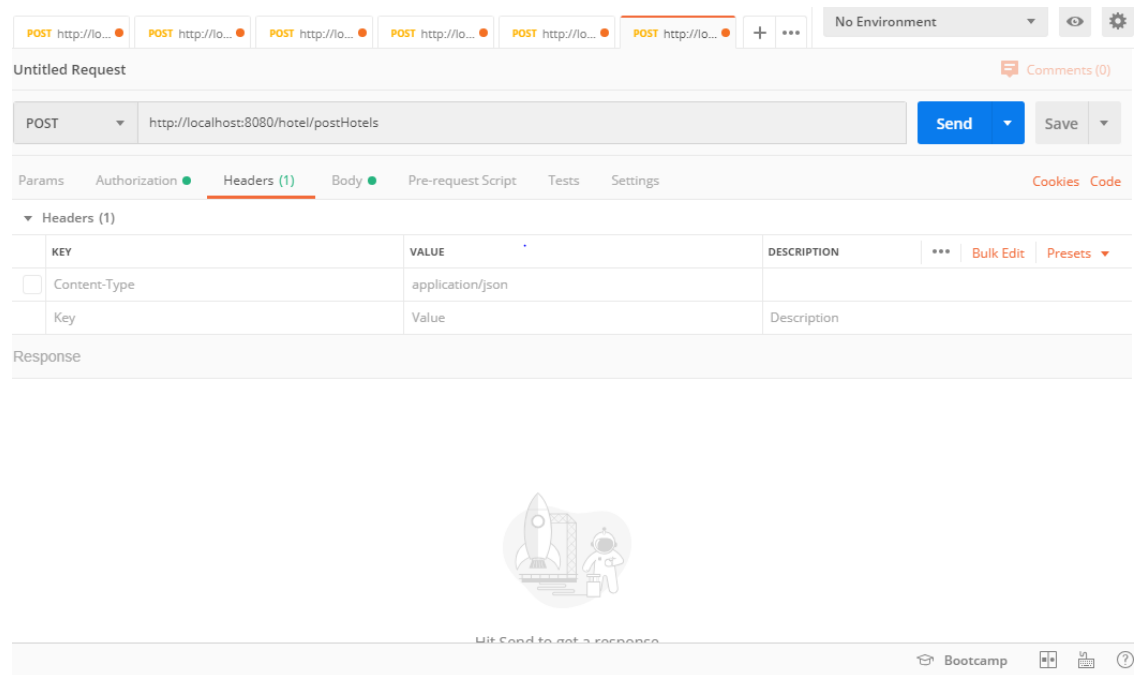


## 1. Database Setup(MySql): Execute the following queries:

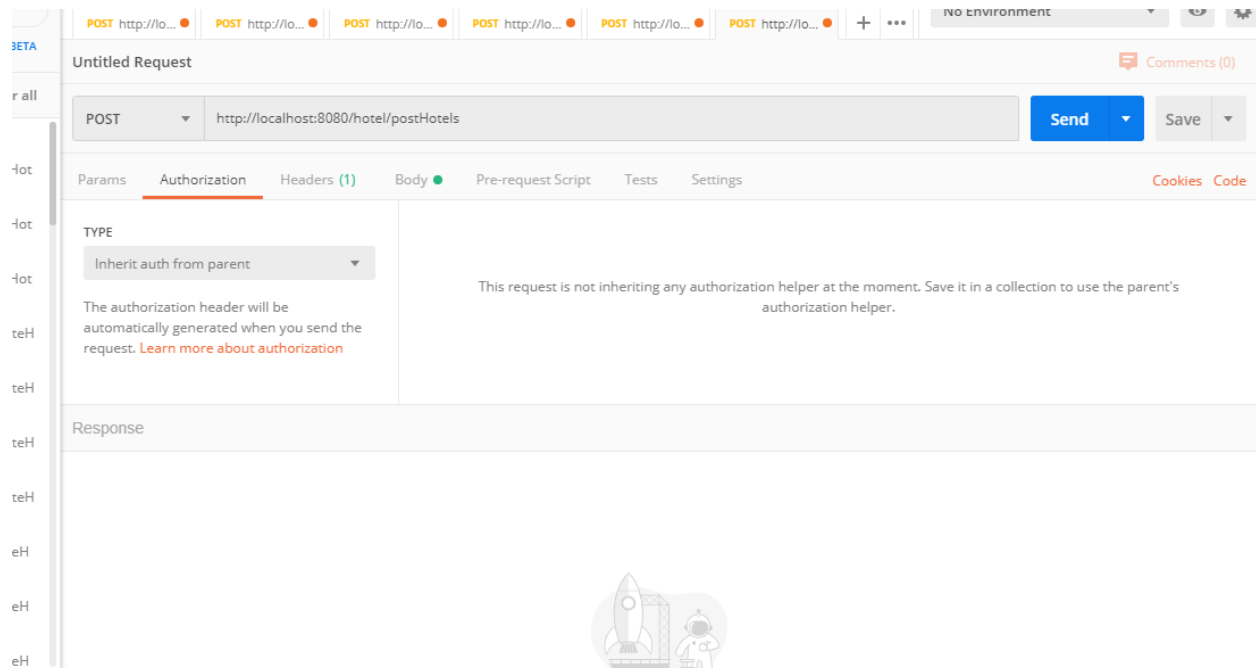
- create database test
- create table hotel (  
HOTEL\_ID int(11) NOT NULL PRIMARY KEY,  
HOTEL\_NAME varchar(100) NOT NULL,  
LOCATION varchar(100) NOT NULL,  
PRICE int(11) NULL,  
SPECIALITIES varchar(10000) NULL  
);
- alter table hotel MODIFY HOTEL\_ID int NOT null AUTO\_INCREMENT

## 2. Postman Setup

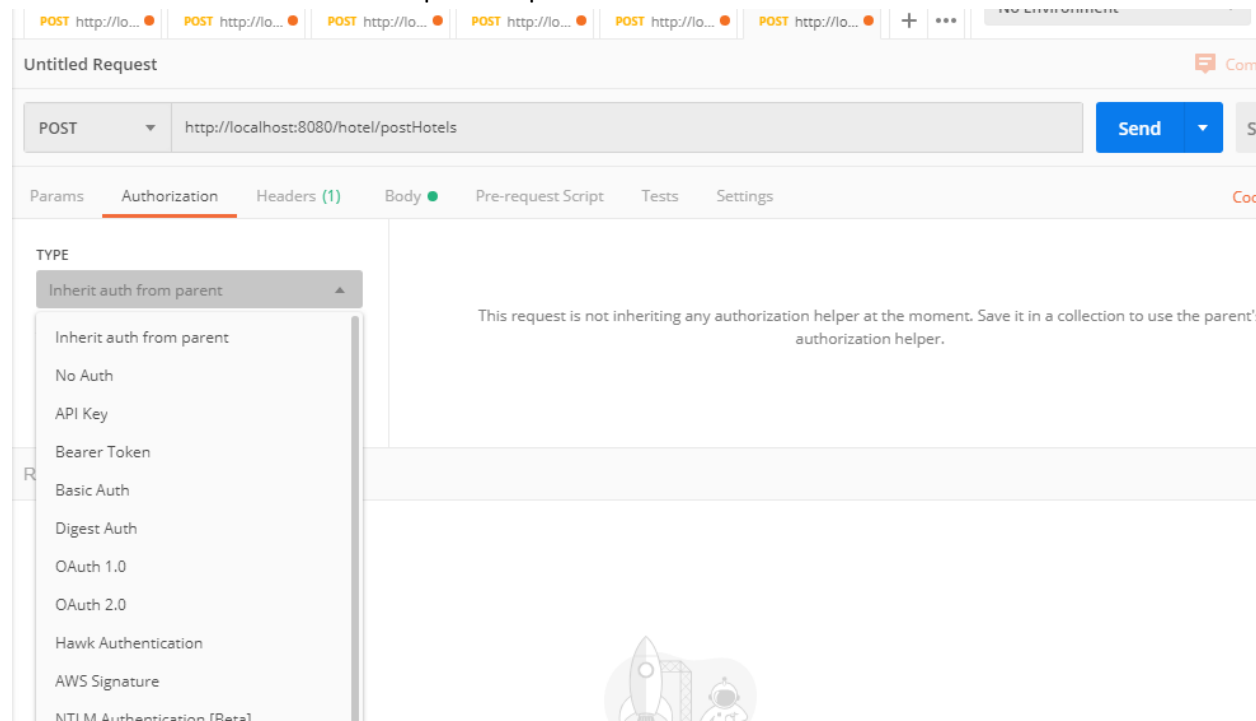
- Install postman
- Body--> raw--> JSON
- Settings for Admin/User authentication :(will be used during insert, update and delete hotels)
  - a. Open postman: Enter URL eg: `http://localhost:8080/hotel/postHotels`



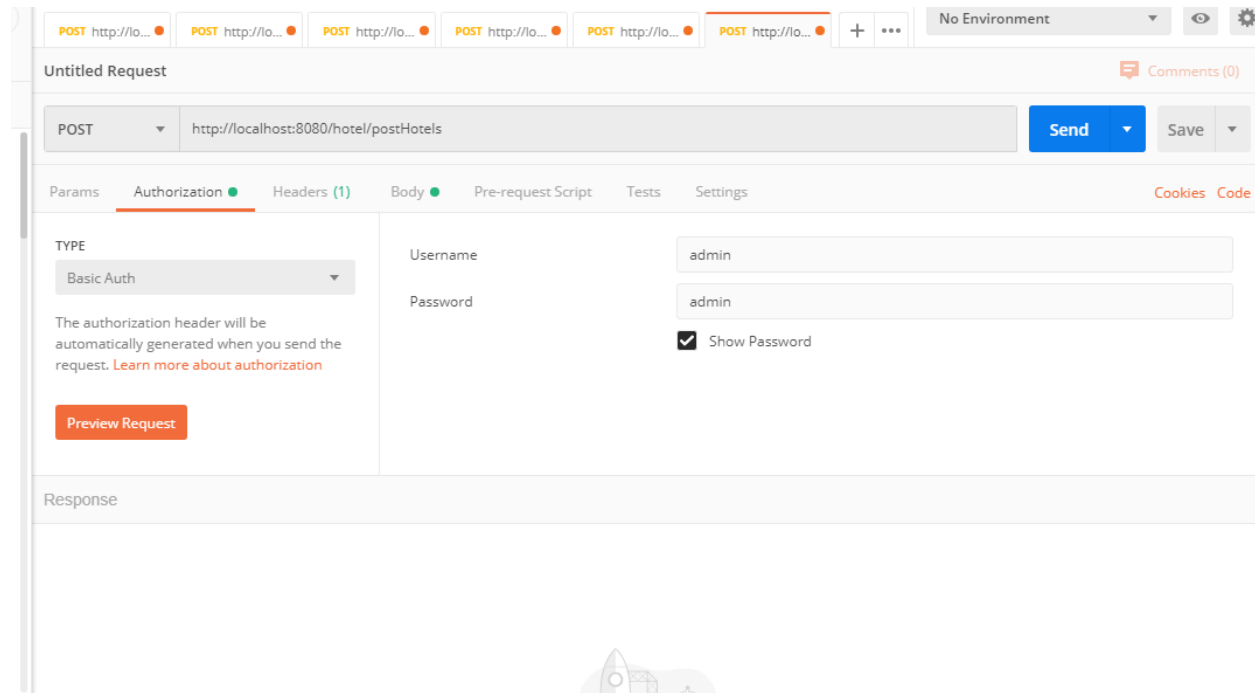
- b. Click Authorization



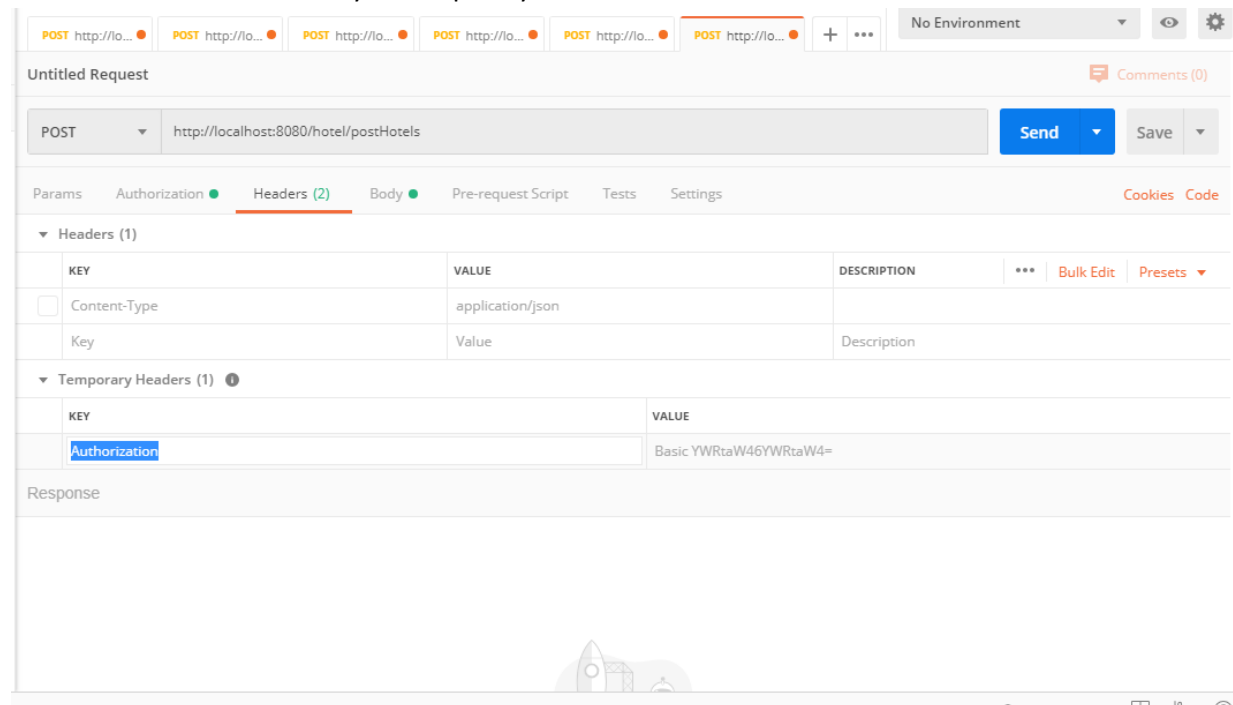
- c. Select **"Basic Auth"** from TYPE dropdown option



- d. Enter username and password as **"admin"**



- e. Select Preview Request
- f. Click on Headers
- g. You will see Authorization key in Temporary Headers section



### 3. Insert Single/Multiple records in table:

- URL: http://localhost:8080/hotel/postHotels
- RequestMapping : POST

Untitled Request Comments (0)


POST  Send Save

Params Authorization Headers Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Response



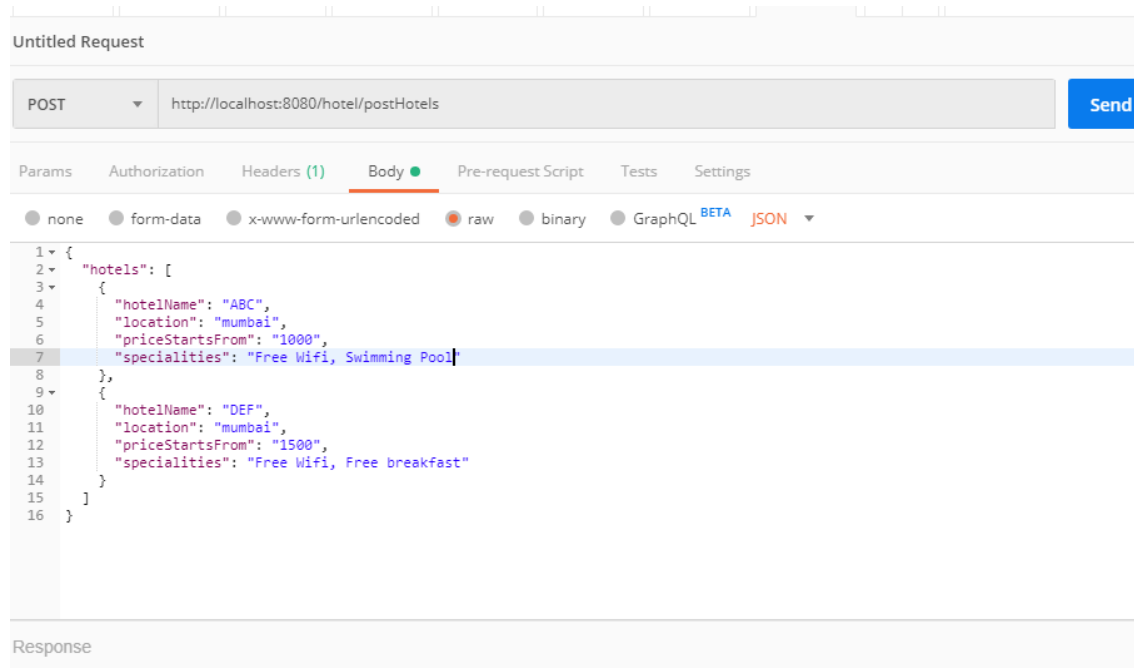
- Request: (single record)

```
{
  "hotels": [
    {
      "hotelName": "ABC",
      "location": "mumbai",
      "priceStartsFrom": "1000",
      "specialities": "Free Wifi"
    }
  ]
}
```

- Request : Multiple records(List of object(hotels))

```
{
  "hotels": [
    {
      "hotelName": "ABC",
      "location": "mumbai",
      "priceStartsFrom": "1000",
      "specialities": "Free Wifi"
    },
    {
      "hotelName": "DEF",
      "location": "mumbai",
      "priceStartsFrom": "1500",
      "specialities": "Free Wifi, Free breakfast"
    }
  ]
}
```

- Some basic validations to the field:
  - hotel\_name is mandatory field
  - location is mandatory field
  - priceStartsFrom should be in numeric
- Throws error if validations fails, (in case of multiple records NO records will be inserted if validation fails for any 1 or more records)
- Error Scenarios:
  - a. If authorization of "**admin**" fails
- Create a post request for inserting record in database **without step2:**



Untitled Request


POST http://localhost:8080/hotel/postHotels

Params Authorization Headers (1) Body Pre-request Script Tests Settings

▼ Headers (1)

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	Content-Type	application/json	
	Key	Value	Description

Response




- You should get UNAUTHORIZED error response:

POST http://localhost:8080/hotel/postHotels Send

```
14 }
15 ]
16 }
```

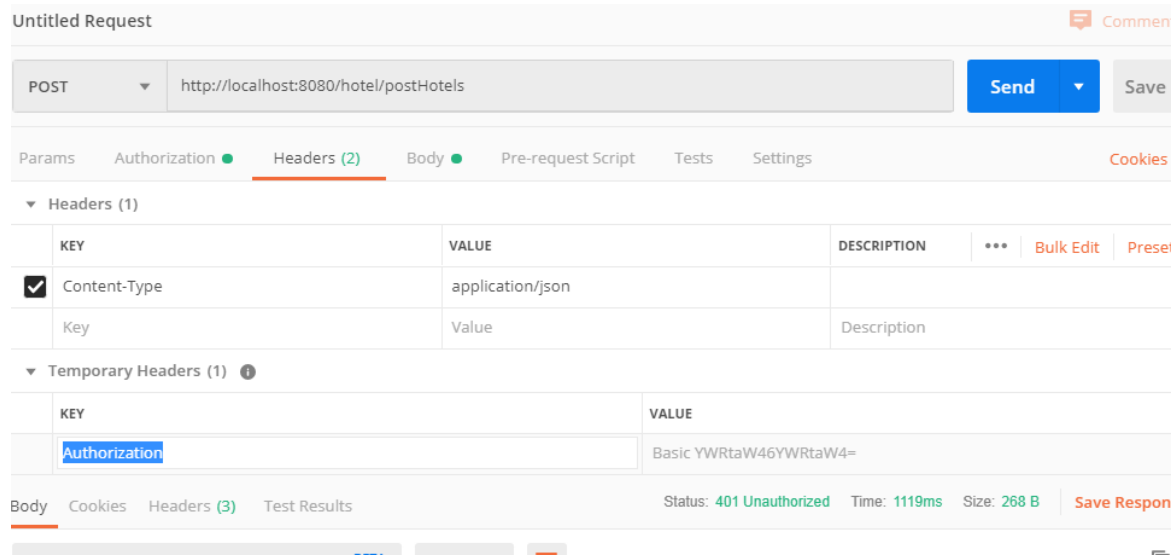
Body Cookies Headers (3) Test Results Status: 401 Unauthorized Time: 1119ms Size: 268 B

Pretty Raw Preview Visualize BETA JSON 

```
1 {
2   "transactionStatus": "Error",
3   "messages": [
4     {
5       "code": "INVALID_LOGIN",
6       "description": "You are not authorized to insert",
7       "record": ""
8     }
9   ]
10 }
```

Postman

- b. Setup authorization in postman same as **step2:(Mandatory)**



- c. Now post the request with 1 record with hotel\_name empty:

```
{
  "hotels": [
    {
      "hotelName": "",
      "location": "mumbai",
      "priceStartsFrom": "1000",
      "specialities": "Free Wifi, Swimming Pool"
    },
    {
      "hotelName": "DEF",
      "location": "mumbai",
      "priceStartsFrom": "1500",
      "specialities": "Free Wifi, Free breakfast"
    }
  ]
}
```

- o Validation should fail for when hotelName is empty. Error response :

Untitled Request

POST http://localhost:8080/hotel/postHotels

Params Authorization Headers (2) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON

```
1 {
2   "hotels": [
3     {
4       "hotelName": "",
5       "location": "mumbai",
6       "priceStartsFrom": "1000",
7       "specialities": "Free Wifi, Swimming Pool"
8     },
9     {
10      "hotelName": "DEF",
11      "location": "mumbai",
12      "priceStartsFrom": "1500",
13      "specialities": "Free Wifi, Free breakfast"
14    }
15  ]
16 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 41ms Size: 395 B Save R

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "transactionStatus": "Error",
3   "messages": [
4     {
5       "code": "HOTEL_NAME_INVALID",
6       "description": "Hotel Name is mandatory field",
7       "record": "HotelPostRequest [hotelName=, location=mumbai, priceStartsFrom=1000, specialities=Free Wifi, Sw
8         Pool]"
9     }
10  ]
11 }
```

- Check in db:  
select \* from hotel order by hotel\_id desc:



Server: 127.0.0.1 » Database: test » Table: hotel

Browser Structure SQL Search Insert Export Import Privileges Oper

Show query box

Showing rows 0 - 0 (1 total, Query took 0.0025 seconds.) [HOTEL\_ID: 104... - 104...]

SELECT \* FROM `hotel` order by HOTEL\_ID desc

Profiling [Edit inline] [Ex]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	HOTEL_ID	HOTEL_NAME	LOCATION	PRICE	SPECIALITIES
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	104	KBC	MUMBAI	200	Free Breakfast

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label:  ☐ Let every user access this bookmark

- Similarly try for validations of hotelName, location and price
- **Happy Path Scenarios:**
  - Setup authorization for admin using postman refer **step2(Mandatory)**
  - Request for single record insert:

Untitled Request

POST http://localhost:8080/hotel/postHotels Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON

```

1 {
2   "hotels": [
3     {
4       "hotelName": "ABC",
5       "location": "mumbai",
6       "priceStartsFrom": "1000",
7       "specialities": "Free Wifi, Swimming Pool"
8     }
9   ]
10 }

```

- **Success Response:**

```
Body Cookies Headers (3) Test Results Status: 200 OK Time: 1500ms Size: 27

Pretty Raw Preview Visualize BETA JSON ↺

1 {
2   "transactionStatus": "Success",
3   "messages": [
4     {
5       "code": "RECORDS_INSERTED",
6       "description": "Records have been successfully inserted",
7       "record": "1"
8     }
9   ]
10 }
```

- Request for inserting multiple records at a time:

```
{
  "hotels": [
    {
      "hotelName": "LMO",
      "location": "delhi",
      "priceStartsFrom": "2000",
      "specialities": ""
    },
    {
      "hotelName": "DEF",
      "location": "mumbai",
      "priceStartsFrom": "1500",
      "specialities": "Free Wifi, Free breakfast"
    },
    {
      "hotelName": "LPQRTS",
      "location": "Pune",
      "priceStartsFrom": "3500",
      "specialities": "Free Wifi, Free breakfast"
    }
  ]
}
```

POST http://localhost:8080/hotel/postHotels

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON

```
1 {
2   "hotels": [
3     {
4       "hotelName": "LMO",
5       "location": "delhi",
6       "priceStartsFrom": "2000",
7       "specialities": ""
8     },
9     {
10      "hotelName": "DEF",
11      "location": "mumbai",
12      "priceStartsFrom": "1500",
13      "specialities": "Free Wifi, Free breakfast"
14    },
15    {
16      "hotelName": "LPQRTS",
17      "location": "Pune",
18      "priceStartsFrom": "3500",
19      "specialities": "Free Wifi, Free breakfast"
20    }
21  ]
22 }
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 1500ms Size: 271 B Save Response

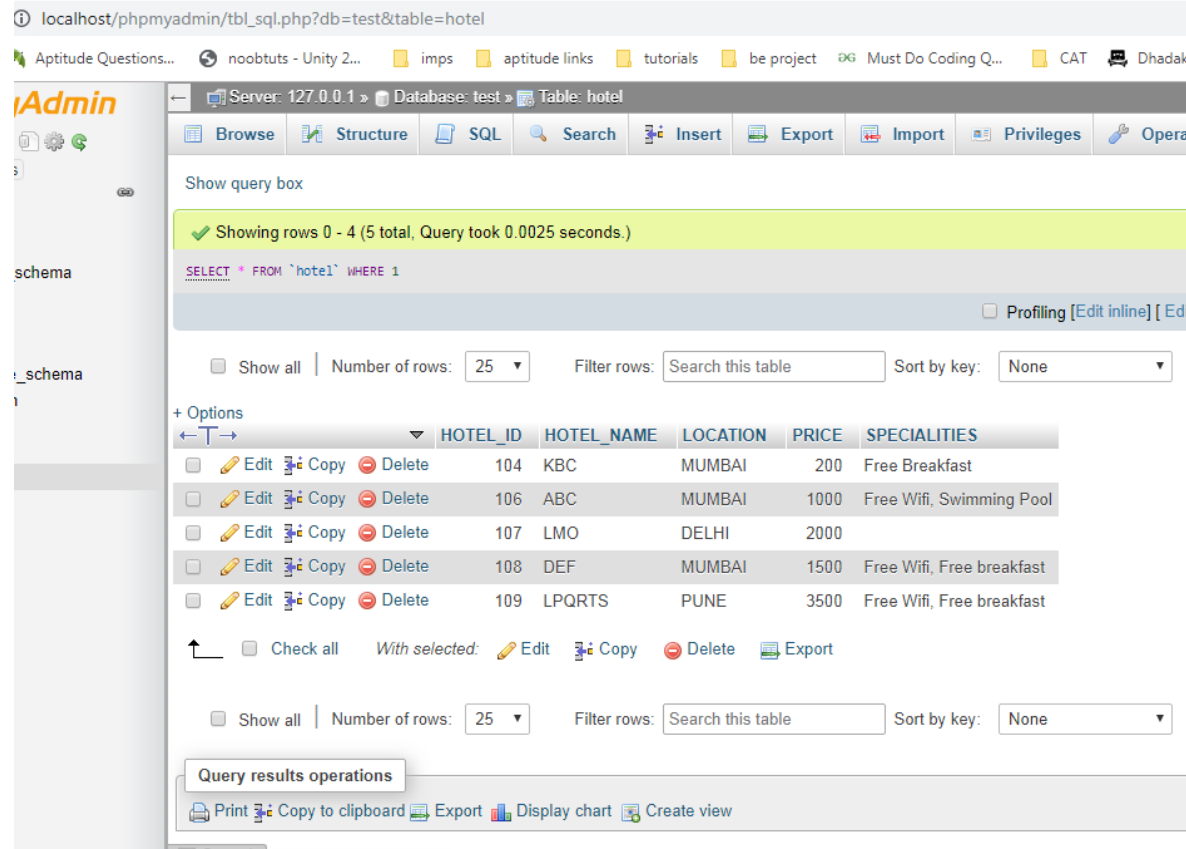
- Success Response:

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "transactionStatus": "Success",
3   "messages": [
4     {
5       "code": "RECORDS_INSERTED",
6       "description": "Records have been successfully inserted",
7       "record": "3"
8     }
9   ]
10 }
```

- check in db



#### 4. GetRecords by dynamically search criteria

- URL : `http://localhost:8080/hotel/getHotels`
- RequestMapping : **POST**
- Request:**

```
{
  "hotelId": "",
  "hotelName": "",
  "priceRangeFrom": "",
  "priceRangeTo": "",
  "location": "",
  "keyword": ""
}
```

Description:

- if hotelId is provided then, it fetches as per hotelId
- If hotelName is provided then it fetches records with hotelName which contains chars/strings provided as hotelName (for eg: "hotelName": "A", it searches for hotelName which contains "A" in it)

- If priceRangeFrom is provided then it fetches records of hotels having price greater than or equal to priceRangeFrom
- If priceRangeTo is provided then it fetches records of hotel having price less than or equal to priceRangeTo
- If location is provided then it fetches records of hotels whose locations contains chars/strings provided as location (for eg: " location ":"mum", it searches for location which contains "mum" in it)
- If keyword is provided then it fetches records of hotels whose specialities contains chars/strings provided as keyword (for eg: " keyword":"wifi", it searches for specialities which contains "wifi" in it)
- If **none of the above fields** are provided then ,it fetches all the records

d. Here, no admin authorization is required

e. Validations:

- priceRangeFrom , priceRangeTo and hotelId should be numeric

f. Error Scenario:

- Request:

```
{
  "hotelId": "",
  "hotelName": "",
  "priceRangeFrom": "abc",
  "priceRangeTo": "",
  "location": "",
  "keyword": ""
}
```

Untitled Request

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8080/hotel/getHotels
- Body Type:** JSON (selected from a dropdown menu that also includes none, form-data, x-www-form-urlencoded, raw, binary, and GraphQL BETA)
- Body Content:**

```
1 {
2   "hotelId": "",
3   "hotelName": "",
4   "priceRangeFrom": "abc",
5   "priceRangeTo": "",
6   "location": "",
7   "keyword": ""
8 }
```

- Resposne:

```
Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 87ms Size: 405 B Save Respons

Pretty Raw Preview Visualize BETA JSON

1 {
2   "transactionStatus": "Error",
3   "messages": [
4     {
5       "code": "PRICE_FROM_INVALID",
6       "description": "Price Range From should be in numeric",
7       "record": "HotelSearchRequest [hotelId=, hotelName=, priceRangeFrom=abc, priceRangeTo=, location=, keyword=]"
8     }
9   ],
10  "hotels": []
11 }
```

Similarly try for princeRangeTo and hotelId

g. Happy Scenarios: table hotel contains following data:

localhost/phpmyadmin/tbl\_sql.php?db=test&table=hotel

Server: 127.0.0.1 » Database: test » Table: hotel

Showing rows 0 - 4 (5 total, Query took 0.0025 seconds.)

```
SELECT * FROM `hotel` WHERE 1
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	HOTEL_ID	HOTEL_NAME	LOCATION	PRICE	SPECIALITIES
<input type="checkbox"/> Edit Copy Delete	104	KBC	MUMBAI	200	Free Breakfast
<input type="checkbox"/> Edit Copy Delete	106	ABC	MUMBAI	1000	Free Wifi, Swimming Pool
<input type="checkbox"/> Edit Copy Delete	107	LMO	DELHI	2000	
<input type="checkbox"/> Edit Copy Delete	108	DEF	MUMBAI	1500	Free Wifi, Free breakfast
<input type="checkbox"/> Edit Copy Delete	109	LPQRTS	PUNE	3500	Free Wifi, Free breakfast

Check all With selected: Edit Copy Delete Export

Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

h. search hotels with location="Mumbai" and specialties="wifi", here we should retrieve 2 records as per data in table

- Request:

```
{  
  "hotelId": "",  
  "hotelName": "",  
  "priceRangeFrom": "",  
  "priceRangeTo": "",  
  "location": "Mumbai",  
  "keyword": "wifi"  
}
```

#### Untitled Request

POST

http://localhost:8080/hotel/getHotels

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

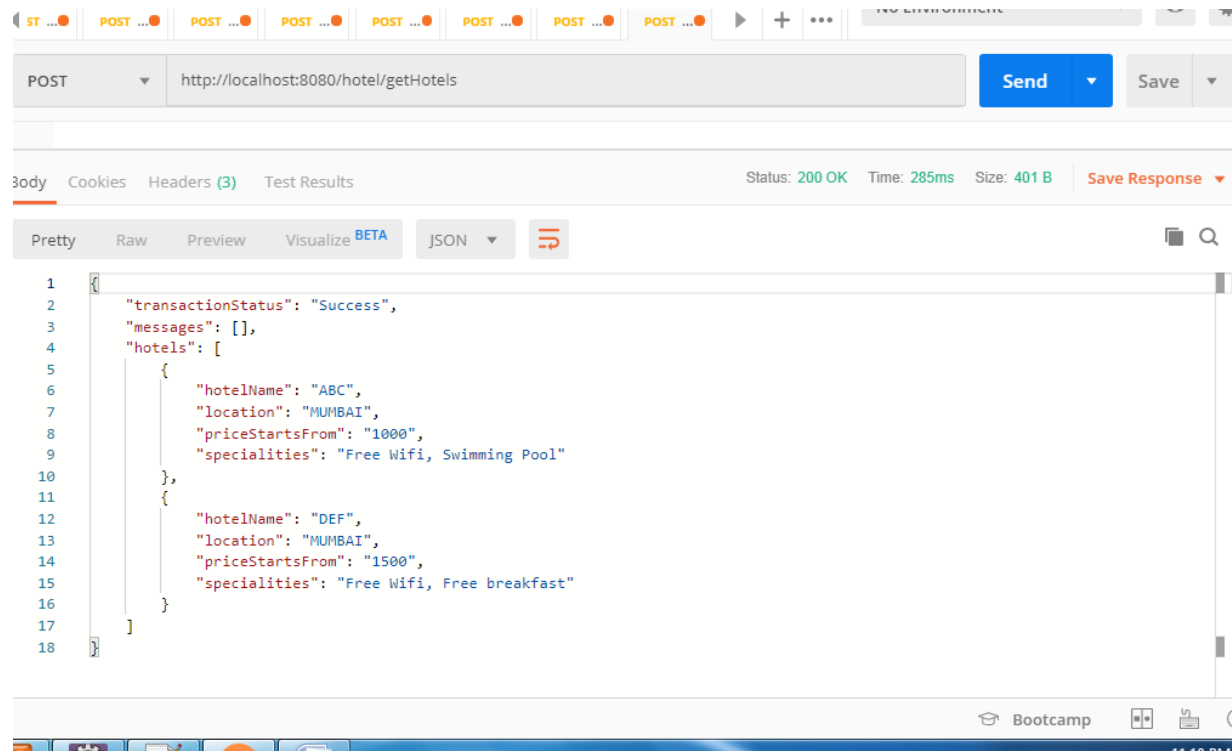
binary

GraphQL BETA

JSON

```
1 {  
2   "hotelId": "",  
3   "hotelName": "",  
4   "priceRangeFrom": "",  
5   "priceRangeTo": "",  
6   "location": "Mumbai",  
7   "keyword": "wifi"  
8 }
```

- Response:

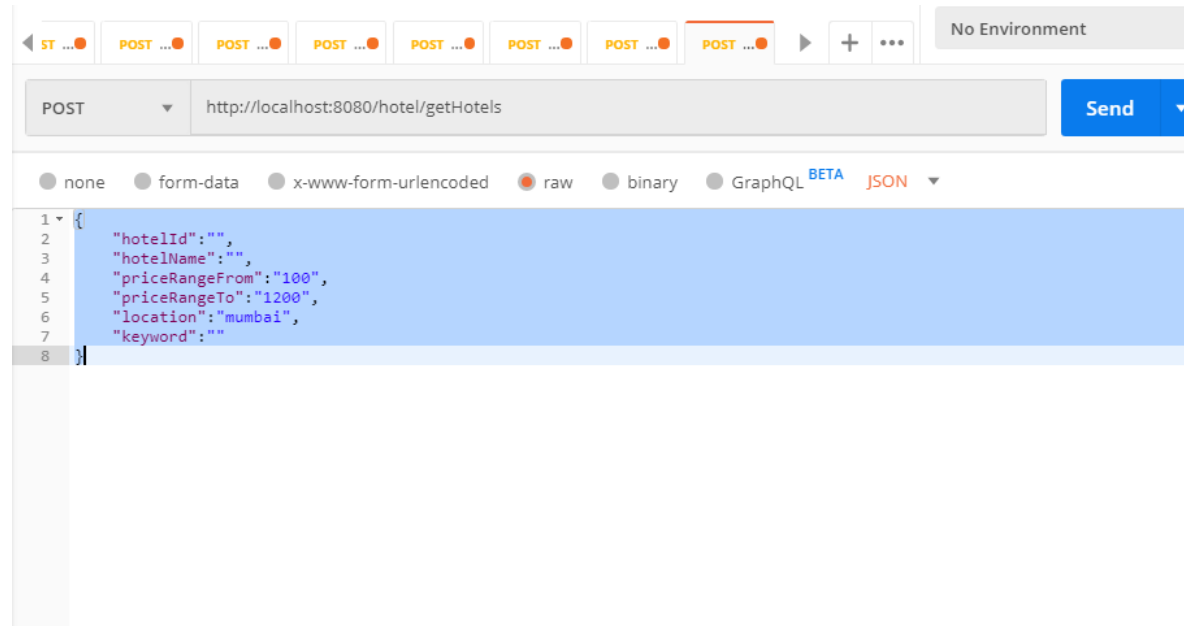


i. example 2: search records for having price between 100 to 1200 and in mumbai; As per data in table 2 records will be fetched

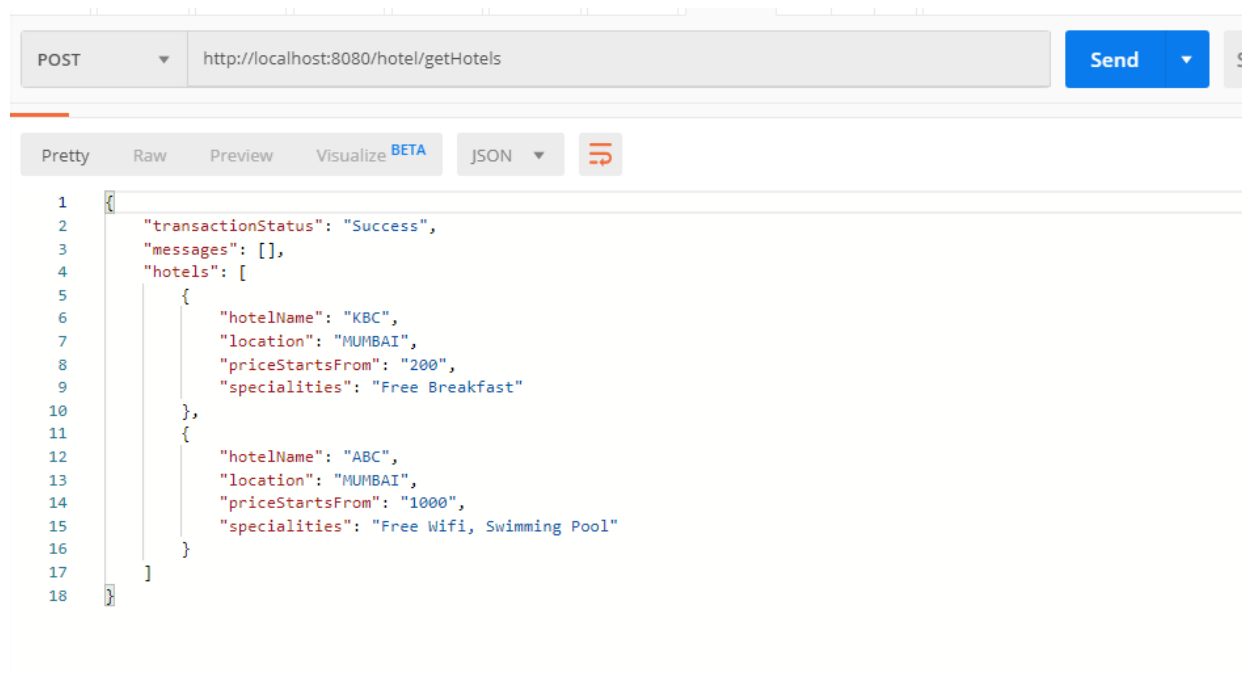
- Request:

```
{
  "hotelId": "",
  "hotelName": "",
  "priceRangeFrom": "100",
  "priceRangeTo": "1200",
  "location": "mumbai",
  "keyword": ""
}
```





- Response:



Similarly try for other scenarios:

## 5. Update records by single or in bulk:

- URL: `http://localhost:8080/hotel/updateHotels`
- RequestMapping: POST
- Request:

```
{
  "updateValues": {
```

```

    "hotelName": "",
    "priceToAddOrSub": "",
    "location": "",
    "addKeywords": ""
  },
  "whenCondition": {
    "hotelId": "",
    "hotelName": "",
    "priceGreaterThan": "",
    "priceLessThan": "",
    "location": "",
    "matchingKeywords": ""
  }
}

```

Description:

- updateValues and whenCondition is mandatory
- In updateValues : any one of the field is mandatory
- In whenCondition : any one of the field is mandatory
- In updateValues:
  - a) if hotelName is provided then it will set hotelName against the records
  - b) if priceToAddOrSub is provided it will add or subtract with respect to price of records
  - c) if location is provided then it will set location against the records
  - d) if addKeywords is provided then it appends to specialities of records
- In whereCondition: (searches record dynamically)
  - a) if hotelId is provided then, it fetches records as per hotelId
  - b) If hotelName is provided then it fetches records with hotelName which contains chars/strings provided as hotelName (for eg: "hotelName":"A", it searches for hotelName which contains "A" in it)
  - c) If priceGreaterThan is provided then it fetches records of hotels having price greater than priceRangeFrom
  - d) If priceLessThan is provided then it fetches records of hotel having price less than priceRangeTo
  - e) If location is provided then it fetches records of hotels whose locations contains chars/strings provided as location (for eg: " location ":"mum", it searches for location which contains "mum" in it)
  - f) If matchingKeywords is provided then it fetches records of hotels whose specialities contains chars/strings provided as matchingKeywords (for eg: "

matchingKeywords ":"wifi", it searches for specialities which contains "wifi" in it)

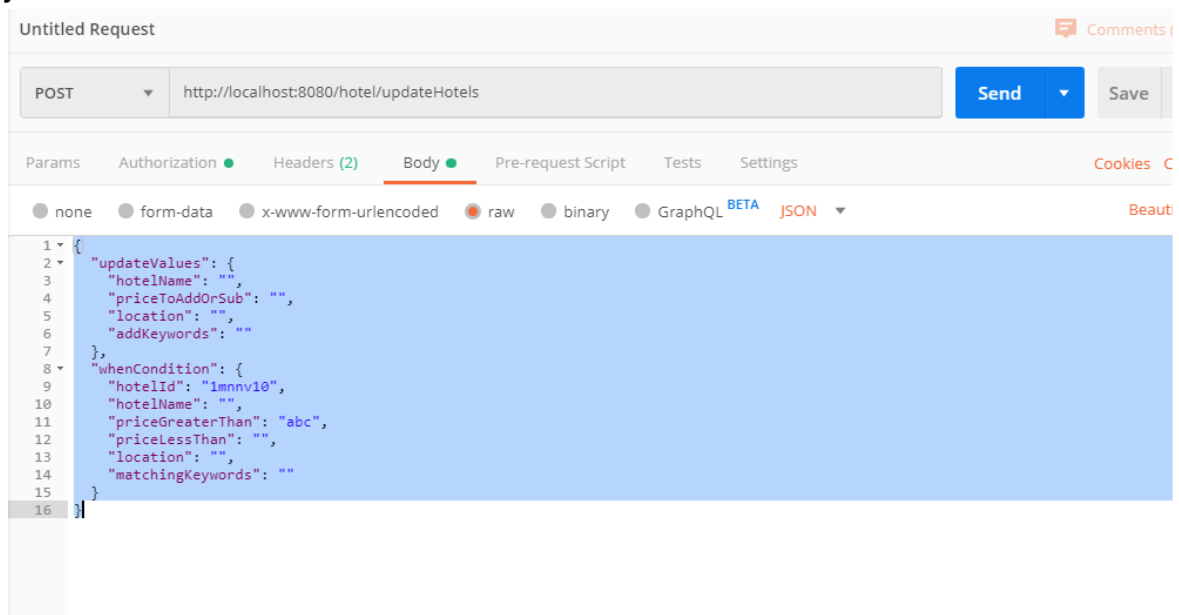
- Validations:
  - In updateValues: priceToAddOrSub should be numeric
  - In whereCondition: hotelId, priceLessThan and priceGreaterThan should be numeric
    - updateValues and whenCondition is mandatory
    - In updateValues : any one of the field is mandatory
  - In whenCondition : any one of the field is mandatory

- **Admin authorization is required, refer step2 (Mandatory)**

- Error Scenarios: validation fails

- Request:

```
{
  "updateValues": {
    "hotelName": "",
    "priceToAddOrSub": "",
    "location": "",
    "addKeywords": ""
  },
  "whenCondition": {
    "hotelId": "1mnnv10",
    "hotelName": "",
    "priceGreaterThan": "abc",
    "priceLessThan": "",
    "location": "",
    "matchingKeywords": ""
  }
}
```



Response:

```
POST http://localhost:8080/hotel/updateHotels

{
  "transactionStatus": "Error",
  "messages": [
    {
      "code": "HOTEL_ID_INVALID",
      "description": "Hotel Id should be in numeric",
      "record": "HotelUpdateWhereParams [hotelId=1mnnv10, hotelName=, priceGreaterThan=abc, priceLessThan=, location=, matchingKeywords=]"
    },
    {
      "code": "PRICE_GREATER_THAN_INVALID",
      "description": "Price Value From should be in numeric",
      "record": "HotelUpdateWhereParams [hotelId=1mnnv10, hotelName=, priceGreaterThan=abc, priceLessThan=, location=, matchingKeywords=]"
    },
    {
      "code": "INVALID_UPDATE_VALUES_CONDITION",
      "description": "Atleast one of the field is mandatory in updateValues",
      "record": "HotelUpdateParams [hotelName=, priceToAddOrSub=, location=, addKeywords=]"
    }
  ]
}
```

Similarly try for other validations

- Happy Path Scenarios

Showing rows 0 - 4 (5 total, Query took 0.0020 seconds.)

```
SELECT * FROM `hotel` WHERE 1
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PH]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	HOTEL_ID	HOTEL_NAME	LOCATION	PRICE	SPECIALITIES
<input type="checkbox"/> Edit Copy Delete	104	KBC	MUMBAI	200	Free Breakfast
<input type="checkbox"/> Edit Copy Delete	106	ABC	MUMBAI	1000	Free Wifi, Swimming Pool
<input type="checkbox"/> Edit Copy Delete	107	LMO	DELHI	2000	
<input type="checkbox"/> Edit Copy Delete	108	DEF	MUMBAI	1500	Free Wifi, Free breakfast
<input type="checkbox"/> Edit Copy Delete	109	LPQRTS	PUNE	3500	Free Wifi, Free breakfast

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

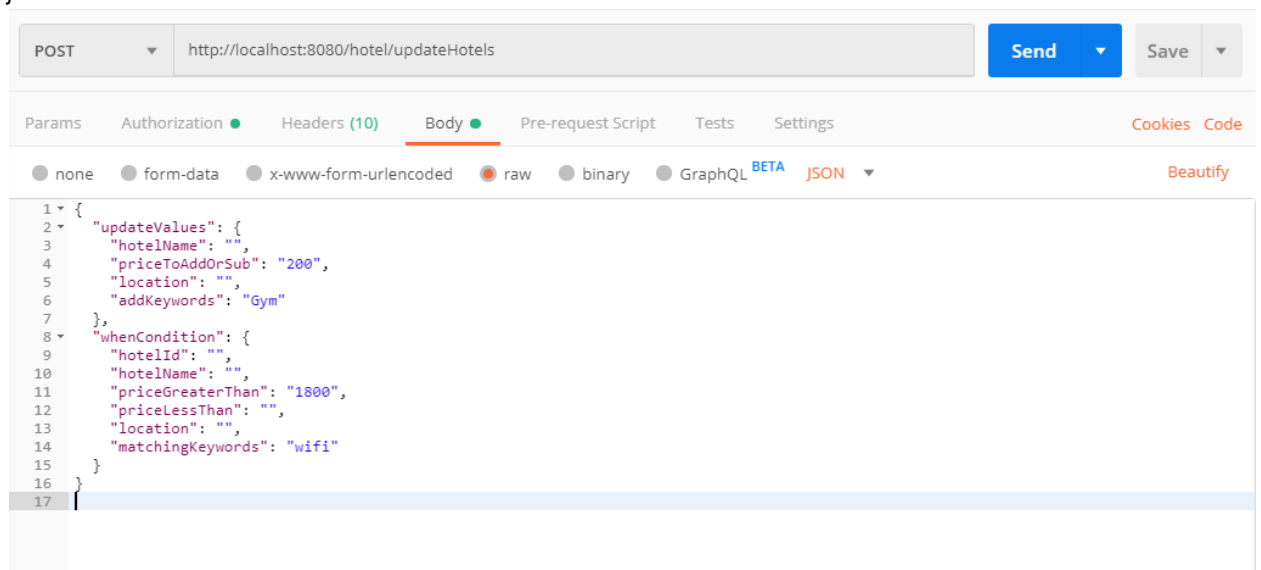
As per data in table:

example 1: increase the price of hotels by 200 and add extra specialities ="gym"  
where speciliaties contains "wifi" and price > 1800.

As per data in table only 1 record will be updates:

Request:

```
{
  "updateValues": {
    "hotelName": "",
    "priceToAddOrSub": "200",
    "location": "",
    "addKeywords": "Gym"
  },
  "whenCondition": {
    "hotelId": "",
    "hotelName": "",
    "priceGreaterThan": "1800",
    "priceLessThan": "",
    "location": "",
    "matchingKeywords": "wifi"
  }
}
```



Response:

```
14     matchingkeywords : wlt1
15   }
16 }
17
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 1609ms Size: 269 B Save

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "transactionStatus": "Success",
3   "messages": [
4     {
5       "code": "RECORDS_UPDATED",
6       "description": "Records have been updated successfully",
7       "record": "1"
8     }
9   ]
10 }
```

Bootcamp

check in db:

✓ Showing rows 0 - 4 (5 total, Query took 0.0010 seconds.)

```
SELECT * FROM `hotel` WHERE 1
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	HOTEL_ID	HOTEL_NAME	LOCATION	PRICE	SPECIALITIES
<input type="checkbox"/> Edit Copy Delete	104	KBC	MUMBAI	200	Free Breakfast
<input type="checkbox"/> Edit Copy Delete	106	ABC	MUMBAI	1000	Free Wifi, Swimming Pool
<input type="checkbox"/> Edit Copy Delete	107	LMO	DELHI	2000	
<input type="checkbox"/> Edit Copy Delete	108	DEF	MUMBAI	1500	Free Wifi, Free breakfast
<input type="checkbox"/> Edit Copy Delete	109	LPQRTS	PUNE	3700	Free Wifi, Free breakfast, Gym

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

Similarly try combinations for others:

## 6. Delete records by single or in bulk

- URL: <http://localhost:8080/hotel/deleteHotels>
- RequestMapping : POST
- Request:

```
{
  "hotelId":"","
  "hotelName":"","
  "priceGreaterThan":"","
  "priceLessThan":"","
  "location":"","
  "noMatchingKeywords":""
}
```

#### Description:

- a) if hotelId is provided then, it fetches records as per hotelId
- b) If hotelName is provided then it fetches records with hotelName which contains chars/strings provided as hotelName (for eg: "hotelName":"A", it searches for hotelName which contains "A" in it)
- c) If priceGreaterThan is provided then it fetches records of hotels having price greater than priceRangeFrom
- d) If priceLessThan is provided then it fetches records of hotel having price less than priceRangeTo
- e) If location is provided then it fetches records of hotels whose locations contains chars/strings provided as location (for eg: " location ":"mum", it searches for location which contains "mum" in it)
- f) If noMatchingKeywords is provided then it fetches records of hotels whose specialities **doesnot** contains chars/strings provided as matchingKeywords (for eg: " matchingKeywords ":"wifi", it searches for specialities which doesnot contains "wifi" in it)

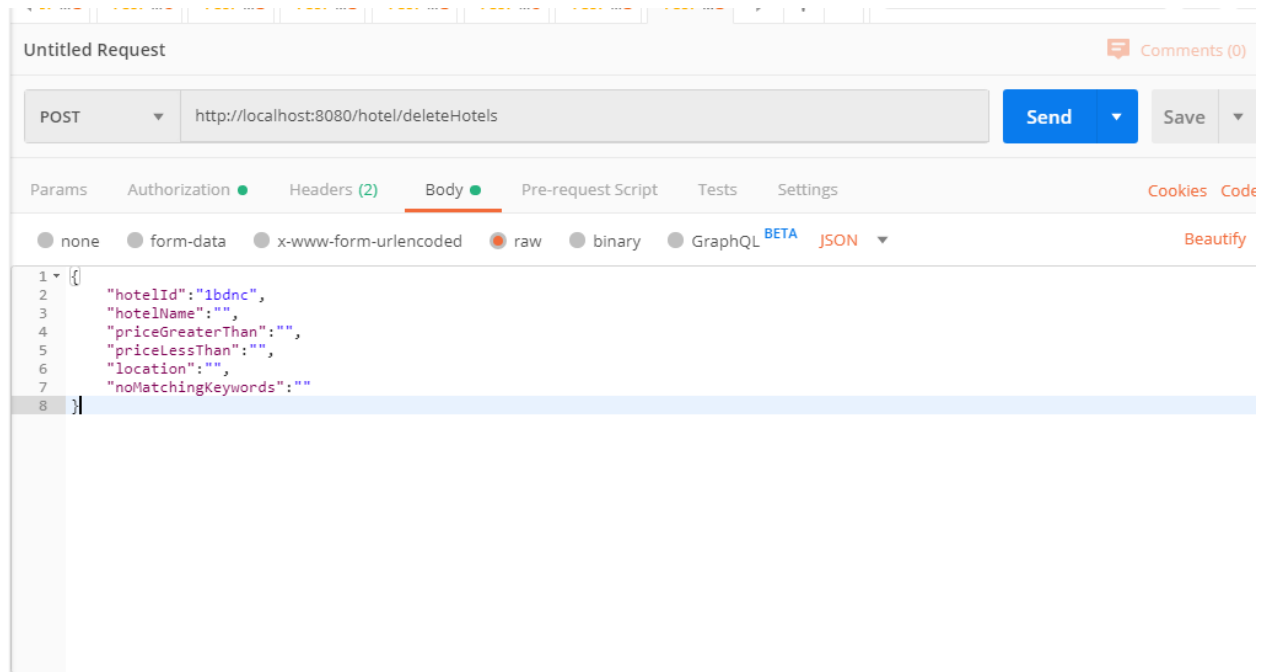
- Validations:
  - hotelId, priceGreaterThan and priceLessThan should be numeric
  - Any one of the request field is mandatory

- **Admin authorization is required, refer step2 (Mandatory)**

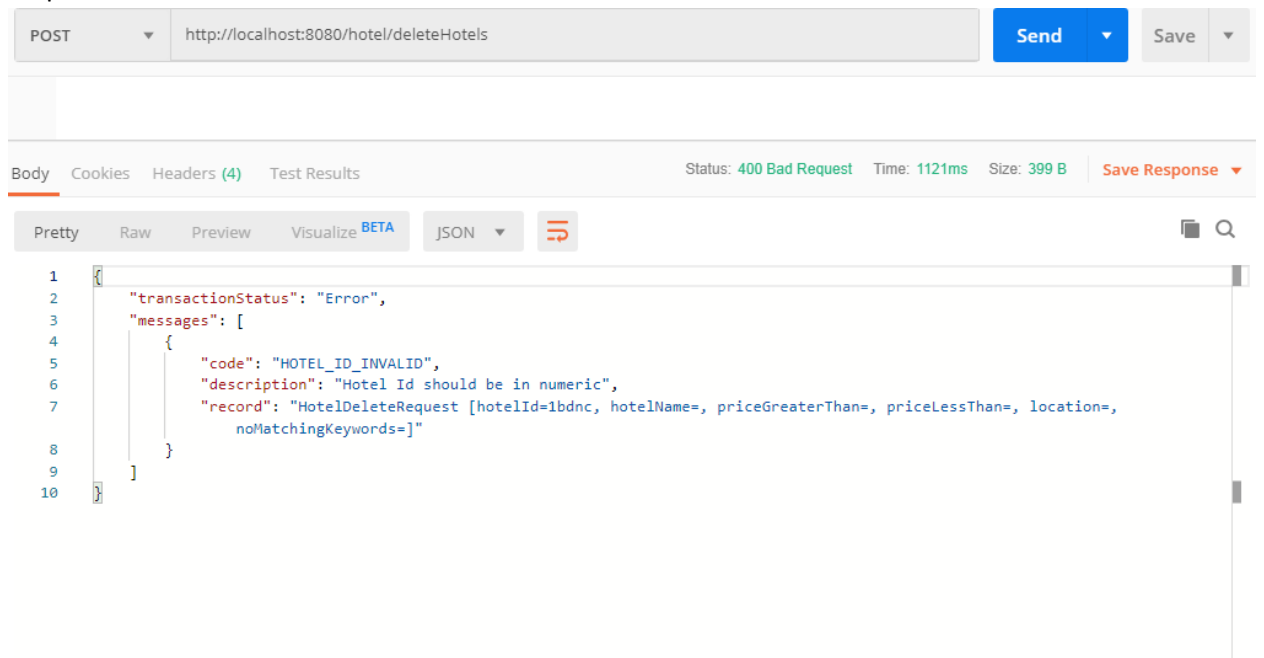
- Error Scenarios: validation fails

- Request:

```
{
  "hotelId":"1bdnc",
  "hotelName":"","
  "priceGreaterThan":"","
  "priceLessThan":"","
  "location":"","
  "noMatchingKeywords":""
}
```



Response:



Similarly try for other validation scenarios

- delete records which have price less than 1200 and specialities does not contains "wifi"



As per table first record should be deleted

Show query box

✓ Showing rows 0 - 2 (3 total, Query took 0.0030 seconds.)

```
SELECT * FROM `hotel` WHERE 1
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[C\]](#)

```
UPDATE `hotel` SET `SPECIALITIES` = 'Free Gym, Swimming Pool' WHERE `hotel`.`HOTEL_ID` = 106;
```

[\[Edit inline\]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

				HOTEL_ID	HOTEL_NAME	LOCATION	PRICE	SPECIALITIES
<input type="checkbox"/>				106	ABC	MUMBAI	1000	Free Gym, Swimming Pool
<input type="checkbox"/>				108	DEF	MUMBAI	1500	Free Wifi, Free breakfast
<input type="checkbox"/>				109	LPQRTS	PUNE	3700	Free Wifi, Free breakfast, Gym

↑ ☐ Check all | With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

Request:

```
{
  "hotelId": "",
  "hotelName": "",
  "priceGreaterThan": "",
  "priceLessThan": "1200",
  "location": "",
  "noMatchingKeywords": "wifi"
}
```

POST http://localhost:8080/hotel/deleteHotels

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON Beautify

```
1 {
2   "hotelId": "",
3   "hotelName": "",
4   "priceGreaterThan": "",
5   "priceLessThan": "1200",
6   "location": "",
7   "noMatchingKeywords": "wifi"
8 }
9
```

Status: 200 OK Time: 1736ms Size: 268 B

Response:

Body Cookies Headers (3) Test Results Status: 200 OK Time: 1736ms Size: 268 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "transactionStatus": "Success",
3   "messages": [
4     {
5       "code": "RECORDS_DELETED",
6       "description": "Records have been deleted successfully",
7       "record": ""
8     }
9   ]
10 }
```



Check db

Show query box

✓ Showing rows 0 - 1 (2 total, Query took 0.0020 seconds.)

```
SELECT * FROM `hotel` WHERE 1
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create view\]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

				HOTEL_ID	HOTEL_NAME	LOCATION	PRICE	SPECIALITIES
<input type="checkbox"/>				108	DEF	MUMBAI	1500	Free Wifi, Free breakfast
<input type="checkbox"/>				109	LPQRTS	PUNE	3700	Free Wifi, Free breakfast,Gym

☐ Check all | With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Similarly try for other combinations