal Server Error: Server error.

**Update Car Information**

* **URL:** /api/cars/{id}
* **Method:** PUT
* **Description:** Updates a car's information, including images.
* **Headers:** Requires Authorization header with Bearer token.
* **Path Parameters:**
  + id (string): The car ID.
* **Body Parameters:**
  + title (string, optional): New title.
  + description (string, optional): New description.
  + tags (string, optional): New comma-separated tags.
  + images (file[], optional): New images (up to 10).
* **Response:**
  + 200 OK: Updated car details.
  + 404 Not Found: Car not found.
  + 500 Internal Server Error: Server error.

**Delete a Car**

* **URL:** /api/cars/{id}
* **Method:** DELETE
* **Description:** Deletes a car along with its associated images.
* **Headers:** Requires Authorization header with Bearer token.
* **Path Parameters:**
  + id (string): The car ID.
* **Response:**
  + 200 OK: Confirmation of deletion.
  + 404 Not Found: Car not found.
  + 500 Internal Server Error: Server error.

**Error Responses**

The API may return the following HTTP status codes:

* **200 OK** - The request was successful.
* **201 Created** - The resource was successfully created.
* **400 Bad Request** - The request was invalid or missing parameters.
* **401 Unauthorized** - Authentication failed.
* **404 Not Found** - The requested resource does not exist.
* **500 Internal Server Error** - An error occurred on the server.

**Example Requests**

**Create a New Car with Images**

POST /api/cars

Authorization: Bearer <your\_token>

Content-Type: multipart/form-data

{

"title": "My New Car",

"description": "A nice car",

"tags": "SUV, Family",

"images": [image files...]

}

**Search Cars by Title**

GET /api/cars/search?keyword=SUV

Authorization: Bearer <your\_token>

**Update Car Information**

http

PUT /api/cars/{id}

Authorization: Bearer <your\_token>

Content-Type: multipart/form-data

{

"title": "Updated Car Title",

"description": "Updated description",

"tags": "Luxury, Sedan",

"images": [new image files...]

}

**Delete a Car**

DELETE /api/cars/{id}

Authorization: Bearer <your\_token>