Table of Contents

Introduction

The Food Delivery API allows clients to interact with the food delivery platform's backend services. The API follows RESTful principles and uses JSON for request and response bodies.

Authentication

- **Method**: Token-based authentication using JSON Web Tokens (JWT).
- **Header**: `Authorization: Bearer <token>`

Note: Some endpoints may require authentication. Tokens are obtained by logging in and should be included in the `Authorization` header for protected routes.

Endpoints

1. Users

1.1 Register a New User

- **URL**: `/api/users/register`
- **Method**: `POST`

- **Auth Required**: No
- **Request Body**:

```
json
Copy code
{
  "name": "string",
  "email": "string",
  "phone_number": "string",
  "password": "string"
}
```

- **Response**:
  - **201 Created**

```
json
Copy code
{
  "id": "integer",
  "name": "string",
  "email": "string",
  "phone_number": "string",
  "created_at": "timestamp"
}
```

1.2 User Login

- **URL**: `/api/users/login`
- **Method**: `POST`
- **Auth Required**: No
- **Request Body**:

```
json
Copy code
{
  "email": "string",
  "password": "string"
}
```

- **Response**:
  - **200 OK**

```
json
Copy code
{
  "token": "string",
  "user": {
    "id": "integer",
    "name": "string",
    "email": "string",
    "phone_number": "string",
    "created_at": "timestamp"
  }
}
```

1.3 Get User Profile

- **URL**: `/api/users/profile`
- **Method**: `GET`
- **Auth Required**: Yes
- **Response**:
  - **200 OK**

```
json
Copy code
{
  "id": "integer",
  "name": "string",
  "email": "string",
  "phone_number": "string",
  "created_at": "timestamp",
  "addresses": [ /* Array of address objects */ ]
}
```

1.4 Update User Profile

- **URL**: `/api/users/profile`
- **Method**: `PUT`
- **Auth Required**: Yes
- **Request Body**:

```
json
Copy code
{
  "name": "string",
  "phone_number": "string",
  "password": "string" // Optional
}
```

- **Response**:
  - **200 OK**

```
json
Copy code
{
  "id": "integer",
  "name": "string",
  "email": "string",
  "phone_number": "string",
  "updated_at": "timestamp"
}
```

---

2. Restaurants

2.1 Get All Restaurants

- **URL**: `/api/restaurants`
- **Method**: `GET`
- **Auth Required**: No
- **Response**:
  - **200 OK**

```
json
Copy code
[
  {
    "id": "integer",
    "name": "string",
    "description": "string",
    "phone_number": "string",
    "email": "string",
    "created_at": "timestamp",
    "address": { /* Address object */ }
  },
  /* More restaurants */
]
```

2.2 Get Restaurant Details

- **URL**: `/api/restaurants/{restaurant_id}`
- **Method**: `GET`
- **Auth Required**: No
- **Response**:
  - **200 OK**

```
json
Copy code
{
  "id": "integer",
  "name": "string",
  "description": "string",
  "phone_number": "string",
  "email": "string",
  "created_at": "timestamp",
  "address": { /* Address object */ },
  "categories": [ /* Array of categories */ ],
  "menu_items": [ /* Array of menu items */ ]
}
```

---

3. Addresses

3.1 Add a New Address

- **URL**: `/api/addresses`
- **Method**: `POST`
- **Auth Required**: Yes
- **Request Body**:

```
json
Copy code
{
  "street": "string",
  "city": "string",
  "state": "string",
  "country": "string",
  "postal_code": "string",
  "latitude": "decimal",
  "longitude": "decimal"
}
```

- **Response**:
  - **201 Created**

```
json
Copy code
{
  "id": "integer",
  "user_id": "integer",
  "street": "string",
  "city": "string",
  "state": "string",
  "country": "string",
  "postal_code": "string",
  "latitude": "decimal",
  "longitude": "decimal"
}
```

3.2 Get User Addresses

- **URL**: `/api/addresses`
- **Method**: `GET`
- **Auth Required**: Yes
- **Response**:
  - **200 OK**

```
json
Copy code
[
  {
    "id": "integer",
    "street": "string",
    "city": "string",
```

```
    "state": "string",
    "country": "string",
    "postal_code": "string",
    "latitude": "decimal",
    "longitude": "decimal"
  },
  /* More addresses */
]
```

3.3 Update an Address

- **URL**: `/api/addresses/{address_id}`
- **Method**: `PUT`
- **Auth Required**: Yes
- **Request Body**:

```
json
Copy code
{
  "street": "string",
  "city": "string",
  "state": "string",
  "country": "string",
  "postal_code": "string",
  "latitude": "decimal",
  "longitude": "decimal"
}
```

- **Response**:
  - **200 OK**

```
json
Copy code
{
  "id": "integer",
  "street": "string",
  "city": "string",
  "state": "string",
  "country": "string",
  "postal_code": "string",
  "latitude": "decimal",
  "longitude": "decimal",
  "updated_at": "timestamp"
}
```

3.4 Delete an Address

- **URL**: `/api/addresses/{address_id}`
- **Method**: `DELETE`
- **Auth Required**: Yes
- **Response**:
  - **204 No Content**

---

4. Categories

4.1 Get Categories by Restaurant

- **URL**: `/api/restaurants/{restaurant_id}/categories`
- **Method**: `GET`
- **Auth Required**: No
- **Response**:
  - **200 OK**

```
json
Copy code
[
  {
    "id": "integer",
    "name": "string"
  },
  /* More categories */
]
```

---

5. Menu Items

5.1 Get Menu Items by Restaurant

- **URL**: `/api/restaurants/{restaurant_id}/menu_items`
- **Method**: `GET`
- **Auth Required**: No
- **Response**:
  - **200 OK**

```
json
Copy code
[
  {
    "id": "integer",
    "name": "string",
    "description": "string",
    "price": "decimal",
    "is_available": "boolean",
    "category": {
      "id": "integer",
      "name": "string"
    }
  },
  /* More menu items */
]
```

---

6. Orders

6.1 Place a New Order

- **URL**: `/api/orders`
- **Method**: `POST`
- **Auth Required**: Yes
- **Request Body**:

```
json
Copy code
{
  "restaurant_id": "integer",
  "delivery_address_id": "integer",
  "menu_items": [
    {
      "menu_item_id": "integer",
      "quantity": "integer"
    },
    /* More items */
  ]
}
```

- **Response**:
  - **201 Created**

```
json
Copy code
{
  "id": "integer",
  "user_id": "integer",
  "restaurant_id": "integer",
  "delivery_address_id": "integer",
  "order_status": "string",
  "total_amount": "decimal",
  "payment_status": "string",
  "created_at": "timestamp",
  "order_items": [ /* Array of order items */ ]
}
```

6.2 Get User Orders

- **URL**: `/api/orders`
- **Method**: `GET`
- **Auth Required**: Yes
- **Response**:
  - **200 OK**

```
json
Copy code
[
  {
    "id": "integer",
```

```
    "restaurant": {
      "id": "integer",
      "name": "string"
    },
    "order_status": "string",
    "total_amount": "decimal",
    "created_at": "timestamp"
  },
  /* More orders */
]
```

6.3 Get Order Details

- **URL**: `/api/orders/{order_id}`
- **Method**: `GET`
- **Auth Required**: Yes
- **Response**:
  - **200 OK**

```
json
Copy code
{
  "id": "integer",
  "restaurant": {
    "id": "integer",
    "name": "string"
  },
  "delivery_address": { /* Address object */ },
  "order_status": "string",
  "total_amount": "decimal",
  "payment_status": "string",
  "created_at": "timestamp",
  "order_items": [
    {
      "menu_item": {
        "id": "integer",
        "name": "string"
      },
      "quantity": "integer",
      "unit_price": "decimal",
      "total_price": "decimal"
    },
    /* More items */
  ],
  "delivery_assignment": { /* Delivery assignment object */ }
}
```

6.4 Cancel an Order

- **URL**: `/api/orders/{order_id}`
- **Method**: `DELETE`
- **Auth Required**: Yes
- **Response**:

- **200 OK**

```
json
Copy code
{
  "message": "Order canceled successfully."
}
```

---

7. Order Items

Note: Order items are typically managed through the Orders endpoint when placing or viewing orders.

---

8. Payments

8.1 Make a Payment

- **URL**: `/api/payments`
- **Method**: `POST`
- **Auth Required**: Yes
- **Request Body**:

```
json
Copy code
{
  "order_id": "integer",
  "payment_method": "string", // e.g., "Credit Card"
  "transaction_id": "string" // Optional, depending on payment gateway
}
```

- **Response**:
  - **201 Created**

```
json
Copy code
{
  "id": "integer",
  "order_id": "integer",
  "user_id": "integer",
  "amount": "decimal",
  "payment_method": "string",
  "payment_status": "string",
  "transaction_id": "string",
  "created_at": "timestamp"
}
```

8.2 Get Payment Details

- **URL**: `/api/payments/{payment_id}`
- **Method**: `GET`

- **Auth Required**: Yes
- **Response**:
  - **200 OK**

```
json
Copy code
{
  "id": "integer",
  "order_id": "integer",
  "amount": "decimal",
  "payment_method": "string",
  "payment_status": "string",
  "transaction_id": "string",
  "created_at": "timestamp"
}
```

---

9. Delivery Personnel

Note: Endpoints for delivery personnel are typically restricted to admin users or the personnel themselves.

9.1 Get Assigned Deliveries

- **URL**: `/api/delivery_personnel/assignments`
- **Method**: `GET`
- **Auth Required**: Yes (Delivery Personnel)
- **Response**:
  - **200 OK**

```
json
Copy code
[
  {
    "order_id": "integer",
    "assignment_status": "string",
    "assigned_at": "timestamp",
    "order_details": { /* Order object */ }
  },
  /* More assignments */
]
```

---

10. Delivery Assignments

Note: Typically managed internally or through admin interfaces.

10.1 Update Delivery Assignment Status

- **URL**: `/api/delivery_assignments/{assignment_id}`
- **Method**: `PUT`

- **Auth Required**: Yes (Delivery Personnel)
- **Request Body**:

```
json
Copy code
{
  "assignment_status": "string" // e.g., "In Transit", "Delivered"
}
```

- **Response**:
  - **200 OK**

```
json
Copy code
{
  "id": "integer",
  "order_id": "integer",
  "delivery_personnel_id": "integer",
  "assignment_status": "string",
  "updated_at": "timestamp"
}
```

---

11. Reviews

11.1 Submit a Review

- **URL**: `/api/reviews`
- **Method**: `POST`
- **Auth Required**: Yes
- **Request Body**:

```
json
Copy code
{
  "restaurant_id": "integer",
  "order_id": "integer",
  "rating": "integer", // 1 to 5
  "comments": "string"
}
```

- **Response**:
  - **201 Created**

```
json
Copy code
{
  "id": "integer",
  "user_id": "integer",
  "restaurant_id": "integer",
  "order_id": "integer",
  "rating": "integer",
  "comments": "string",
  "created_at": "timestamp"
```

```
}
```

11.2 Get Reviews for a Restaurant

- **URL**: `/api/restaurants/{restaurant_id}/reviews`
- **Method**: `GET`
- **Auth Required**: No
- **Response**:
  - **200 OK**

```
json
Copy code
[
  {
    "id": "integer",
    "user": {
      "id": "integer",
      "name": "string"
    },
    "rating": "integer",
    "comments": "string",
    "created_at": "timestamp"
  },
  /* More reviews */
]
```

Error Handling

- **Standard Error Response**:

```
json
Copy code
{
  "error": {
    "code": "string",
    "message": "string",
    "details": "string" // Optional
  }
}
```

- **Common Error Codes**:
  - `INVALID_REQUEST`: The request parameters are invalid.
  - `UNAUTHORIZED`: Authentication failed or missing.
  - `FORBIDDEN`: Access denied.
  - `NOT_FOUND`: Resource not found.
  - `INTERNAL_ERROR`: Server encountered an error.

Status Codes

- **200 OK**: The request was successful.
- **201 Created**: A new resource has been created.
- **204 No Content**: The request was successful but there's no content to return.
- **400 Bad Request**: The request could not be understood due to invalid syntax.
- **401 Unauthorized**: Authentication is required and has failed or has not been provided.
- **403 Forbidden**: The request was valid, but the server is refusing action.
- **404 Not Found**: The requested resource could not be found.
- **500 Internal Server Error**: An unexpected error occurred on the server.

Conclusion

This API documentation provides the necessary information to interact with the food delivery app's backend services. Developers can use this guide to implement client applications, integrate with the platform, or extend functionality.

Additional Notes

- **Data Formats**:
    - Dates and timestamps should be in ISO 8601 format.
    - Decimal numbers should use string representations to avoid precision loss.
- **Pagination**:
    - For endpoints returning lists, consider implementing pagination using query parameters like `?page=1&limit=20`.
- **Filtering and Sorting**:
    - Endpoints can support query parameters for filtering (e.g., `?status=Delivered`) and sorting (e.g., `?sort=created_at&order=desc`).
- **Versioning**:
    - Prefix endpoints with `/v1/` to support future API versions.
- **Rate Limiting**:
    - Implement rate limiting to prevent abuse. Inform clients of limits via response headers.

Example Workflow

1. **User Registration and Authentication**:
    - User registers via `/api/users/register`.
    - User logs in via `/api/users/login` and receives a JWT token.

2. **Browsing Restaurants and Menu Items**:
    - User retrieves restaurants via `/api/restaurants`.
    - User views a restaurant's menu via `/api/restaurants/{restaurant_id}/menu_items`.
3. **Placing an Order**:
    - User adds an address via `/api/addresses`.
    - User places an order via `/api/orders` with selected menu items.
4. **Payment**:
    - User makes a payment via `/api/payments`.
5. **Order Tracking**:
    - User checks order status via `/api/orders/{order_id}`.
6. **Delivery**:
    - Delivery personnel updates assignment status via `/api/delivery_assignments/{assignment_id}`.
7. **Review**:
    - After order completion, user submits a review via `/api/reviews`.