



# API Endpoints and details

## Code Docs

Aa Name
<a href="#">Overview</a>
API Endpoints and details
<a href="#">Untitled</a>



## ENVIRONMENT VAR SET:

token -

env - <https://foooddelivery-production-4779.up.railway.app>

### POST /auth/register

- Register a new user.

• Payload: `{ "name": "", "email": "", "password": "" }`

## Request payload

```
{  
  "name": "ata",  
  "email": "ata@yopmail.com",  
  "password": "pass123"  
}
```

## Table of Content

### ENVIRONMENT VAR SET:

[POST /auth/register](#)

Request payload

Response

CURL

[POST /auth/login](#)

Request payload

Response

CURL

### RESTAURANT CRUD

[POST /restaurant](#)

Request payload

Response

CURL

[GET /restaurant](#)

Response

CURL

# Response

```
{
  "message": "User registered successfully"
}
```

**GET**  
**restaurants**  
Response  
**PATCH**  
**/restaurant**  
Request  
payload  
Response

**ADD/UPDATE/GE  
MENU**

**POST**  
**/restaurant/{res**  
Request  
payload  
Response

**GET**  
**/restaurant**  
Response  
**CURL**  
**PATCH**  
**/menu/<item  
id>**

**Request  
Payload**  
**Response**  
**CURL**  
**DELIVERY  
AGENT**  
**REGISTRATION**

**POST**  
**/delivery/regis  
ter**  
Request  
Payload

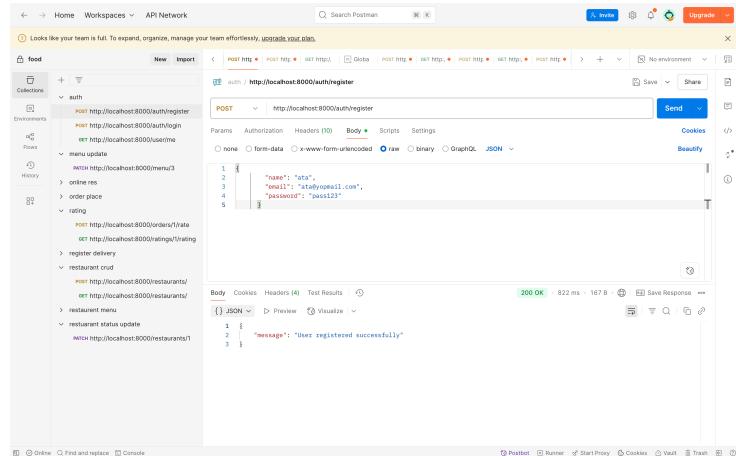
**Response**  
**CURL**  
**PLACE ORDER  
BY USER/  
UPDATING OF  
STATUS BY  
CUSTOMER/  
MARKING IT  
DELIVERED BY  
AGENT**

**POST /orders/**  
Request  
Payload

**Response**  
**CURL**  
**POST**  
**/orders/<or  
id>/status**  
Request  
Payload

# CURL

```
curl --location 'http://localhost:8000/auth/register' \
--header 'Content-Type: application/json' \
--data-raw '{
  "name": "ata",
  "email": "ata@yopmail.com",
  "password": "pass123"
}'
```



## POST /auth/login

- Login and receive JWT token.
- Payload: `{ "username": "", "password": "" }`
- Returns: `{ "access_token": "", "token_type": "bearer" }`

## Request payload

```
{  
  "email": "ag@yopmail.com",  
  "password": "string"  
}
```

## Response

```
{  
  "access_token": "eyJhbGciOiJIUzI1NilsInR5cCl6IkpXVC",  
  "token_type": "bearer"  
}
```

## CURL

```
curl --location 'http://localhost:8000/auth/login' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "email": "ag@yopmail.com",  
  "password": "string"  
}'
```

The screenshot shows the Postman interface with a POST request to `http://localhost:8000/auth/login`. The request body is set to `JSON` and contains the following JSON payload:

```
{  
  "email": "ag@yopmail.com",  
  "password": "string"  
}
```

The response status is `200 OK` with a response time of `562 ms` and a size of `286 B`. The response body is identical to the one shown in the previous code block.

# RESTAURANT CRUD

### POST /restaurant

- Create a restaurant.

## Request payload

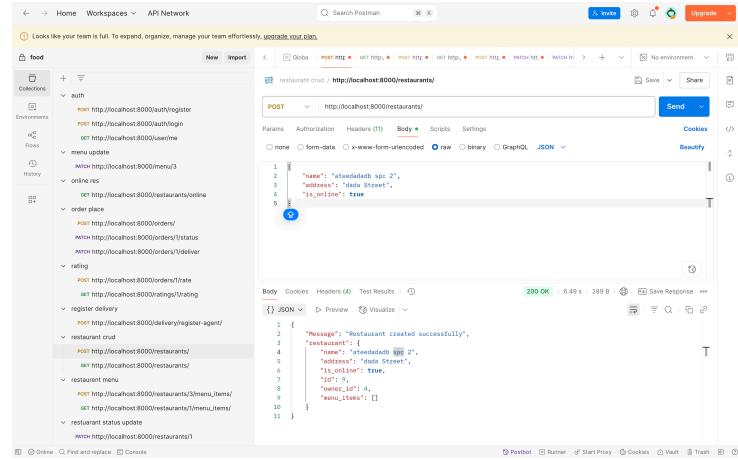
```
{  
    "name": "ateedadadb spc 2",  
    "address": "dada Street",  
    "is_online": true  
}
```

## Response

```
{  
    "Message": "Restaurant created successfully",  
    "restaurant": {  
        "name": "ateedadadb spc 2",  
        "address": "dada Street",  
        "is_online": true,  
        "id": 9,  
        "owner_id": 4,  
        "menu_items": []  
    }  
}
```

## CURL

```
curl --location 'http://localhost:8000/restaurants/' \  
--header 'Content-Type: application/json' \  
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NilsInR5c  
--data '{  
    "name": "ateedadadb spc 2",  
    "address": "dada Street",  
    "is_online": true  
}'
```



## GET /restaurant

- List all restaurants with their menus

## Response

```
[
  {
    "name": "sw 2",
    "address": "123 frnace Street",
    "is_online": true,
    "id": 1,
    "owner_id": 1,
    "menu_items": [
      {
        "name": "kachac piazza",
        "price": 13.5,
        "available": true,
        "id": 2
      },
      {
        "name": "addd piazza",
        "price": 11.5,
        "available": true,
        "id": 3
      },
      {
        "name": "kakka oadsta",
        "price": 249.99,
        "available": false,
        "id": 4
      }
    ]
  }
]
```

```

        "id": 1
    }
]
},
{
    "name": "ateeb spc 2",
    "address": "eng Street",
    "is_online": true,
    "id": 2,
    "owner_id": 1,
    "menu_items": []
}
]
```

## CURL

```
curl -H "Authorization: Bearer eyJhbGciOiJIUzI1NilsInR5cC..."
```

The screenshot shows the Postman interface with a collection named 'food'. A specific request is selected for the 'restaurant crud' endpoint at `/http://localhost:8000/restaurants/`. The method is set to 'GET'. In the 'Headers' tab, there is a single entry for 'Authorization' with the value 'Bearer eyJhbGciOiJIUzI1NilsInR5cC...'. The response status is 200 OK, and the response body is displayed in JSON format:

```

{
    "id": 2,
    "name": "ateeb spc 2",
    "address": "eng Street",
    "is_online": true,
    "menu_items": []
}

```

## GET restaurants/online

- Fetches all the restaurants that are online at the moment

## Reponse

```
[  
  {  
    "owner_id": 1,  
    "id": 1,  
    "address": "123 frnace Street",  
    "name": "sw 2",  
    "is_online": true  
  },  
  {  
    "owner_id": 1,  
    "id": 2,  
    "address": "eng Street",  
    "name": "ateeb spc 2",  
    "is_online": true  
  },  
  {  
    "owner_id": 2,  
    "id": 3,  
    "address": "eng Street",  
    "name": "ateeb spc 2",  
    "is_online": true  
  },  
  {  
    "owner_id": 4,  
    "id": 5,  
    "address": "sakak",  
    "name": "jsjs",  
    "is_online": true  
  },  
  {  
    "owner_id": 4,  
    "id": 6,  
    "address": "sakak",  
    "name": "jsjs",  
    "is_online": true  
  }  
]
```

```

[{"owner_id": 1, "id": 1, "name": "123 France Street", "address": "123 France Street"}, {"owner_id": 2, "id": 2, "name": "2nd Street", "address": "2nd Street"}, {"owner_id": 3, "id": 3, "name": "3rd Street", "address": "3rd Street"}]

```

## PATCH /restaurants/<restaurant\_id>

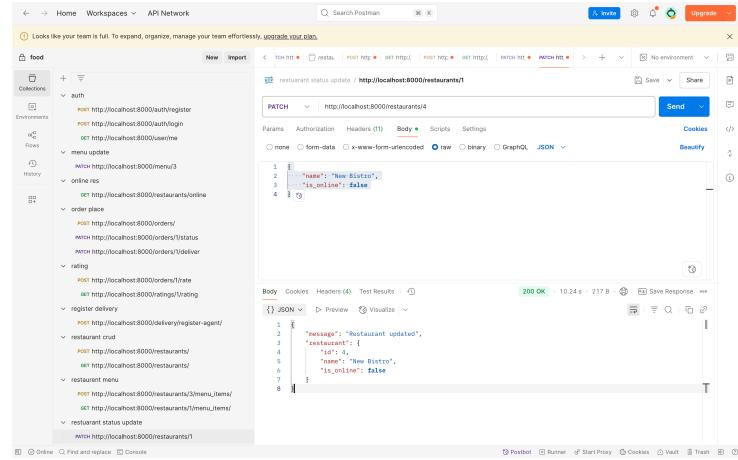
- Updates the name or the status of the particular restaurant.

## Request payload

```
{
  "name": "New Bistro",
  "is_online": false
}
```

## Response

```
{
  "message": "Restaurant updated",
  "restaurant": {
    "id": 4,
    "name": "New Bistro",
    "is_online": false
  }
}
```



## ADD/UPDATE/GET MENU

**POST /restaurant/{restaurant\_id}/menu**

- Add a menu item to a restaurant.

### Request payload

```
{
  "name": "pizza red",
  "price": 11.5,
  "available": true
}
```

### Response

```
{
  "name": "pizza ",
  "price": 11.5,
  "available": true,
  "id": 2
}
```

The screenshot shows the Postman interface with a collection named "food". A POST request is being made to `http://localhost:8000/restaurants/{restaurant_id}/menu_items/`. The Headers tab includes `Content-Type: application/json` and `Authorization: Bearer [token]`. The Body tab contains the following JSON:

```
{
  "name": "pizza",
  "price": 11.5,
  "available": true,
  "id": 2
}
```

The Response tab shows a 200 OK status with a response body containing one item:

```
[
  {
    "name": "pizza",
    "price": 11.5,
    "available": true,
    "id": 2
  }
]
```

## GET /restaurant/{restaurant\_id}/menu

- View menu items for a restaurant.

## Response

```
[
  {
    "name": "pizza",
    "price": 11.5,
    "available": true,
    "id": 2
  }
]
```

## CURL

```
curl --location 'https://foodelivery-production-4779.up.railway.app/restaurants/1/menu_items/' --header 'Authorization: Bearer eyJhbGciOiJIUzI1NilsInR5cC...'
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with collections like 'food' and 'auth'. The main area shows a 'GET' request to 'http://localhost:8000/restaurants/1/menu\_items/'. The 'Headers' tab is selected, showing an 'Authorization' header with a value of 'token'. The 'Body' tab shows a JSON response with one item:

```

{
  "id": 2,
  "name": "pizza",
  "price": 11.5,
  "available": true
}

```

### PATCH /menu/<item-id>

- Update the price and availability of item in a menu

## Request Payload

```
[
  {
    "name": "pizza",
    "price": 11.5,
    "available": true,
    "id": 2
  }
]
```

## Response

```
{
  "message": "Menu item updated",
  "menu_item": {
    "id": 2,
    "name": "pizza",
    "price": 249.99,
    "available": true
  }
}
```

## CURL

```
curl --location --request PATCH 'https://fooddelivery-proc
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5c
--header 'Content-Type: application/json' \
--data '{
  "price": 249.99,
  "available": true
}'
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with collections, environments, and various API endpoints listed under categories like auth, menu, online res, order place, rating, register delivery, restaurant crud, restaurant menu, and restaurant status update. The main workspace shows a PATCH request to 'http://localhost:8000/menu/2'. The 'Body' tab contains the JSON payload: { "price": 249.99, "available": true }. The 'Test Results' tab shows a successful response with status 200 OK, a duration of 203 ms, and a size of 323 B. The response body is displayed as JSON: { "message": "Menu item updated", "menu\_item": { "id": 2, "name": "Pasta", "price": 249.99, "available": true } }.

## DELIVERY AGENT REGISTRATION

POST /delivery/register-agent

- Authenticated user can register himself as delivery agent

### Request Payload

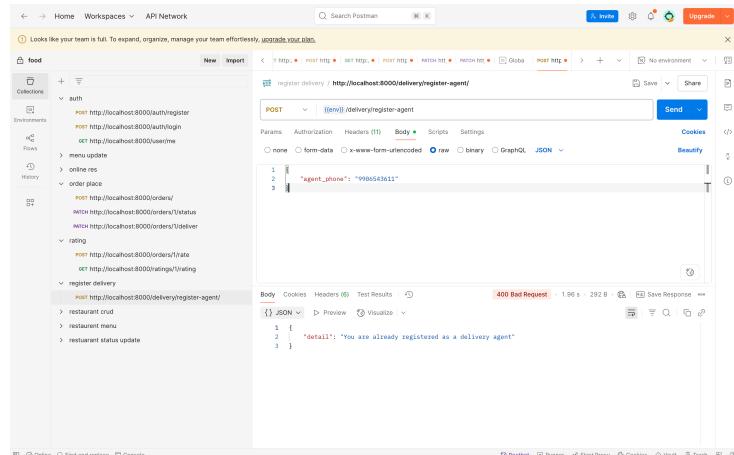
```
{  
  "agent_phone": "9906543611"  
}
```

# Response

```
{  
  "message": "You have been registered as a delivery agent"  
}
```

## CURL

```
curl --location 'https://foodelivery-production-4779.up.railway.app/delivery/register-agent/'  
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NilsInR5c  
--header 'Content-Type: application/json' \  
--data '{  
  "agent_phone": "9906543611"  
}'
```



## PLACE ORDER BY USER/ UPDATING OF STATUS BY CUSTOMER/ MARKING IT DELIVERED BY AGENT

**POST /orders/**

- Placing order for an item/items in menu of a restaurant

## Request Payload

```
{
  "restaurant_id": 2,
  "items": [
    {
      "menu_item_id": 2,
      "quantity": 1
    }
  ]
}
```

## Response

```
{
  "id": 1,
  "user_id": 6,
  "restaurant_id": 2,
  "status": "pending",
  "delivery_agent_id": 1,
  "items": [
    {
      "menu_item_id": 2,
      "quantity": 1,
      "price": 249.99
    }
  ],
  "created_at": "2025-06-06T04:35:40.361352Z"
}
```

## CURL

```
curl -H 'Content-Type: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NilsInR5cC...` \
-d '{
  "restaurant_id": 2,
  "items": [
    {
      "menu_item_id": 2,
      "quantity": 1
    }
  ]
}' https://foodelivery-production-4779.up.railway.app/orders
```

```

"quantity": 1
}
]
}
}

```

The screenshot shows the Postman interface with a collection named 'food'. A POST request is being made to `/orders`. The request body is set to `JSON` and contains the following data:

```

{
  "restaurant_id": 2,
  "items": [
    {
      "menu_item_id": 2,
      "quantity": 1
    }
  ]
}

```

The response status is `200 OK`.

### POST /orders/<order-id>/status

- Accepted or Reject an order by the owner

## Request Payload

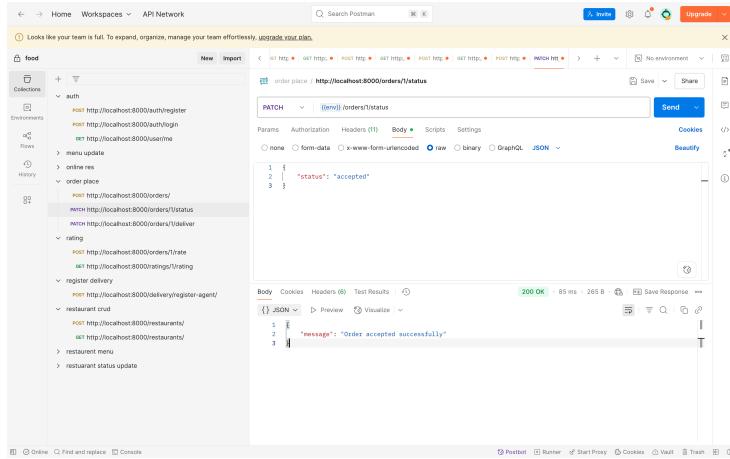
```
{
  "status": "accepted"
}
```

## Response

```
{
  "message": "Order accepted successfully"
}
```

## CURL

```
curl --location --request PATCH 'https://fooddelivery-proc
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5c
--data '{
  "status": "accepted"
}'
```



### POST /orders/<order-id>/deliver

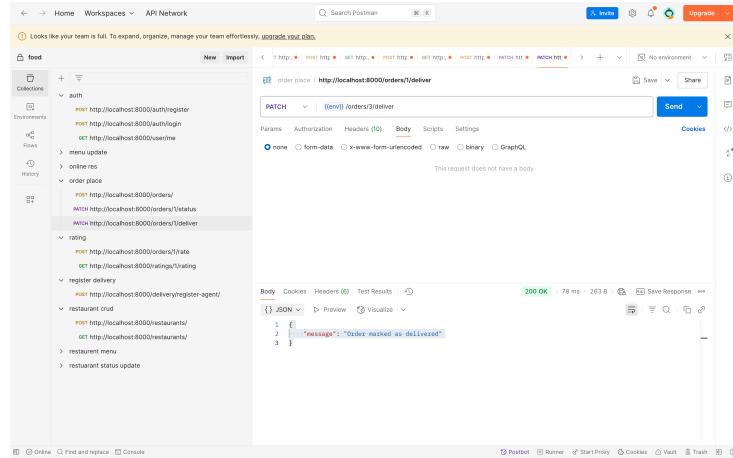
- Mark the the order as delivered by particular delivery agent

## Response

```
{
  "message": "Order marked as delivered"
}
```

## CURL

```
curl --location --request PATCH 'https://fooddelivery-proc
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5c
```



## ORDER RATING FOR ITEM/DELIVERY AGENT

**POST /orders/<order-id>/rate**

- Leave rating for an order and the delivery agent service

### Request payload

```
{
  "restaurant_rating": 4,
  "delivery_agent_rating": 5
}
```

### Response

```
{
  "message": "Thanks for your feedback!"
}
```

### CURL

```
curl --location 'https://foodelivery-production-4779.up.railway.app/orders/1/rate'
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cC'
--header 'Content-Type: application/json' \
```

```
--data '{
  "restaurant_rating": 4,
  "delivery_agent_rating": 5
}'
```

The screenshot shows the Postman application interface. A POST request is being made to `/localhost:8000/orders/3/rating`. The request body is set to `JSON` and contains the following data:

```
{
  "restaurant_rating": 4,
  "delivery_agent_rating": 5
}
```

The response status is `200 OK`, and the response body is:

```
{
  "message": "Thanks for your feedback!"
}
```

## POST /orders/<order-id>/rating

- Get the rating of order and delivery agent

## Request payload

```
{
  "restaurant_rating": 4,
  "delivery_agent_rating": 5
}
```

## Reponse

```
{
  "order_id": 3,
  "restaurant_rating": 4,
  "delivery_agent_rating": 5,
  "rated_by_user_id": 6,
  "rated_by_user_name": "new",
  "restaurant_name": "ateedadadb spc 2",
  "order_items": [
    ...
  ]
}
```

```

    "pizza "
]
}

```

## CURL

```

curl --location 'https://foodelivery-production-4779.up.railway.app/auth/login'
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NilsInR5cGkiXVC...

```

The screenshot shows the Postman application interface. On the left, there's a sidebar with a 'food' collection containing various API endpoints like auth, rating, and order place. The main area shows a single GET request to 'http://localhost:8000/ratings/rating'. The 'Headers' tab is selected, showing an 'Authorization' header with the value 'Bearer eyJhbGciOiJIUzI1NilsInR5cGkiXVC...'. Below the request, the response is displayed in JSON format:

```

{
  "id": 1,
  "customer_id": 1,
  "restaurant_rating": 4,
  "delivery_agent_rating": 5,
  "rate_by_driver": 4,
  "rate_by_user_driver": "no",
  "rate_by_user_reciever": "yes",
  "restaurant_name": "steedadadab spc 2",
  "order_items": [
    {
      "order_id": 1
    }
  ]
}

```