

RestoranKu - API

RestoranKu API

Below is the list of APIs used by frontend side, to get list of restaurants, create, edit, or delete a restaurant.

POST Login

`http://127.0.0.1:8000/api/login`

This is a POST request and it is used to login. There is need a request body for field "email" and "password"

A successful GET response will have a `200 OK` status, and should include some kind of response body - contains token and user information.

Body urlencoded

email	admin@restoranku.com
password	password

POST Register

`http://127.0.0.1:8000/api/register`

This is a POST request and it is used to register. There is need a request body for field "email", "password", "password_confirmation", and "name"

A successful POST response will have a `200 OK` status, and should include some kind of response body - for example, token and created user data.

HEADERS

Accept	application/json
--------	------------------

Body urlencoded

email	admin3@restoranku.com
password	password
password_confirmation	password
name	admin3

GET Get List Restoran

http://127.0.0.1:8000/api/restaurants

This is a GET request and it is used to "get" list of restaurant data from an endpoint. Also in this endpoint we can add filter by using query parameter like "?day=1" or "time=12:00" or "name=abc".

A successful GET response will have a `200 OK` status, and should include some kind of response body - in JSON format

GET Get data

http://127.0.0.1:8000/api/restaurants/2

This is a GET request and it is used to "get" a specific restaurant data from an endpoint.

A successful GET response will have a `200 OK` status, and should include some kind of response body - in JSON format.

POST Post data



http://127.0.0.1:8000/api/restaurants

This is a POST request, to create a new restaurant. In order to request to this endpoint, we need add "token" that get from login/register endpoint to the request authorization header like "Bearer Token". This request will submits JSON data of a new restaurant, and API will give back the new restaurant data as response.

A successful POST request typically returns a `200 OK` or `201 Created` response code.

AUTHORIZATION Bearer Token

Token	<token>
-------	---------

Body raw (json)

json

```
{
  "name": "Add your name in the body",
  "address": "address",
  "phone_number": "081234567",
  "note": "note",
  "open_times": [
    {"day_start": "1", "day_end": "7", "time_start": "07:00", "time_end": "19:00"}
  ]
}
```

PUT Update data



http://127.0.0.1:8000/api/restaurants/6



This is a PUT/PATCH request and it is used to overwrite an existing piece of restaurant data. In order to request to this endpoint, you need to add "token" that get from login/register endpoint to the request authorization header like "Bearer Token". And you typically identify the restaurant being updated by including an identifier in the URL (eg. `id=1`). This request will submit JSON data of the updated restaurant, and API will give back the updated restaurant data as response.

A successful PUT request typically returns a `200 OK`, `201 Created`, or `204 No Content` response code.

AUTHORIZATION Bearer Token

Token

<token>

Body raw (json)

json

```
{
  "name": "Add your name in the body",
  "address": "address",
  "phone_number": "081234567",
  "note": "note",
  "open_times": [
    {"day_start": "1", "day_end": "7", "time_start": "07:00", "time_end": "19:00"}
  ]
}
```

DELETE Delete data



`http://127.0.0.1:8000/api/restaurants/5`

This is a DELETE request, and it is used to delete restaurant data that was previously created via a POST request. In order to request to this endpoint, you need to add "token" that get from login/register endpoint to the request authorization header like "Bearer Token". And you typically identify the entity being updated by including an identifier in the URL (eg. `id=1`).

A successful DELETE request typically returns a `200 OK` , `202 Accepted` , or `204 No Content` response code.

AUTHORIZATION Bearer Token

Token	<token>
-------	---------