**Nhóm 8:**

**Hotel Booking**

**Thông tin Chung**

* Thành viên:
* Các dịch vụ và công cụ đã dùng:

Git, GitHub, Trello

Elastic Beanstalk, S3, RDS, Google Map APIs

**Links sản phẩm**

* Frontend:
* <http://hotel-fe-dev.ap-southeast-1.elasticbeanstalk.com/>
* <https://github.com/voanhhieu10250/hotel-fe>
* Backend:
* <http://hotelbe-env.eba-83wcrkq6.ap-southeast-1.elasticbeanstalk.com/swagger-ui.html#/>
* <https://github.com/trahuuminh/HotelBooking>
* Database design:
* <https://drive.google.com/file/d/1CV3Bo1vkrQ3PMHkzTtJHOw-xIL6mz-2x/view?usp=sharing>
* Trello:
* <https://trello.com/b/GhWIw52V/duancuoikhoajava12>

  - Video Demo:

      + https://www.youtube.com/watch?v=5Xtn6eqhACk

**Note:** Vì Google Map Api đang bị cấm ở Viêt Nam (bị giới hạn 1 request mỗi ngày). Nên để app hoạt động tốt nhất, vui lòng chạy Frontend ở máy local để có thể sử dụng google map 1 cách thoải mái.

**Thông tin tại đây**: <https://cloud.google.com/maps-platform/terms/maps-prohibited-territories/>

**Đặc tả APIs**

* User Controller:

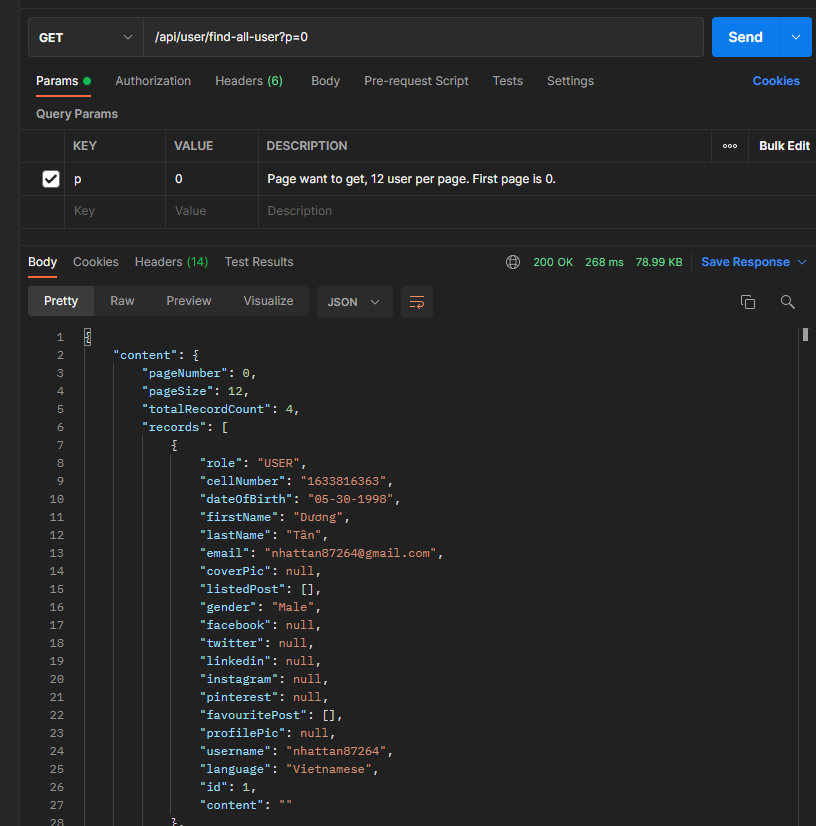
Account for test in production:

Admin:  username: admin  password: 123456

Normal User:  username: user password: 123456

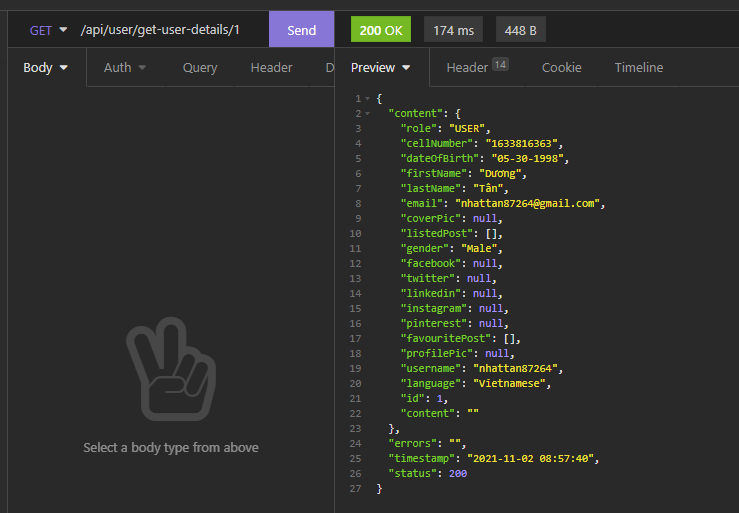
**(GET) /api/user/find-all-user**

Get all users info (with pagination, the first page number is 0, and 12 users per page)



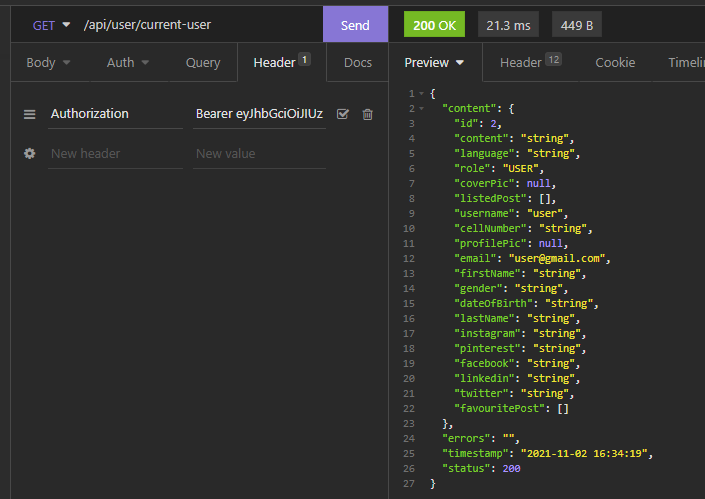
**(GET) /api/user/get-user-details/{user-id}**

Get user detail by id



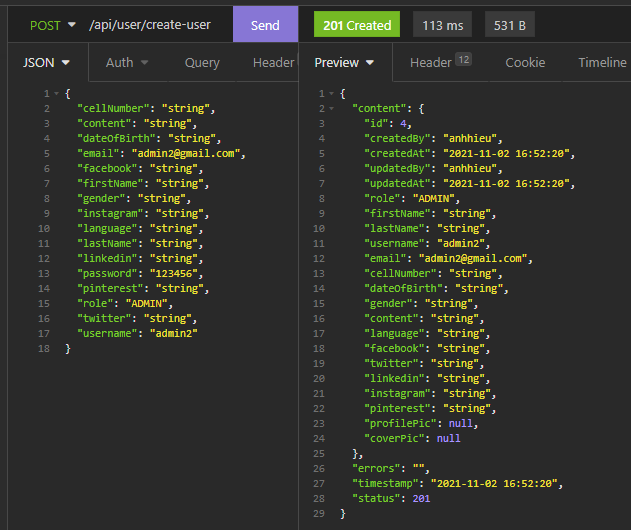
**(GET) /api/user/current-user**

Get user detail by token (Bearer token) in the request header



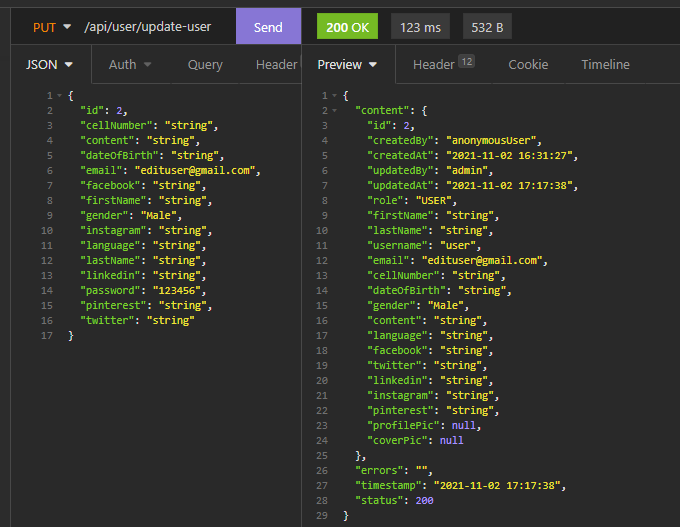
**(POST) /api/user/create-user**

Create a user (both admin and normal user). Only admin users can use this API.



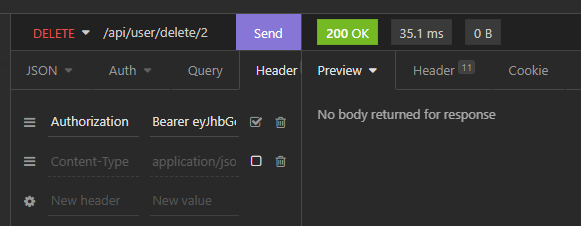
**(PUT) /api/user/update-user**

Update user info based on id (user id) passed in the request data body (id is required, the rest is optional). Only the admin user or the owner of the account can update the info.



**(DELETE) /api/user/delete/{user-id}**

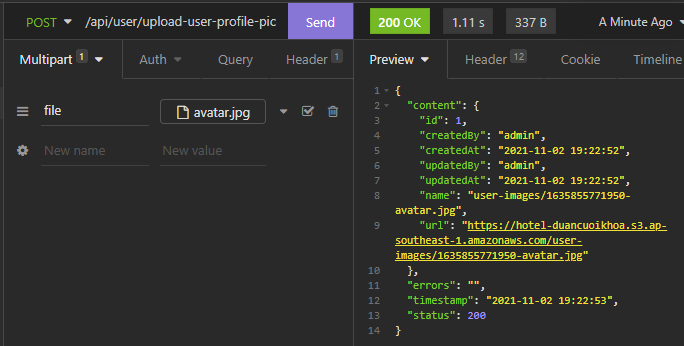
Delete a user by id. Only the admin user or the owner of the account can delete it.

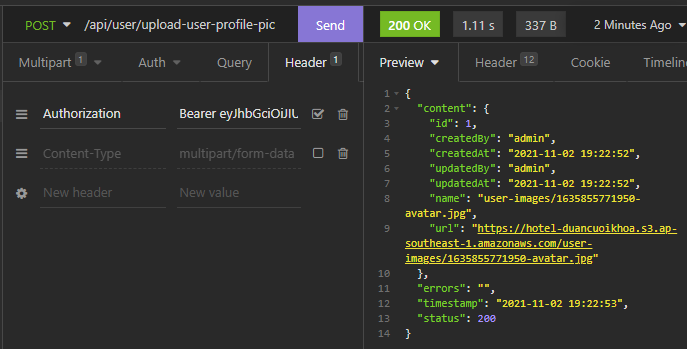


**(POST) /api/user/upload-user-profile-pic**

Upload profile picture for the user by token (Bearer token) in the request header

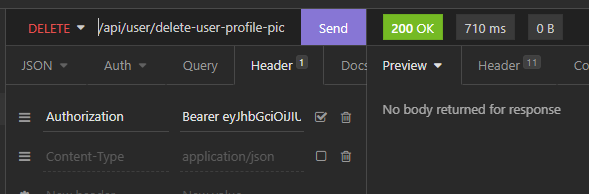
Content-Type: multipart/form-data





**(DELETE) api/user/upload-user-profile-pic**

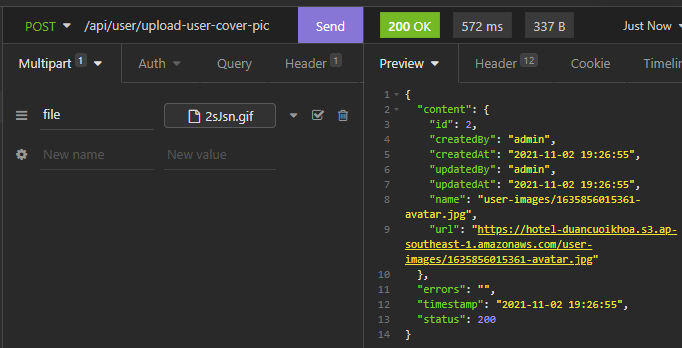
Delete user profile picture

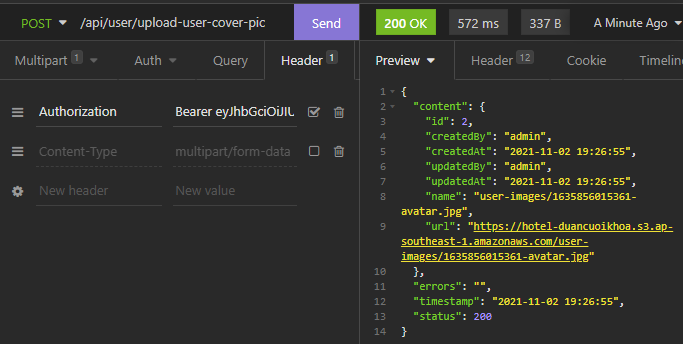


**(POST) /api/user/upload-user-cover-pic**

Upload cover picture for the user by token (Bearer token) in the request header

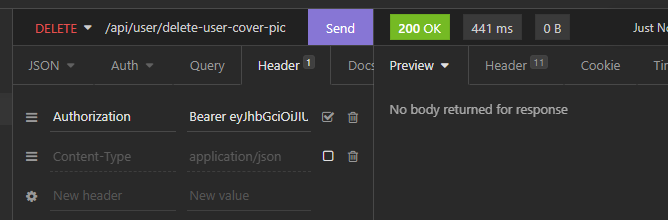
Content-Type: multipart/form-data





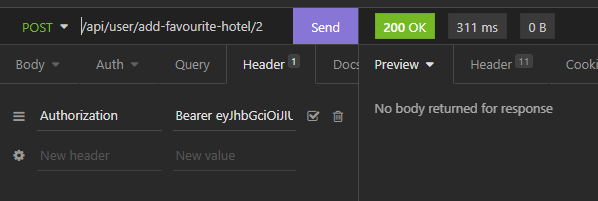
**(DELETE) api/user/upload-user-cover-pic**

Delete user cover picture



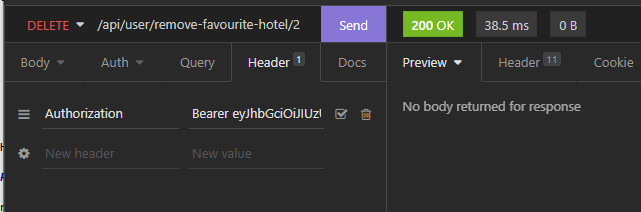
**(POST) /api/user/add-favourite-hotel/{hotel-id}**

Add a hotel to the user’s favourite list. Return 200 status when added success.



**(DELETE) /api/user/remove-favourite-hotel/{hotel-id}**

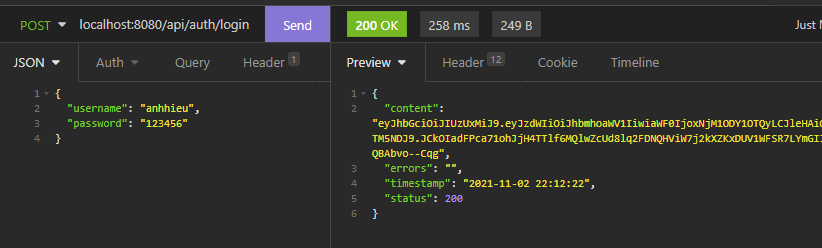
Delete a hotel from user’s favourite list. Return 200 status when deleted success.



* Auth Controller:

**(POST) /api/auth/login**

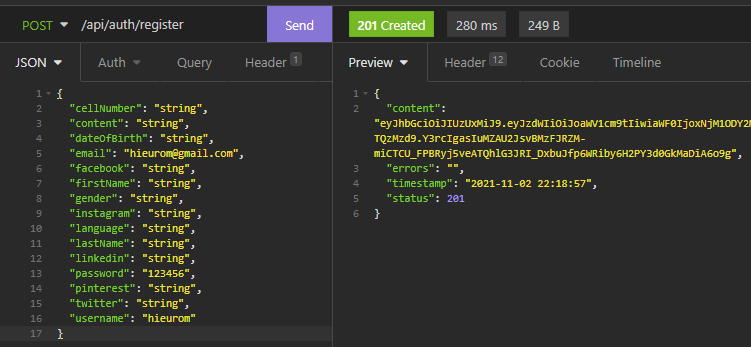
Login. Return user token if success



**(POST) /api/auth/register**

Register new user (create normal user only). Return user token if success

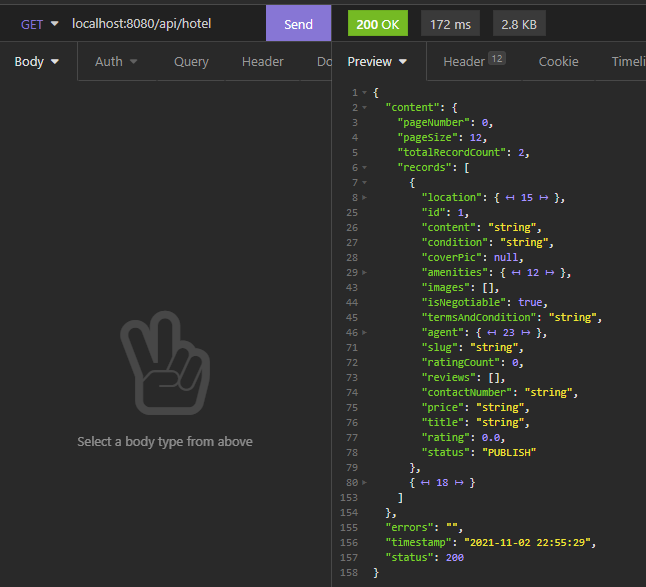
**username**, **password**, and **email** are required, the rest is optional



* Hotel Controller:

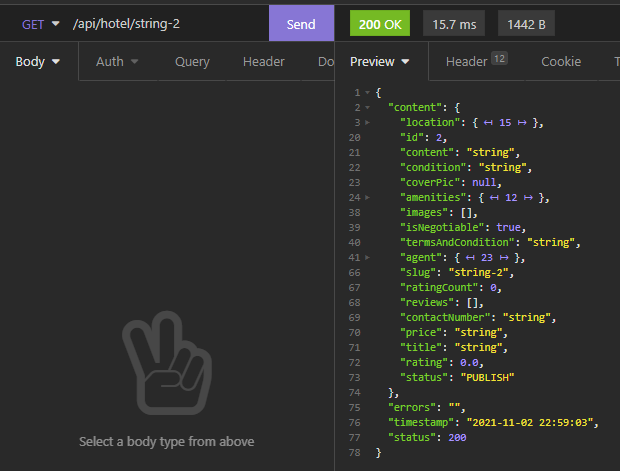
**(GET) /api/hotel**

Get all hotels.  (with pagination, the first page number is 0, and 12 users per page)



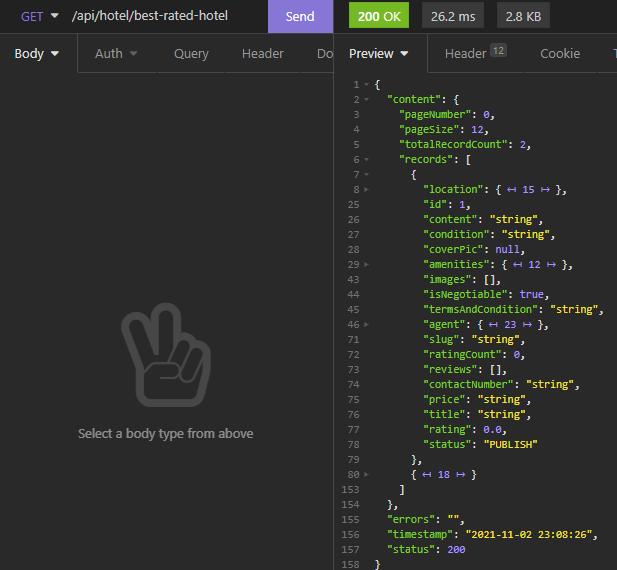
**(GET) /api/hotel/{slug}**

Get hotel detail by slug name (slug name is unique)



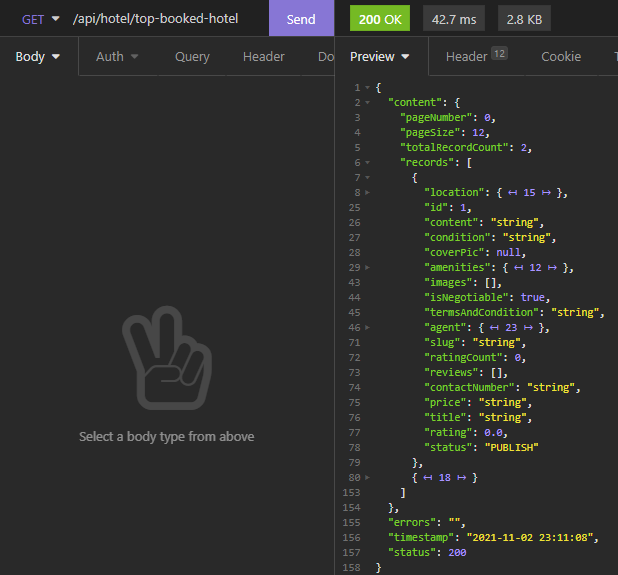
**(GET) /api/hotel/best-rated-hotel**

Get list of top-rated hotels based on reviews



**(GET) /api/hotel/top-booked-hotel**

Get list of top-booked hotels

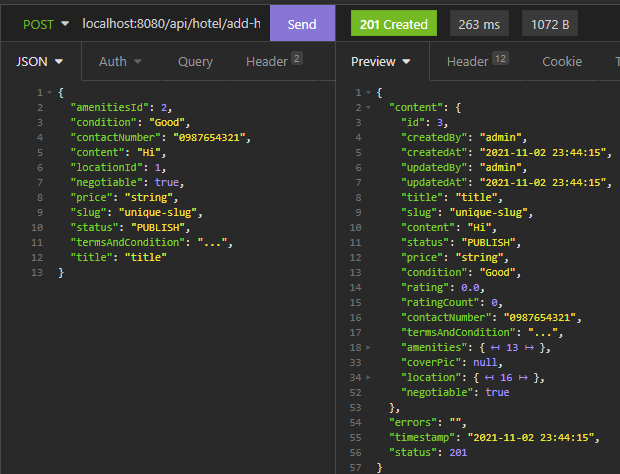


**(POST) /api/hotel/add-hotel**

Add new hotel

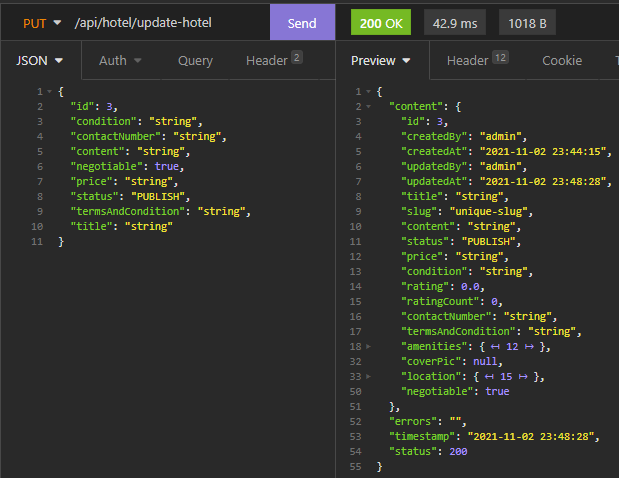
**Note**:

* To have **amenitiesId** and **locationId**, you have to add [new amenities](https://docs.google.com/document/d/10l3_TP9-t1kxb8dZpo-9qtRrqsUUsbTGZyPVXVYuItE/edit?fbclid=IwAR0I6-UIt7PrUH-JMakivRLVflYX_2ByjOQS2xpg3MpkdDpgQ42O43W43Q4#heading=h.r080nt2vouey), and a [new location](https://docs.google.com/document/d/10l3_TP9-t1kxb8dZpo-9qtRrqsUUsbTGZyPVXVYuItE/edit?fbclid=IwAR0I6-UIt7PrUH-JMakivRLVflYX_2ByjOQS2xpg3MpkdDpgQ42O43W43Q4#heading=h.m30jemb1i184) first.



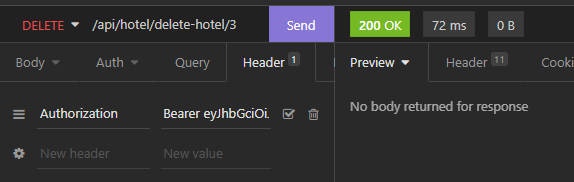
**(PUT) /api/hotel/update-hotel**

Update hotel info



**(DELETE) /api/hotel/delete-hotel/{hotel-id}**

Delete hotel

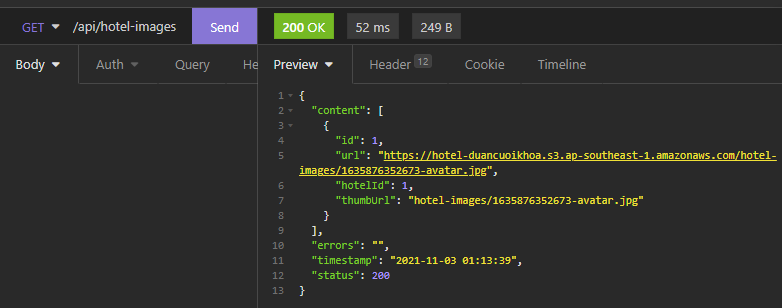


* Hotel Images Controller:

**Note**: Hotel images controller needs to work with [Storage controller](https://docs.google.com/document/d/10l3_TP9-t1kxb8dZpo-9qtRrqsUUsbTGZyPVXVYuItE/edit?fbclid=IwAR0I6-UIt7PrUH-JMakivRLVflYX_2ByjOQS2xpg3MpkdDpgQ42O43W43Q4#heading=h.dhkvb33hk6fx) in other to work correctly.

**(GET) /api/hotel-images**

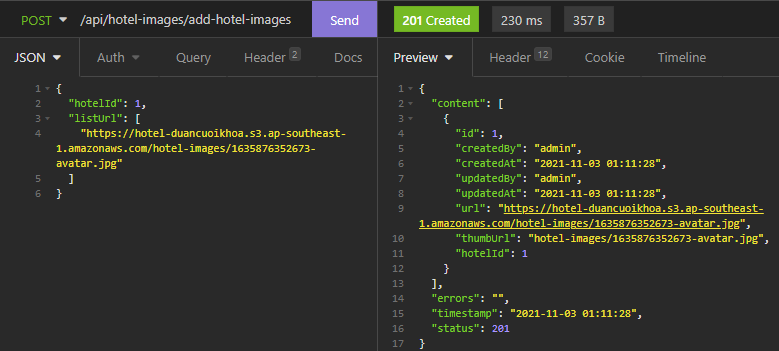
Get all hotel images of all hotels



**(POST) /api/hotel-images/add-hotel-images**

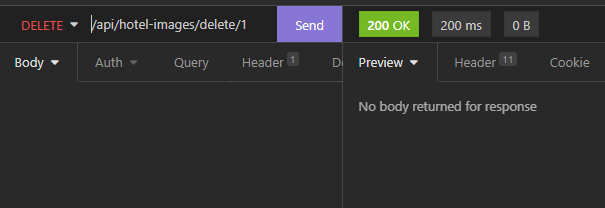
Add images for a hotel. You have to upload images to the server first (use [Storage controller](https://docs.google.com/document/d/10l3_TP9-t1kxb8dZpo-9qtRrqsUUsbTGZyPVXVYuItE/edit?fbclid=IwAR0I6-UIt7PrUH-JMakivRLVflYX_2ByjOQS2xpg3MpkdDpgQ42O43W43Q4#heading=h.g8dsv9nsi4ih)) then put those URLs in this API.

**listUrl**: list of images URLs



**(DELETE) /api/hotel-images/delete/{hotel-images-id}**

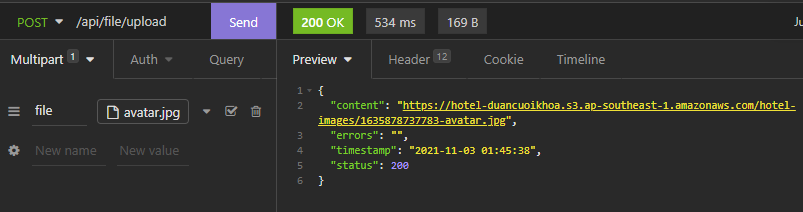
Delete hotel image by id (hotel-images id)



* Storage Controller:

**(POST) /api/file/upload**

Upload file image to S3. Return image URL if uploaded success.

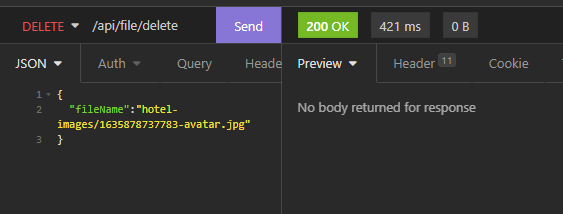


**(DELETE) /api/file/delete**

Delete file image form S3.

**fileName**: File name must begin with 'hotel-images/...'

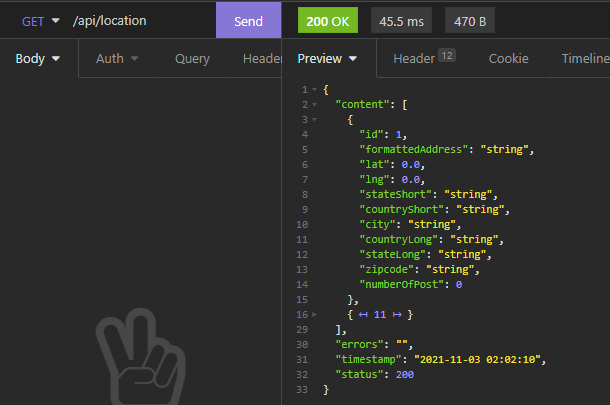
ex: hotel-images/1635878737783-avatar.jpg



* Location Controller:

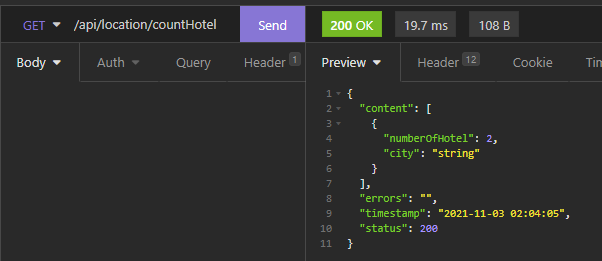
**(GET) /api/location**

Get all locations



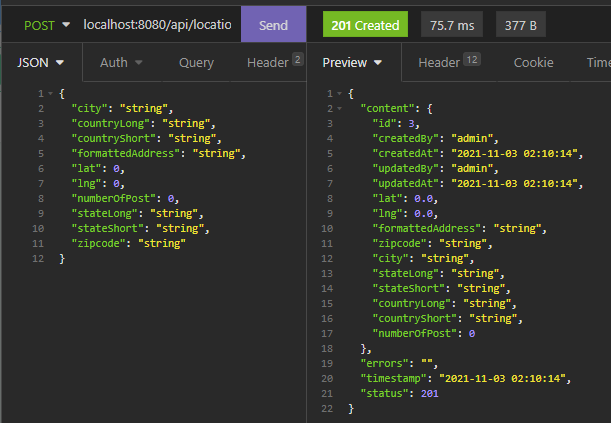
**(GET) /api/location/countHotel**

Get list location with numbers of hotels at that location



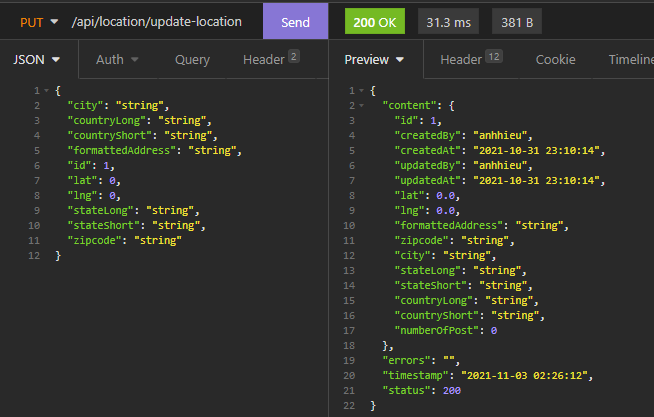
**(POST) /api/location/add-location**

Add a new location



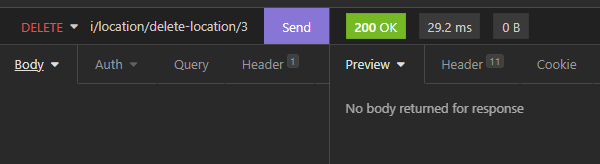
**(PUT) /api/location/update-location**

Update location info



**(DELETE) /api/location/delete-location/{location-id}**

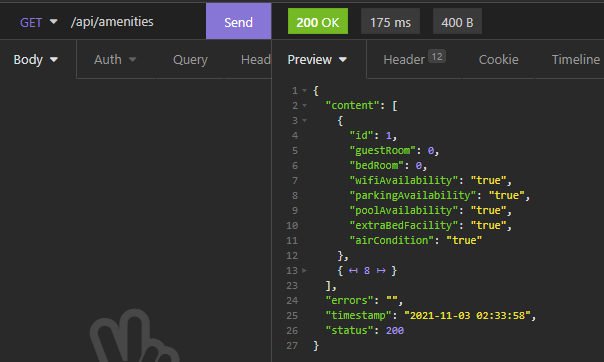
Delete location by id



* Amenities Controller:

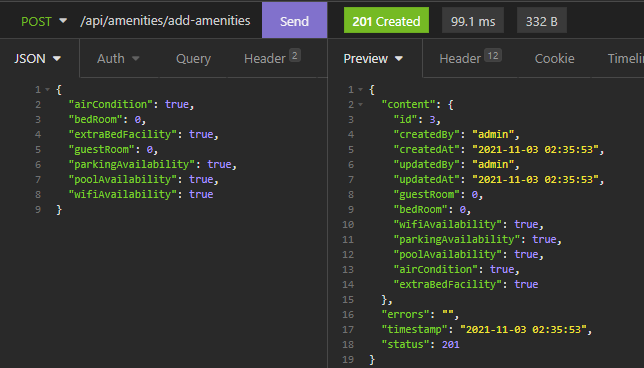
**(GET) /api/amenities**

Get a list of amenities of all hotels



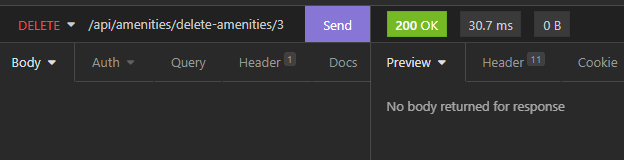
**(POST) /api/amenities/add-amenities**

Add new amenities for new hotel



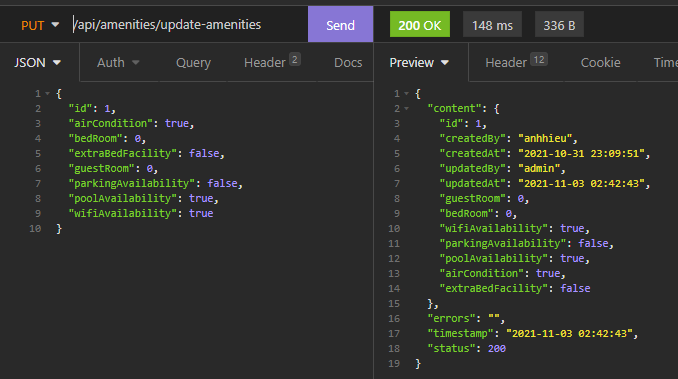
**(DELETE) /api/amenities/delete-amenities/{amenities-id}**

Delete amenities by id



**(PUT) /api/amenities/update-amenities**

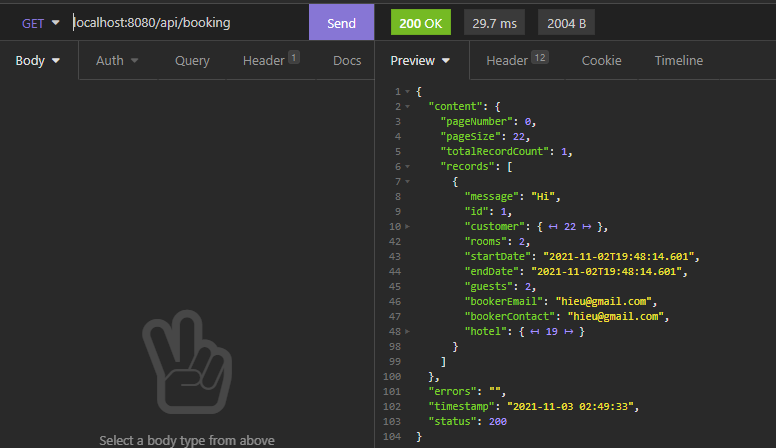
Update amenities info



* Booking Controller:

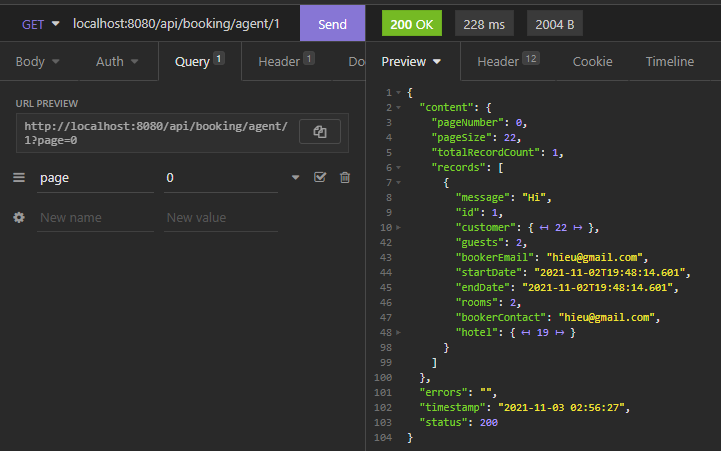
**(GET) /api/booking**

Get list of bookings of all hotels



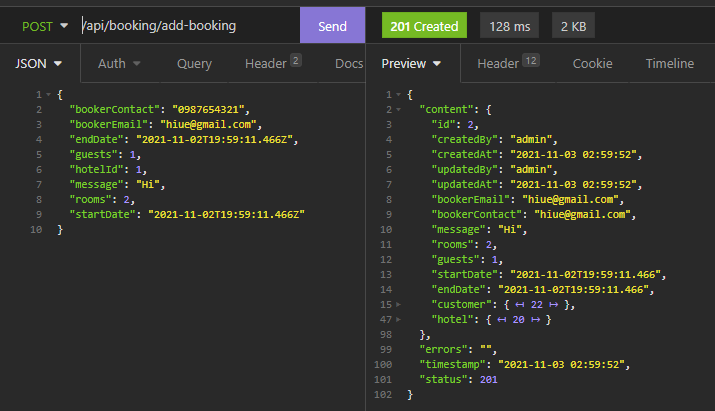
**(GET) /api/booking/agent/{agent-id}**

Get the list of bookings of a agent, based on agent id (with pagination, the first page number is 0, and 22 bookings per page)



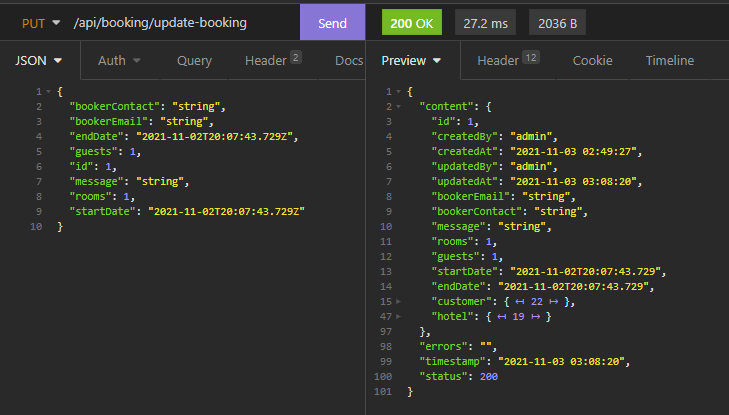
**(POST) /api/booking/add-booking**

Create new booking



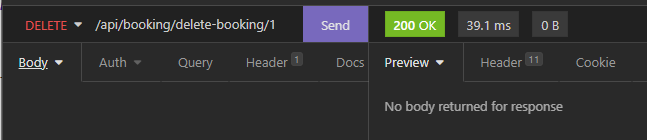
**(PUT) /api/booking/update-booking**

Update booking info



**(DELETE) /api/booking/delete-booking/{booking-id}**

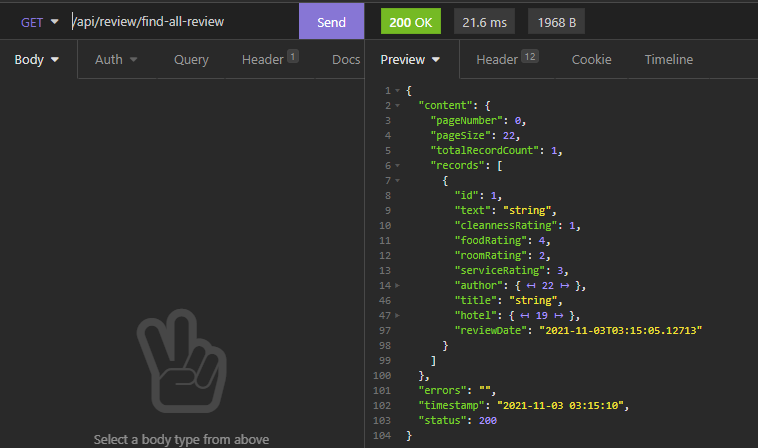
Delete booking



* Review Controller:

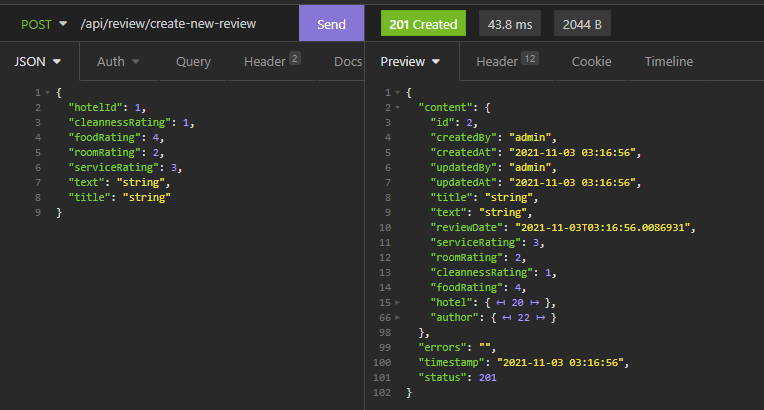
**(GET) /api/review/find-all-review**

Get list of reviews of all hotels



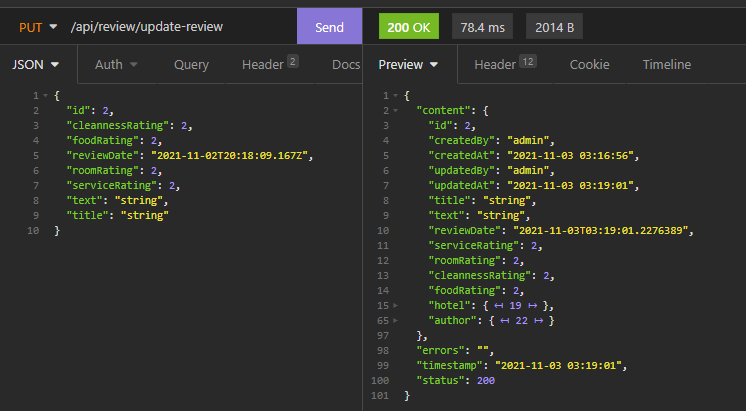
**(POST) /api/review/create-new-review**

Create a new review for a hotel



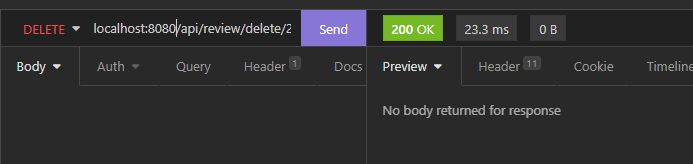
**(PUT) /api/review/update-review**

Update review info



**(DELETE) /api/review/delete/{review-id}**

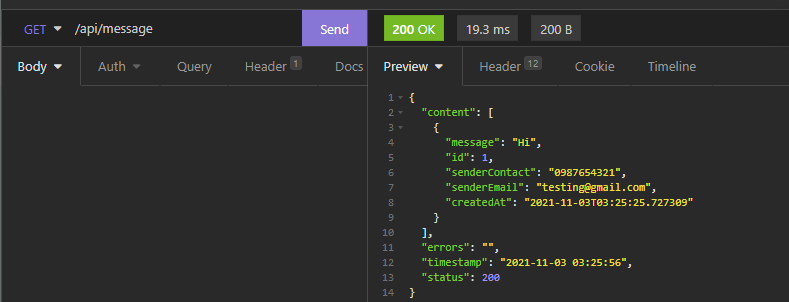
Delete review



* Message Controller:

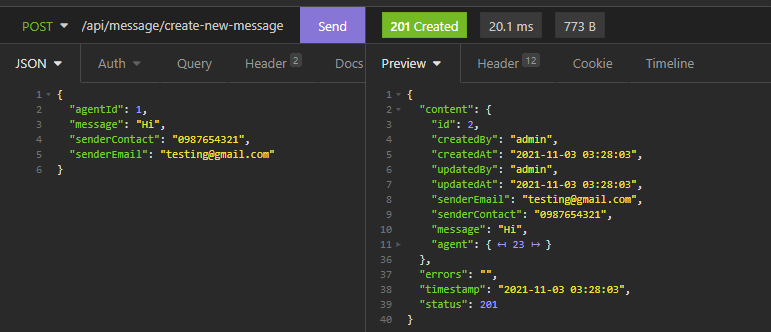
**(GET) /api/message**

Get list of messages of all agents



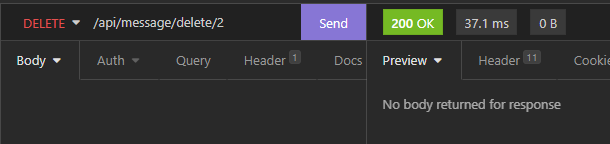
**(GET) /api/message/create-new-message**

Create new message for agent



**(DELETE) /api/message/delete/{message-id}**

Delete message by id (message id)



* Search Controller:

**(GET) /api/search**

Filter hotel listings based on request

