**API Documentation**

1. Installation

* Create database with name “db\_apijwt”.
* Migrate the database and seeder with this command:

“php artisan migrate --seed”

1. API Features

Base URL: **127.0.0.1:8000/api/auth/**

This API using JWT authentication and already test using postman. In this API there are user role, that is:

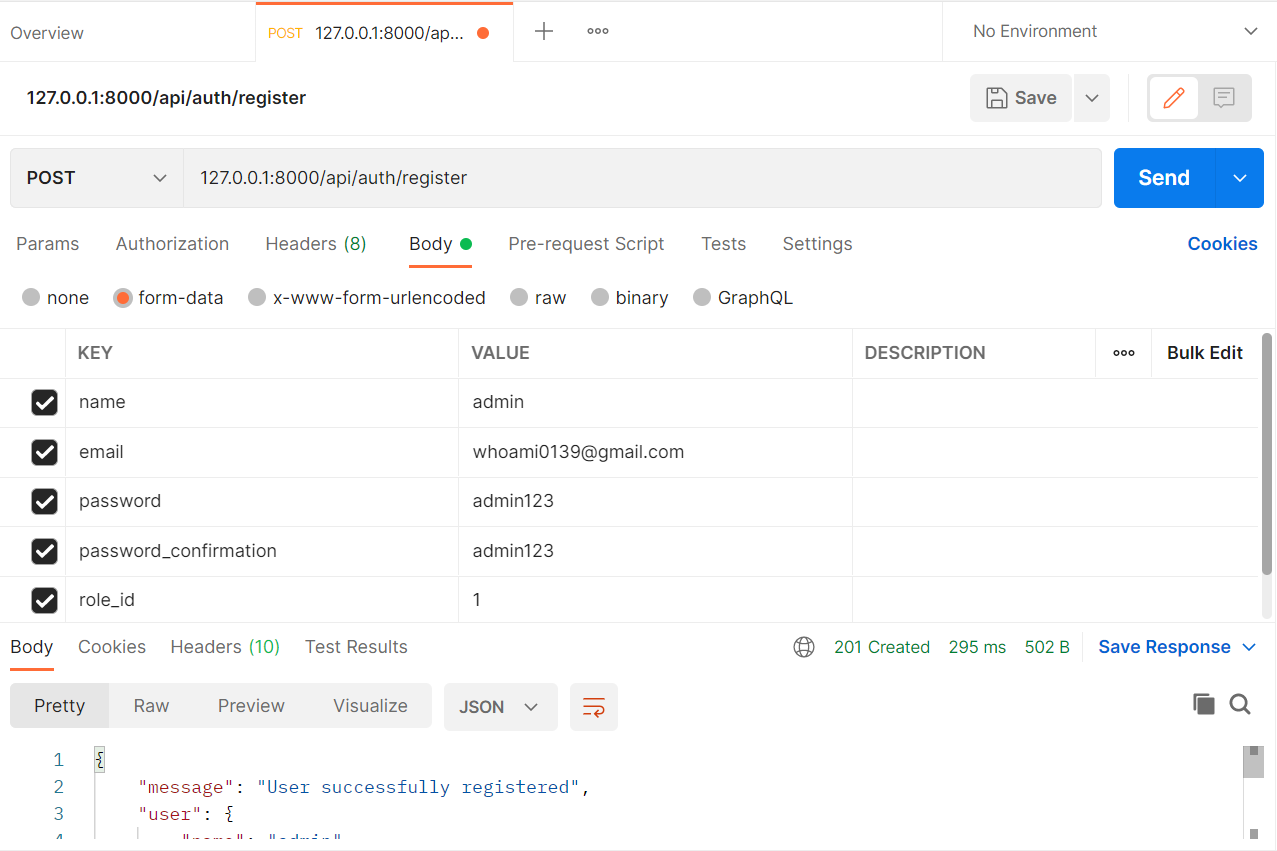
admin 🡺 1 (Role ID)

company 🡺 2

customer 🡺 3

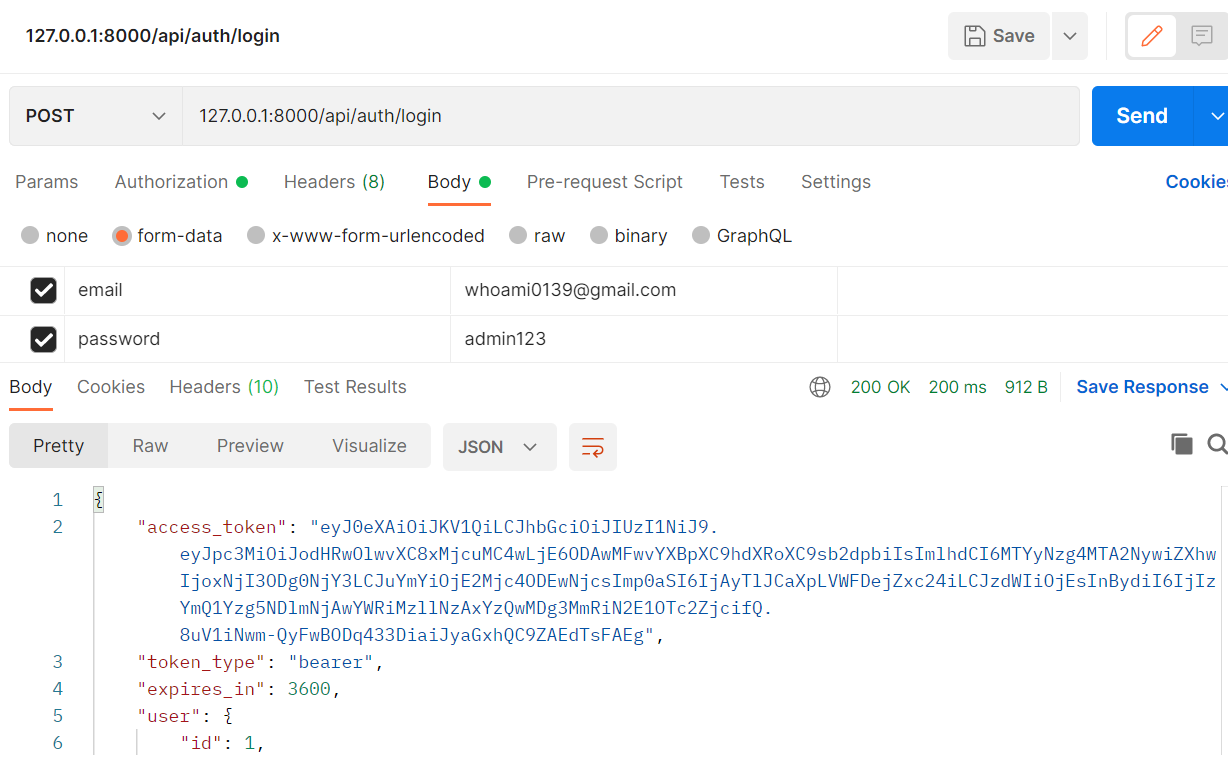
* **Register [ /register ]**

Before start to testing another API. First, create a user with role “admin” with role\_id “1”. Requred key [name, email, password, password\_confirmation, role\_id]. For example:



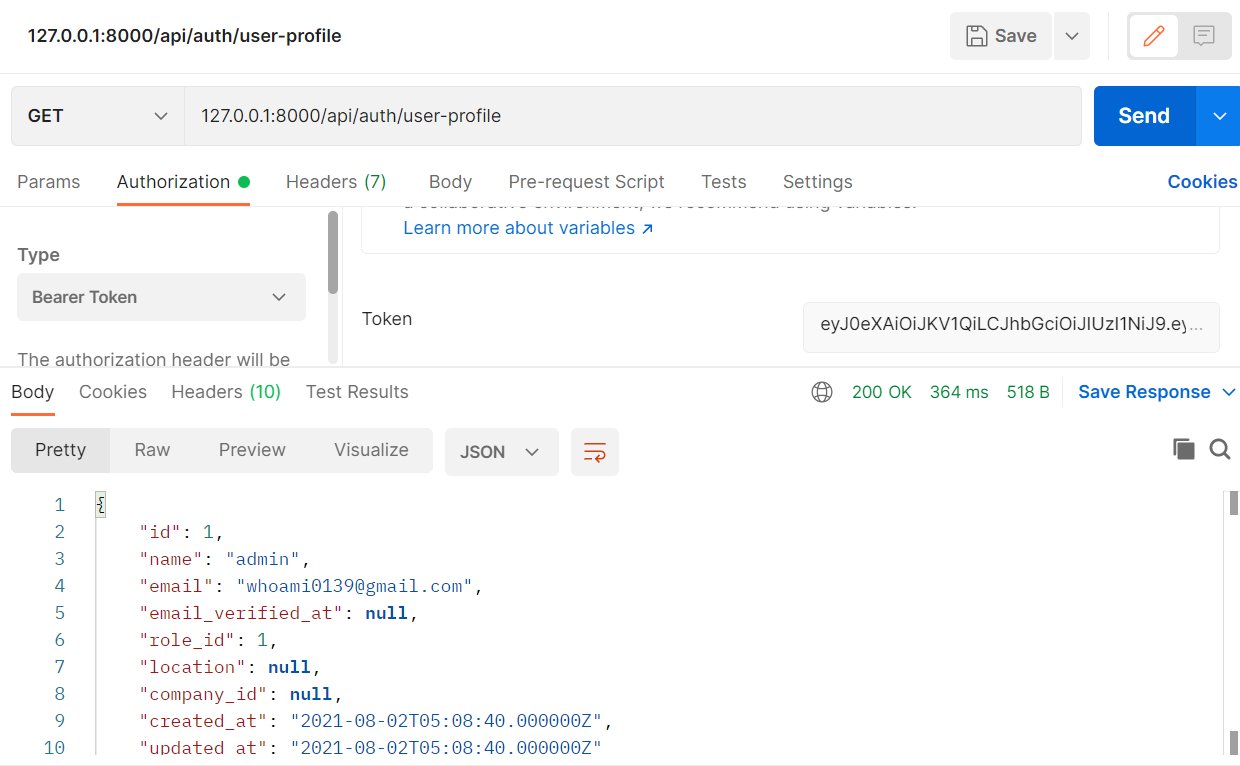
* **User Login [ /login ]**

After that, do login to get access\_token for other authentication. Required key [email, password]. For example:



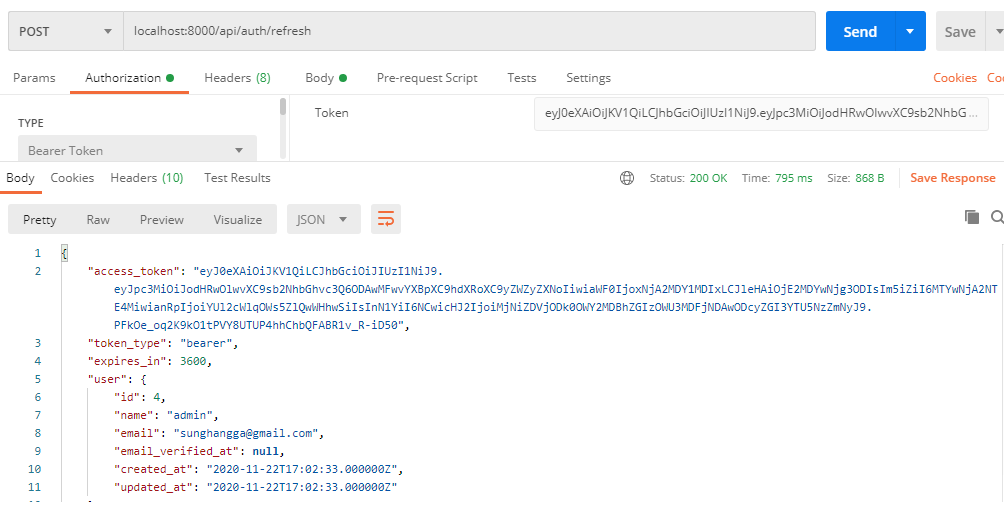
* **User Profile [ /user-profile ]**

To see the details of the logged in user, you can use **access\_token** in User Login API and put that into bearer token in User Profile API. No required key.



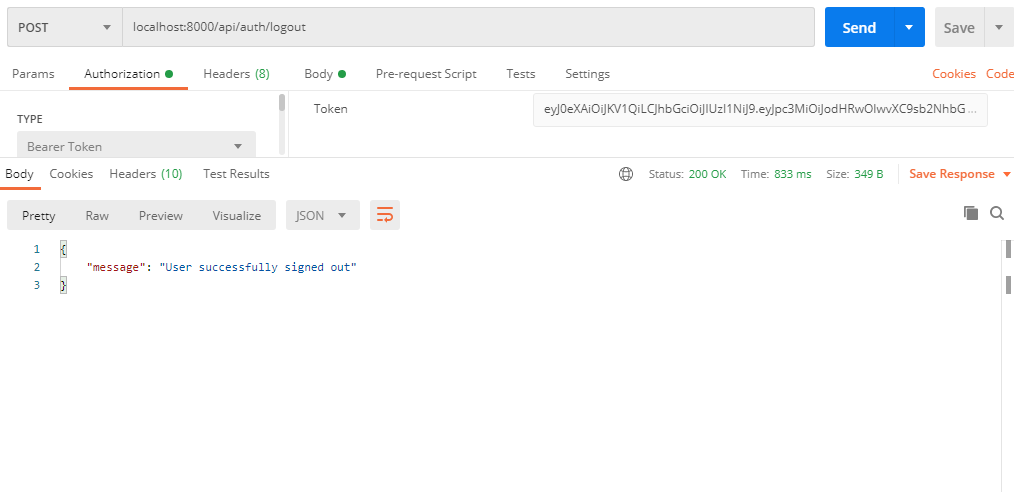
* **Refresh [ /refresh ]**

To get new token, it can be done by doing a refresh token with following API:



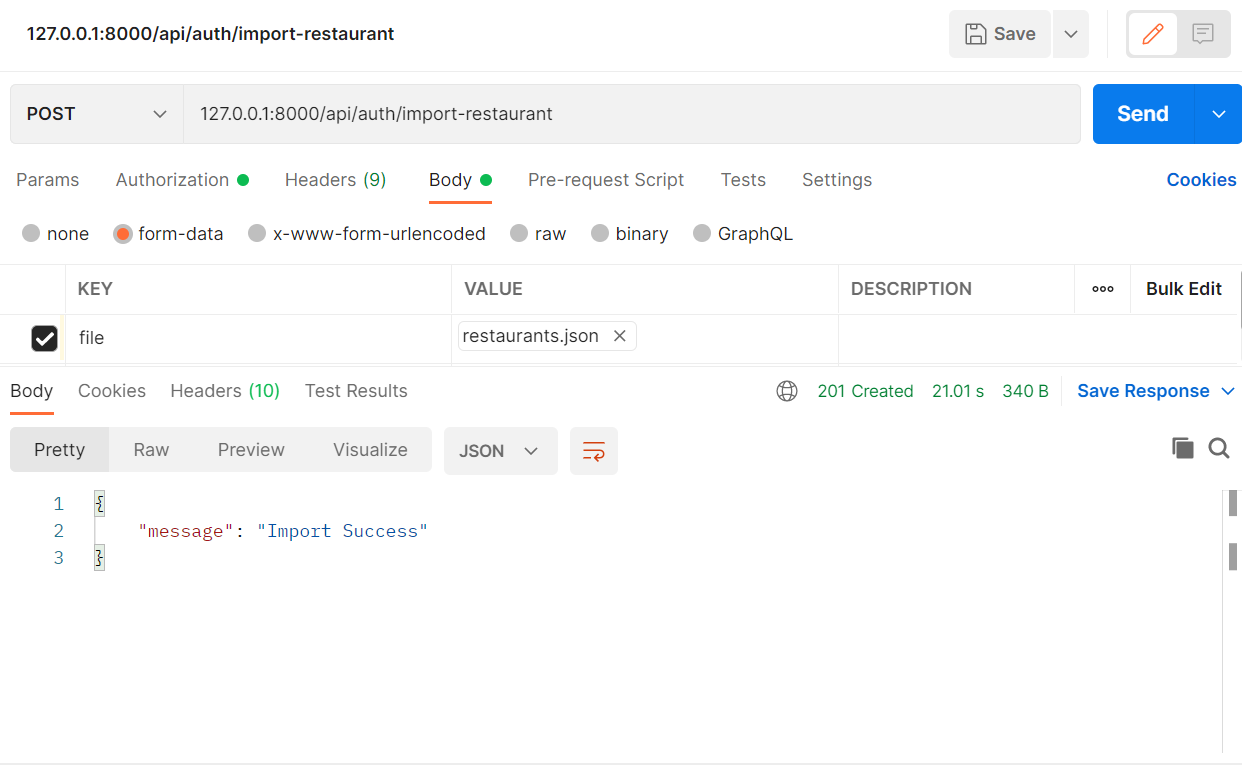
* **Logout [ /logout ]**

To logout from the system use following API:



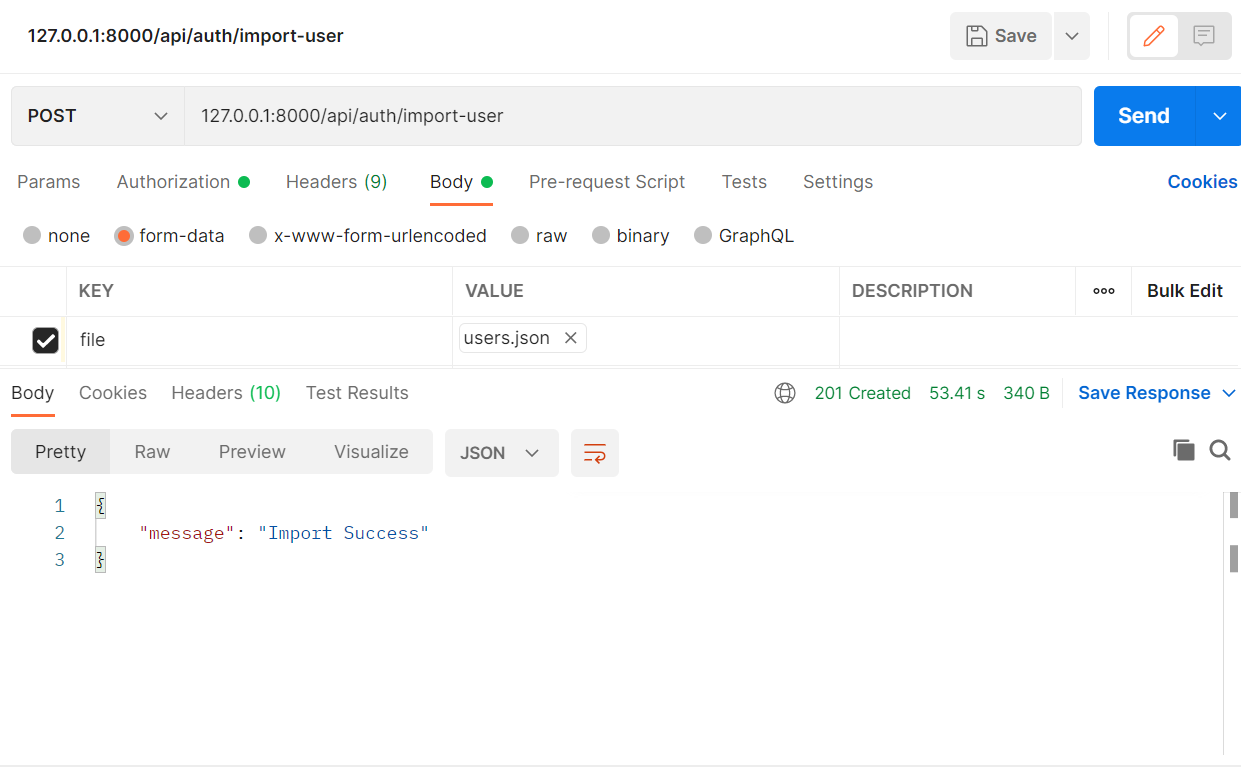
* **Import Restaurant JSON Data [ / import-restaurant ]**

To import data, you must login with the **admin role** and use bearer token, then select restaurant JSON file. Required key [file]. After import success, the result will look like below. This process may take some time.



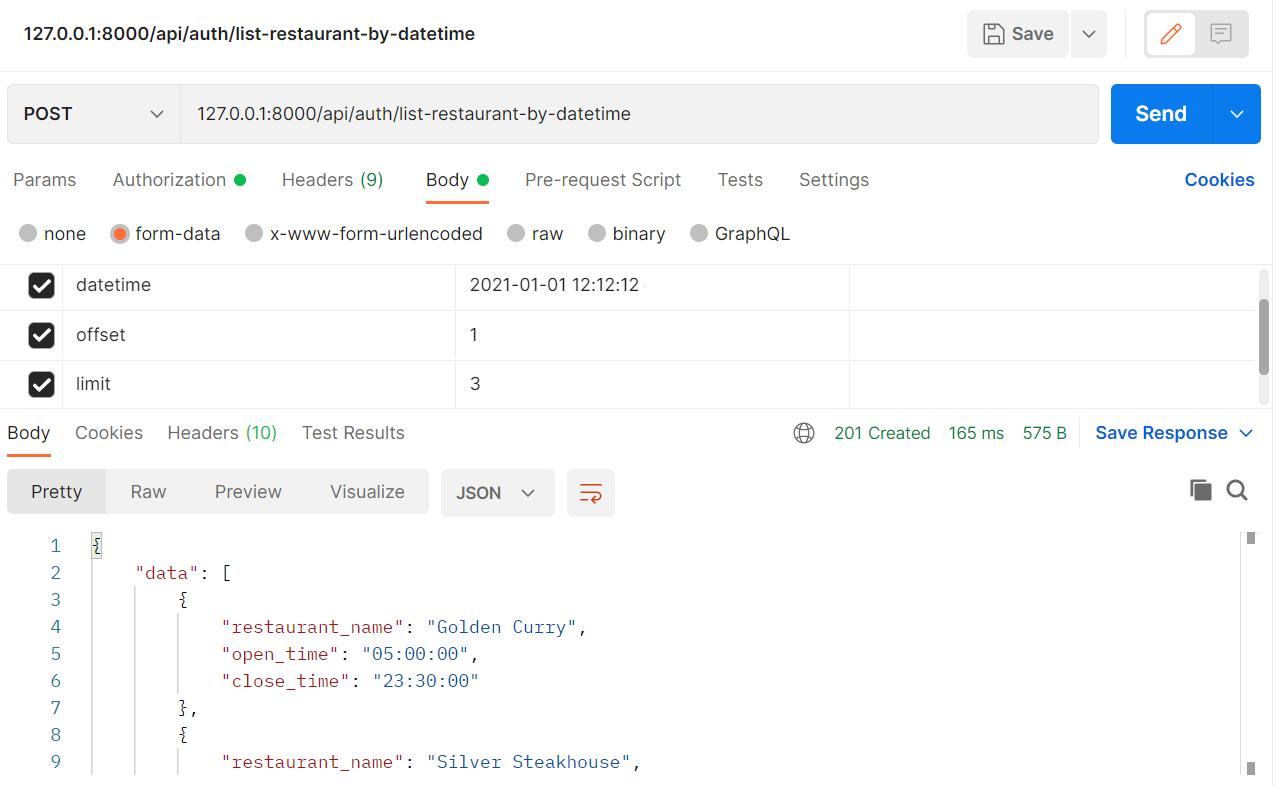
* **Import Users JSON Data [ / import-user ]**

To import data, you must login with the **admin role** and use bearer token, then select restaurant JSON file. Required key [file]. After import success, the result will look like below. This process may take some time.



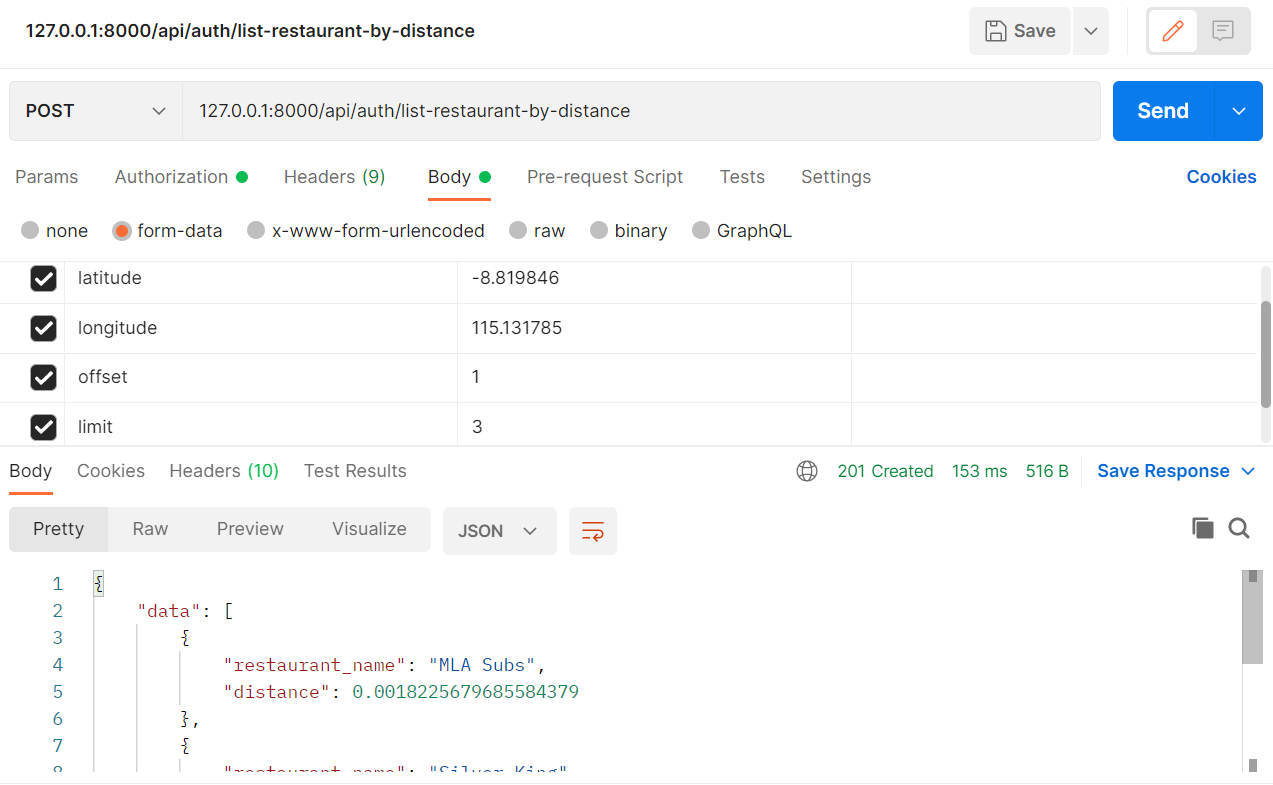
* **List all restaurants that are open at a certain datetime [ /list-restaurant-by-datetime ]**

This API can be used to see if a restaurant is open or not at a certain time. This API can be accessed on the **customer role**. Required key [datetime, offset, limit].



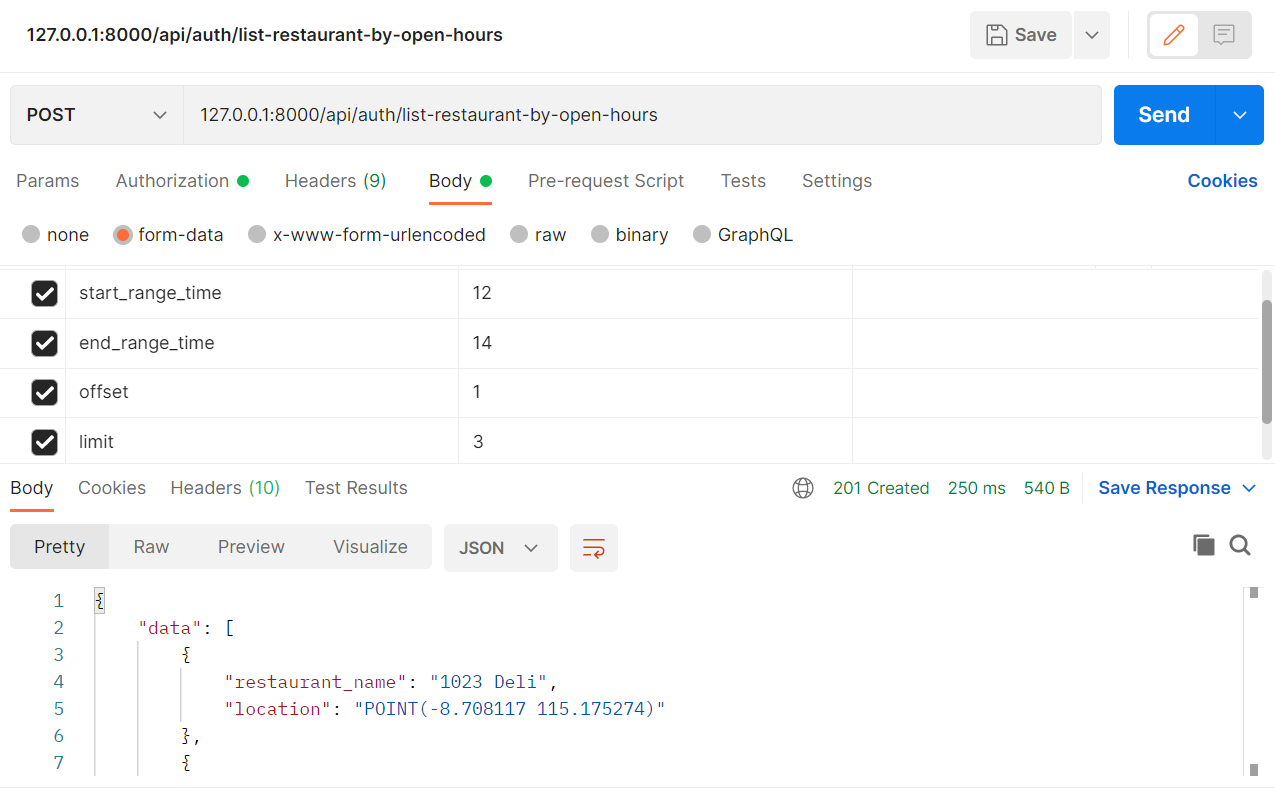
* **List all restaurants within the vicinity of the user’s location or (any location), ranked by distance [ /list-restaurant-by-distance ]**

This API can be used to search for restaurants based on latitude and longitude. This API can be accessed on the **customer role**. If latitude or longtitude is null, then the location used to find the nearest restaurant is the location of the user who logs in. Required key [latitude, longitude, offset, limit].



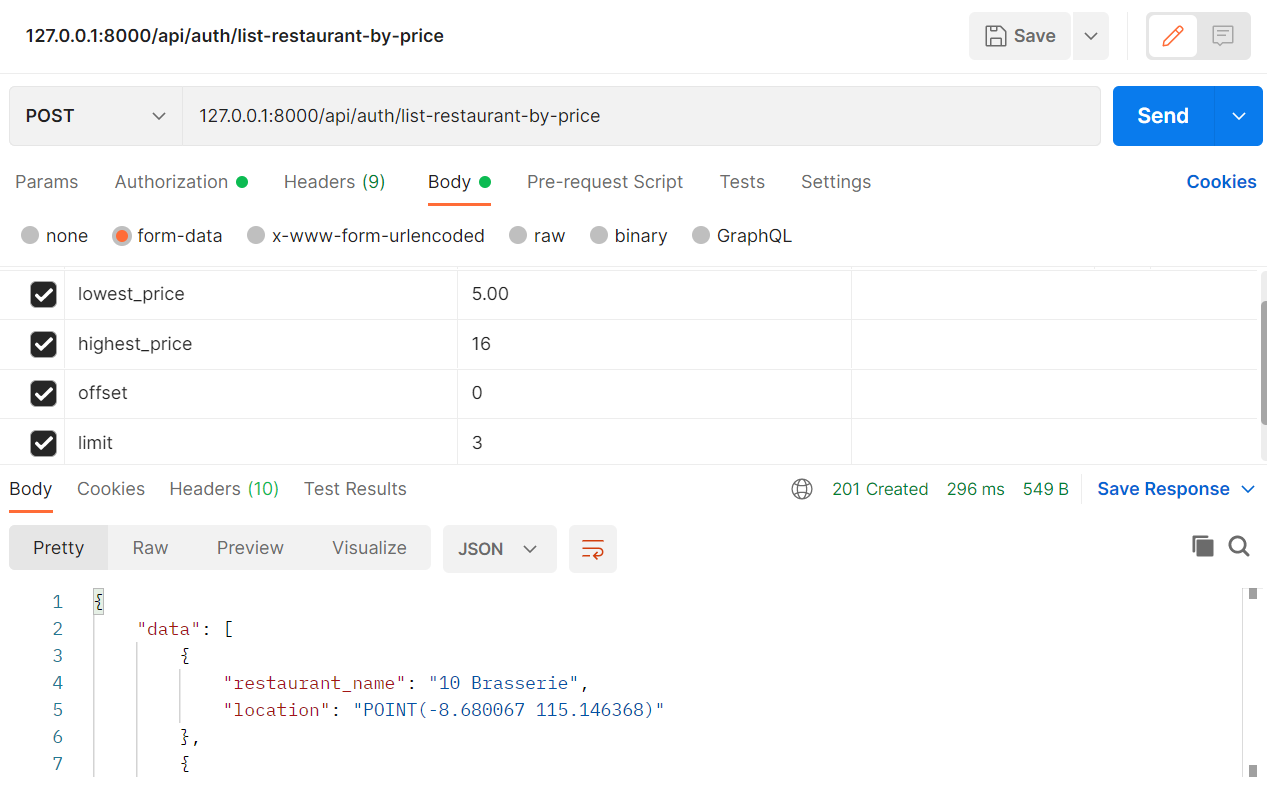
* **List all restaurants that are open for x-z hours per day or week [ /list-restaurant-by-open-hours ]**

This API can be used to search for restaurants based on range open hours. This API can be accessed on the **customer role**. Required key [start\_range\_time, end\_range\_time, offset, limit].



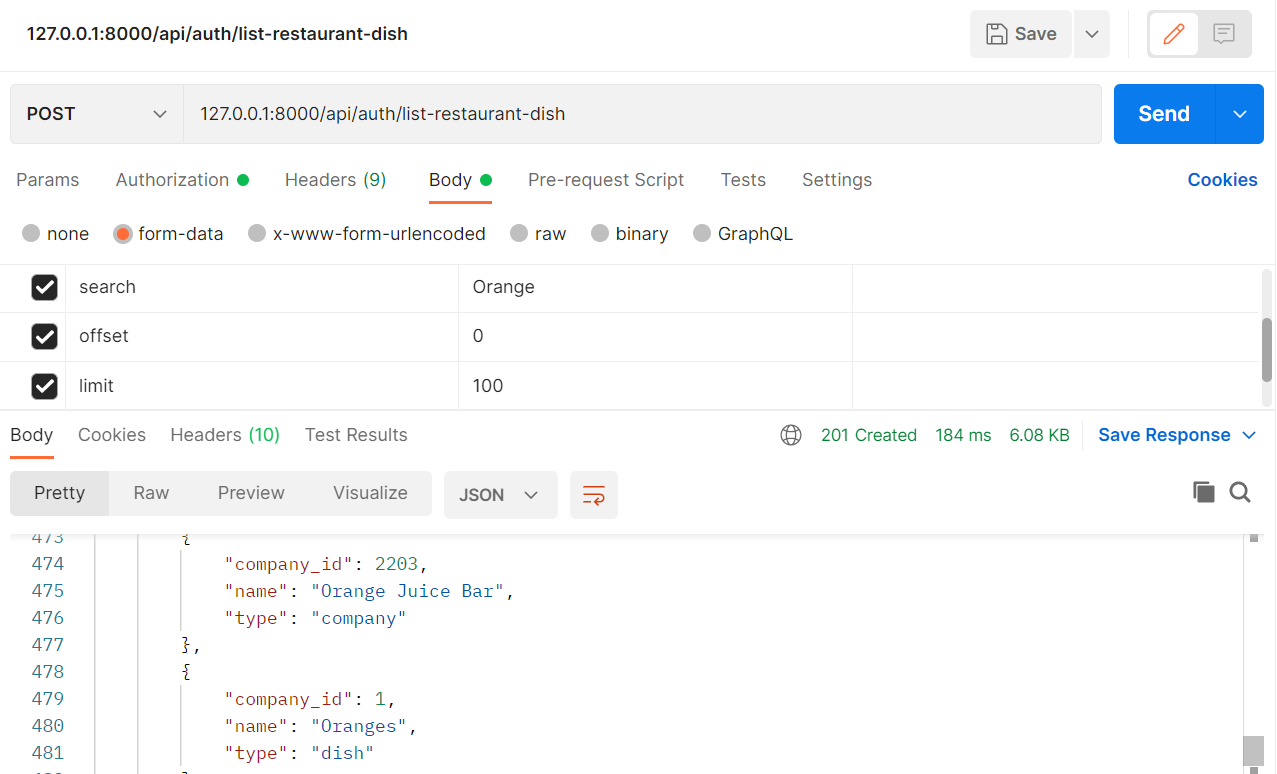
* **List all restaurants that have x-z number of dishes within a price range [ /list-restaurant-by-price ]**

This API can be used to search for restaurants based on a valid price range from the restaurant's menu. This API can be accessed on the **customer role**. Required key [lowest\_price, highest\_price, offset, limit].



* **Search for restaurants or dishes by name, ranked by relevance to search term [ /list-restaurant-dish ]**

This API can be used to search for restaurants or dish that matching search term. This API can be accessed on the **customer role**. Required key [search, offset, limit].



* **Search for restaurants that has a dish matching search term [ /list-restaurant-by-dish ]**

This API can be used to search for restaurants that has a dish matching search term. This API can be accessed on the **customer role**. Required key [dish, offset, limit].

