

Purpose: Planning and maintaining a checklist of things to do is critical as we need to perform personal, professional and leisure activities in our lives. I felt that such a microservice which will help me manage tasks and sync with my calendar(Google Calendar) will be helpful.

Project Overview: Planning Application

Create a microservice, to manage tasks and connect them to Google calendar.

1. Create, view and manage tasks and store them into the user's database.
2. Create events in the user's Google calendar referring to tasks from the database.

Microservices specific to database:

1. Create tasks, view them and mark as done.

Microservices specific to Google Calendar:

1. Retrieve Events from Google Calendar.
2. Create events in Google Calendar.

Technologies used:

Application to use “Google API” using Spring Boot and OAuth2.

1. Spring Boot
2. Rest API
3. Google Calendar API (Maven Dependencies)
4. A Google account with Google Calendar enabled

References: To register a Google App perform the following steps

<https://developers.google.com/calendar/quickstart/java?authuser=1>

<https://console.developers.google.com>

Maven Dependencies:

```
<dependency>

    <groupId>com.google.apis</groupId>
    <artifactId>google-api-services-calendar</artifactId>
    <version>v3-rev224-1.22.0</version>
</dependency>

<!-- Gmail Maven Dependency-->
<dependency>
    <groupId>com.google.apis</groupId>
    <artifactId>google-api-services-gmail</artifactId>
```

```
<version>v1-rev65-1.18.0-rc</version>
</dependency>
```

Create a project in Google developer console <https://console.developers.google.com>
We need to create one having permission to update or edit Events.

Credentials for Google API

console.developers.google.com/apis/credentials?authuser=1&project=ucsc-271906&folder=&organizationId=

Google APIs UCSC

APIs & Services

Credentials + CREATE CREDENTIALS DELETE

Create credentials to access your enabled APIs. [Learn more](#)

To protect you and your users, your consent screen and application need to be verified by Google. [Learn more](#) [CONFIGURE CONSENT SCREEN](#)

API Keys

<input type="checkbox"/>	Name	Creation date ↓	Restrictions	Key	Usage with all services (last 30 days) ?
No API keys to display					

OAuth 2.0 Client IDs

<input type="checkbox"/>	Name	Creation date ↓	Type	Client ID	
<input type="checkbox"/>	Web client 1	Mar 21, 2020	Web application	763435151653-8vd2...	
<input type="checkbox"/>	Desktop client 1	Mar 21, 2020	Desktop	763435151653-bh9c...	

Service Accounts [Manage service accounts](#)

<input type="checkbox"/>	Email	Name ↑	Usage with all services (last 30 days) ?
No service accounts to display			

Download the JSON file and put in our project.

Database Info:

```
mysql> show Databases;
+-----+
| Database |
+-----+
| db_example |
| dev_db |
| event_planner_task_db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
7 rows in set (0.02 sec)

mysql>
```

Table Info:

```
mysql> DESCRIBE task_info;
```

Field	Type	Null	Key	Default	Extra
task_id	int(11)	NO	PRI	NULL	auto_increment
task_name	varchar(255)	NO		NULL	
task_description	varchar(255)	NO		NULL	
Task_date	date	NO		NULL	
created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
mark_as_done	tinyint(1)	YES		0	

6 rows in set (0.00 sec)

```
mysql> select * from task_info;
```

task_id	task_name	task_description	Task_date	created_at	mark_as_done
1	Outing	Visit Almeden Lake	2020-04-05	2020-03-31 05:34:46	0
2	Laundry	Go for Laundry	2020-04-06	2020-03-31 05:35:45	0
3	Groceries	Go to Safeway	2020-04-07	2020-03-31 05:39:48	0
4	Groceries	Go to India Bazar	2020-04-08	2020-03-31 05:40:22	1
6	Groceries	Go to India Bazar	2020-04-08	2020-03-31 05:50:03	1

5 rows in set (0.00 sec)

Rest APIs:

These results are derived through Postman

1. Print All Task

Url: <http://localhost:8080/printAllTasks>

The screenshot shows a Postman interface with a GET request to `http://localhost:8080/printAllTasks`. The response is a JSON array of 6 tasks. The response status is 200 OK, and the response body is displayed in a pretty-printed JSON format.

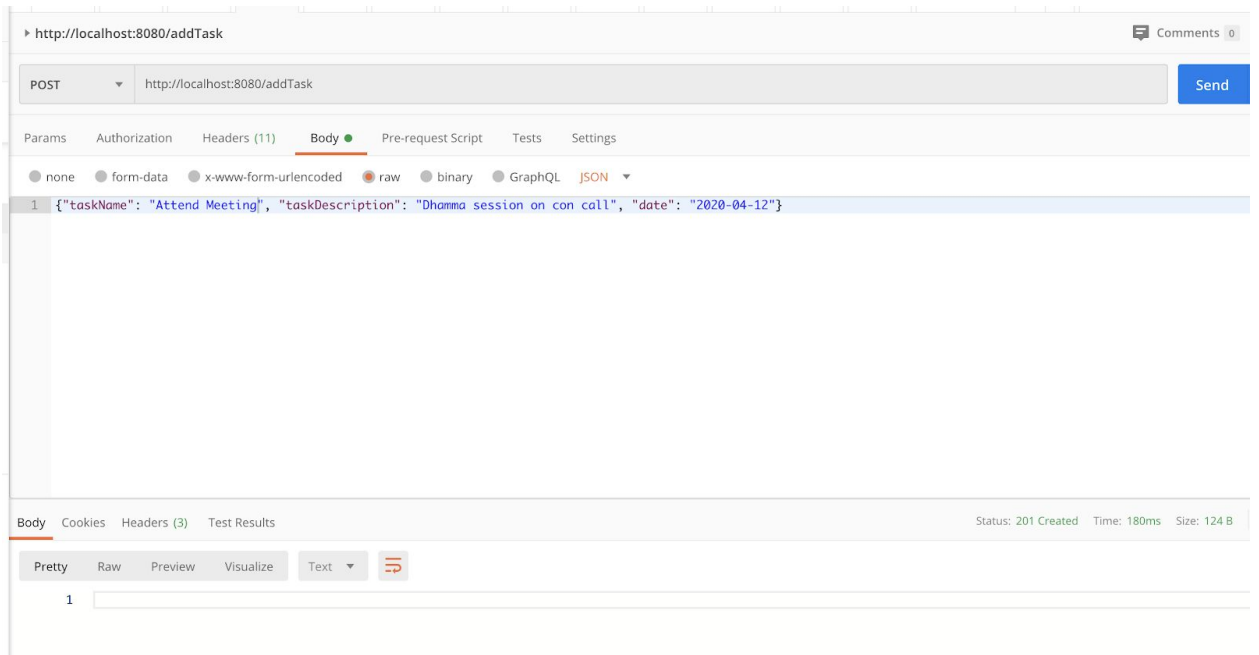
```
1 SELECT * from task_info;
2 1, Outing, Visit Almeden Lake, 2020-04-04, false, 2020-03-30 22:34:46.0
3 2, Laundry, Go for Laundry, 2020-04-05, false, 2020-03-30 22:35:45.0
4 3, Groceries, Go to Safeway, 2020-04-06, false, 2020-03-30 22:39:48.0
5 4, Groceries, Go to India Bazar, 2020-04-07, true, 2020-03-30 22:40:22.0
6 6, Groceries, Go to India Bazar, 2020-04-07, true, 2020-03-30 22:50:03.0
7
```

2. Add Task:

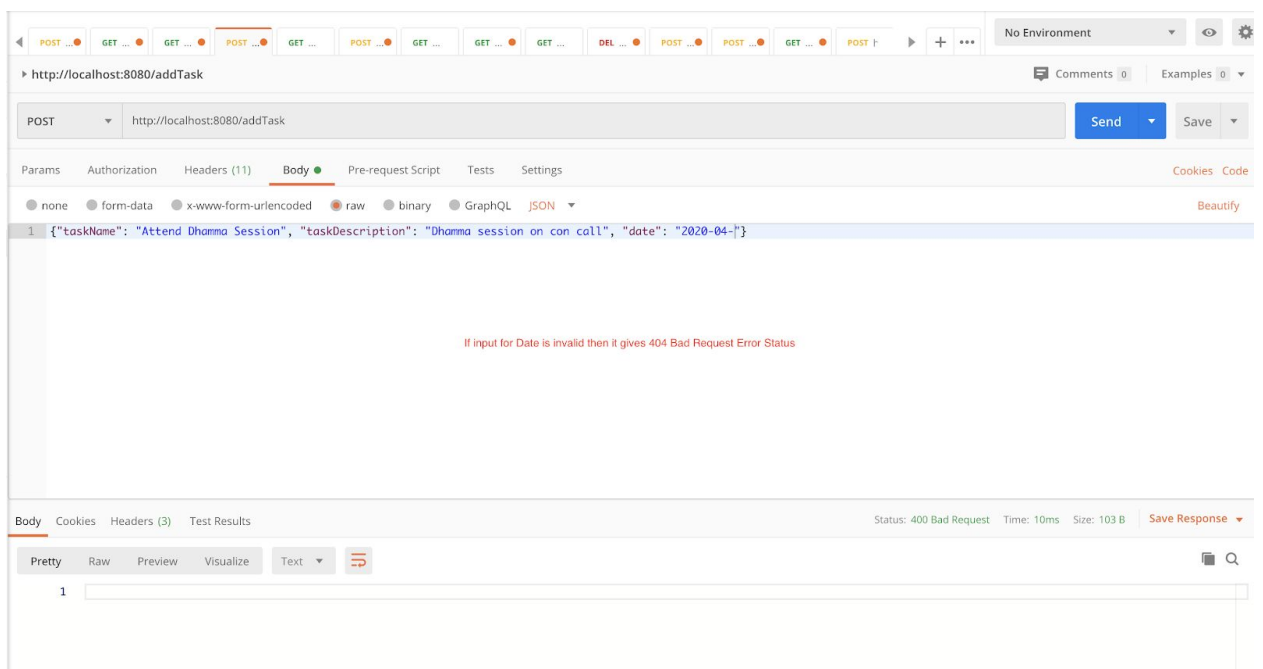
Url: <http://localhost:8080/addTask>

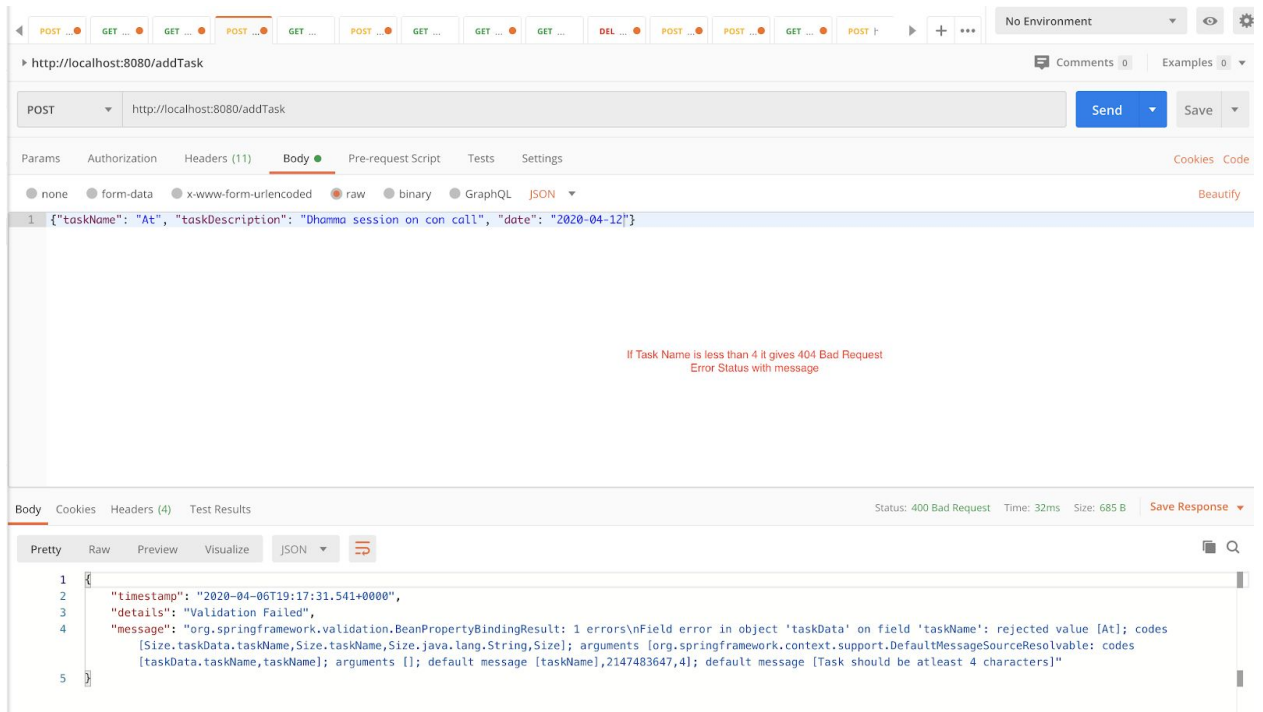
Raw data:

```
{"taskName": "Get Medicines", "taskDescription": "Medicines from Walgreen", "date": "2020-02-28"}
```



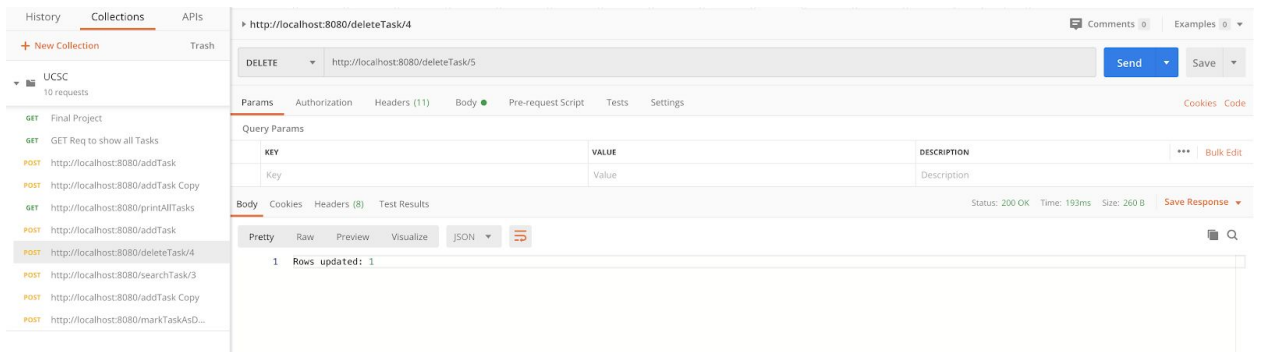
Exception Handling is done for Invalid Date or Task Name with less than 4 characters.





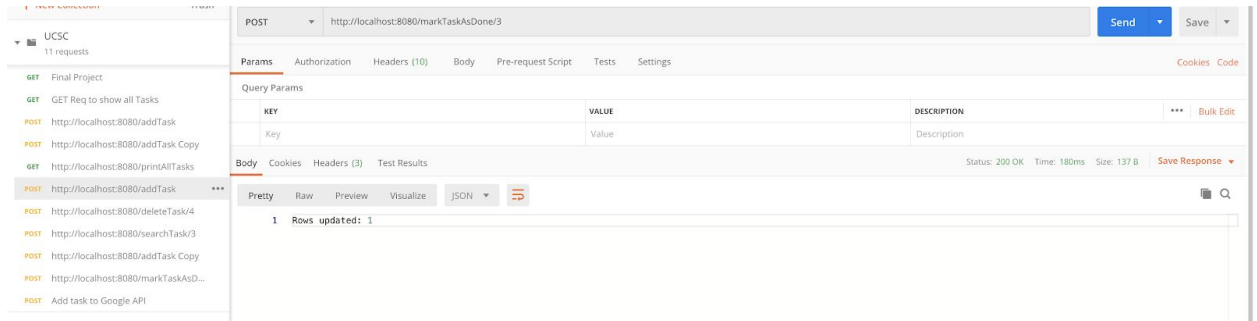
3. Delete task with specified id:

Url: <http://localhost:8080/deleteTask/5>

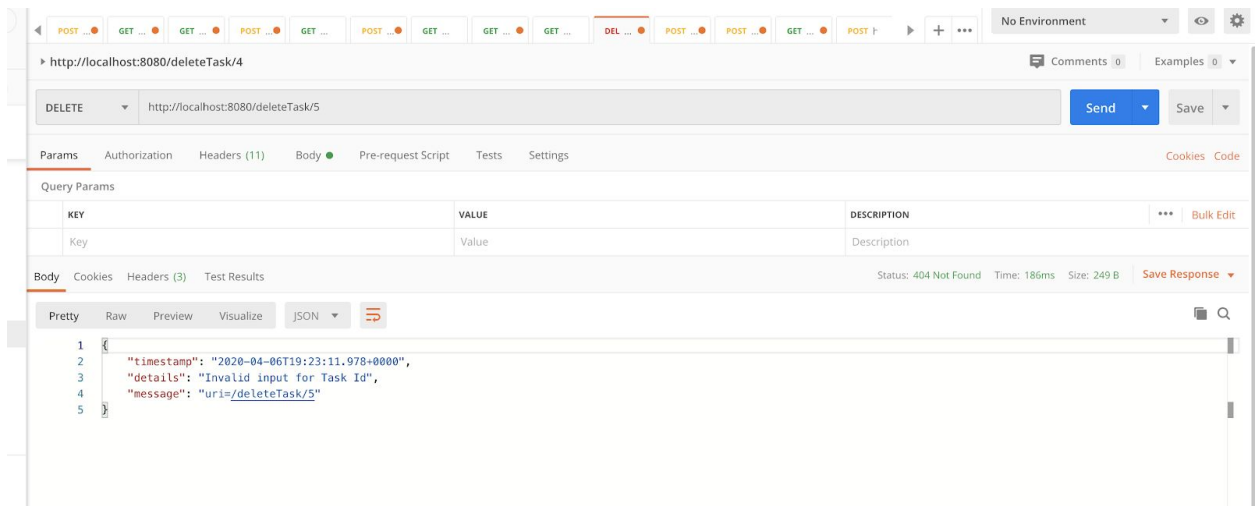


4. Mark Task as Done:

Url: <http://localhost:8080/markTaskAsDone/3>

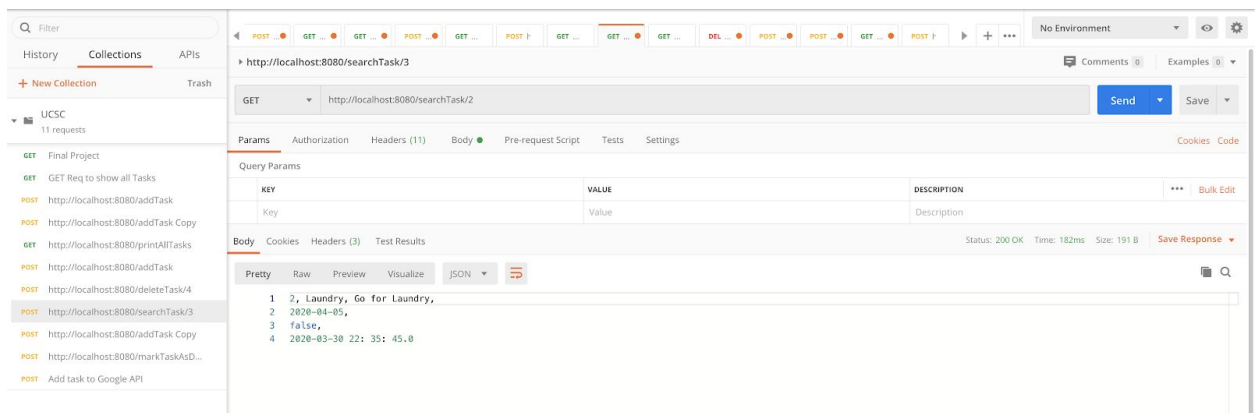


Delete Task Error for invalid Task Id:



5. Search Task:

Url: <http://localhost:8080/searchTask/2>



Search Task Error for invalid Task Id:

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/searchTask/5`. The response status is 404 Not Found. The response body is displayed in JSON format:

```
1 {
2   "timestamp": "2020-04-06T19:25:31.121+0000",
3   "details": "Invalid input for Task Id",
4   "message": "uri=/searchTask/5"
5 }
```

6. Create Event in Google Calendar:

Url: <http://localhost:8080/createCalendarEvent/3>

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/createCalendarEvent/3`. The response status is 200 OK. The response body is displayed in JSON format:

```
1 Event created in Google Calene
```

