

Lab 12 Calling and Developing API using Node.js

Given: November 7, 2018

Deadline: November 15, 2018

Lab 12 Calling and Developing API using Node.js	1
What should be done within lab hours	2
Problem 1 (2 points) Calling Google Map Place API	2
Problem 2 (3 points) Developing Tasks API	2
What can be done after lab hours	8
Problem 3 (2 points) Calling Google Map Directions API	8
Problem 4 (3 points) Developing Person API	9

What should be done within lab hours

Problem 1 (2 points) Calling Google Map Place API

Using Node.js to call <https://maps.googleapis.com/maps/api/place/textsearch/json?query=restaurants%20in%20Khon%20Kaen&key=<key>>



Figure 1. The result of calling Google Map Place API

Problem 2 (3 points) Developing Tasks API

Follow the slide [Calling and Developing RESTful APIs Using Node.js](#) to develop todoListApi

Then, demonstrate that you can successfully develop and test the API by submitting the zip folder of todoListApi that contains all the subdirectories and files as well as the captured screens of the results of performing HTTP methods on various URLs of the API as shown in Figure 2-11.

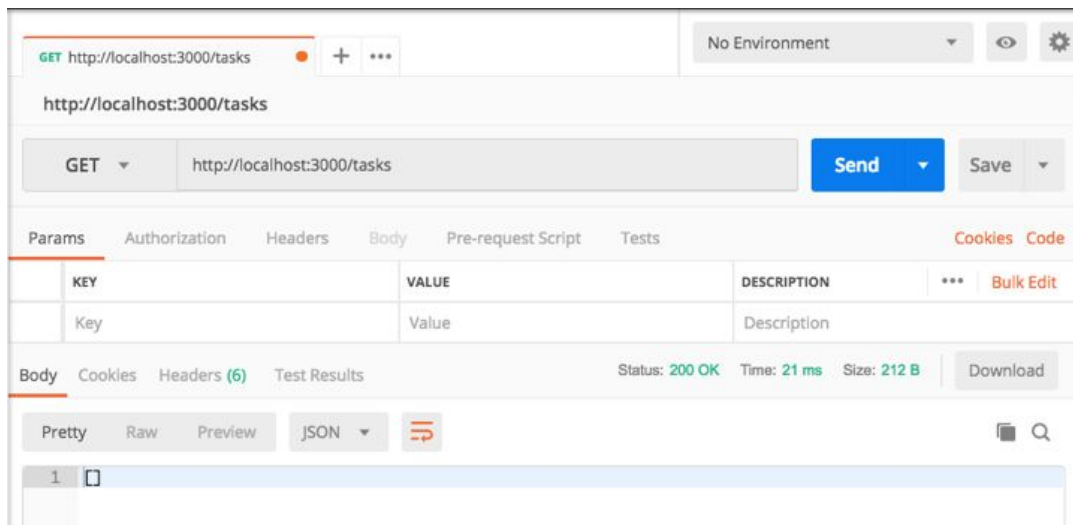


Figure 2: The API originally displayed no task in the list

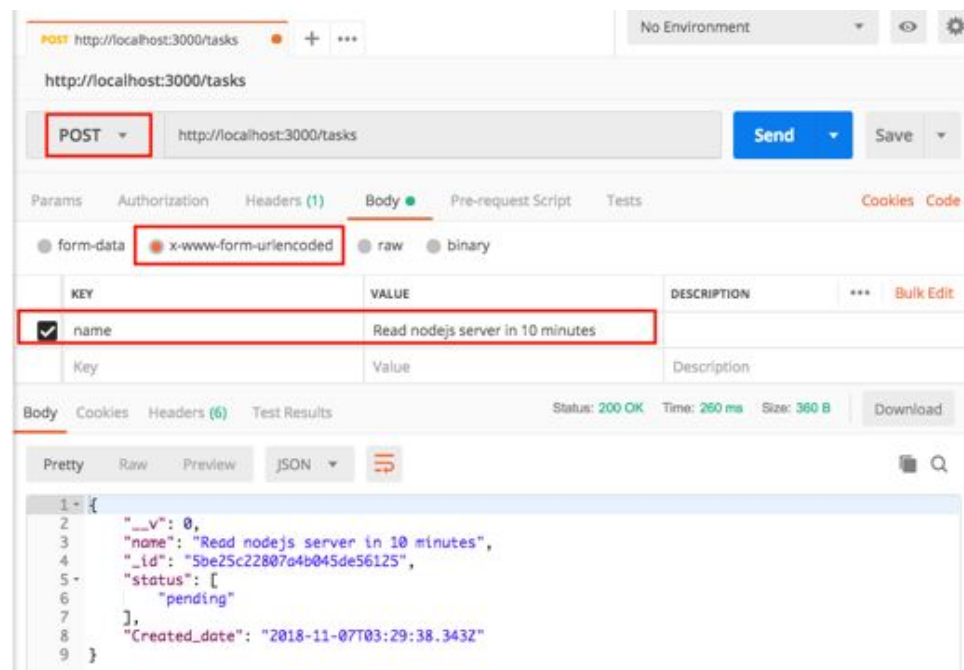


Figure 3: Using POST method to add a new task

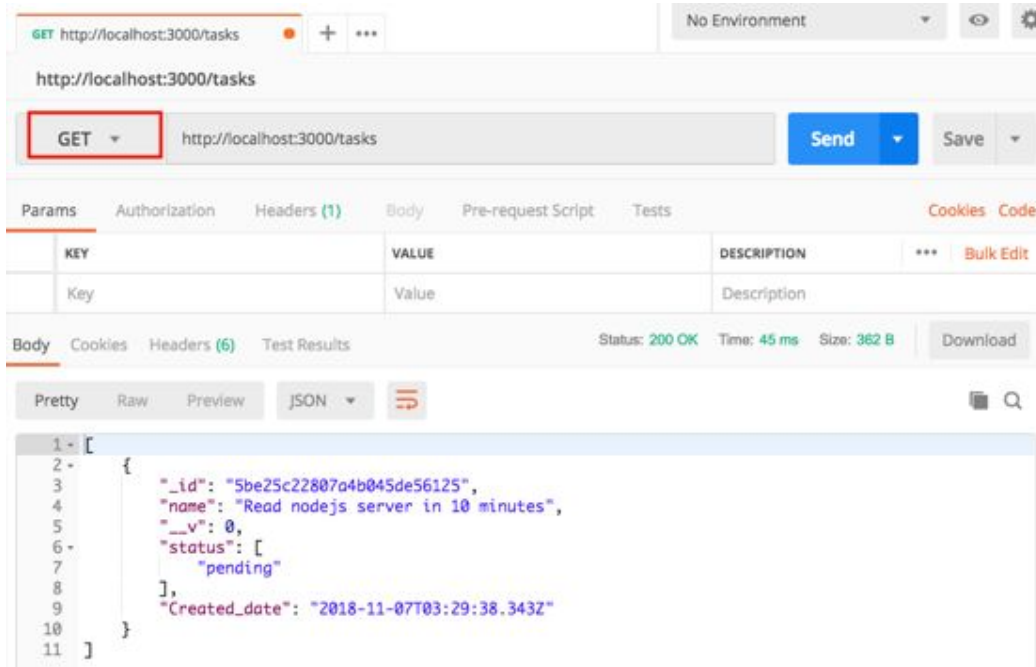


Figure 4: Using GET method to display all tasks including a newly added task

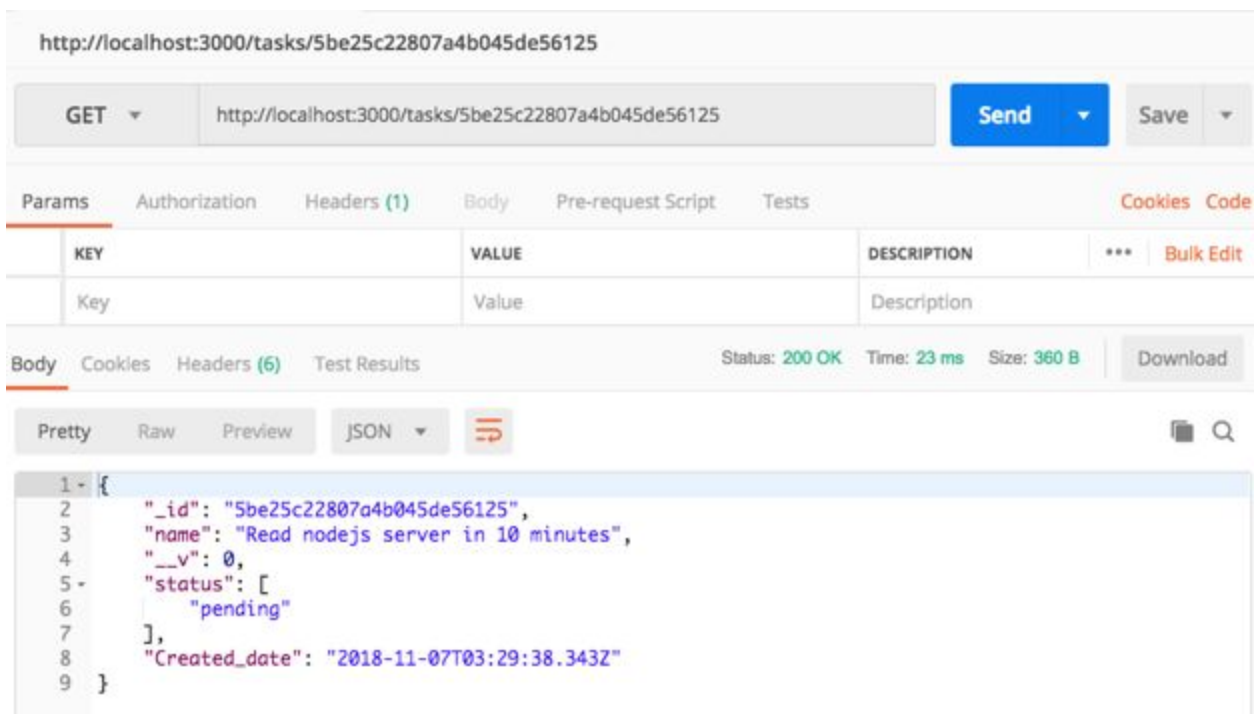


Figure 5: Using GET method with the task ID to see the detail of the task

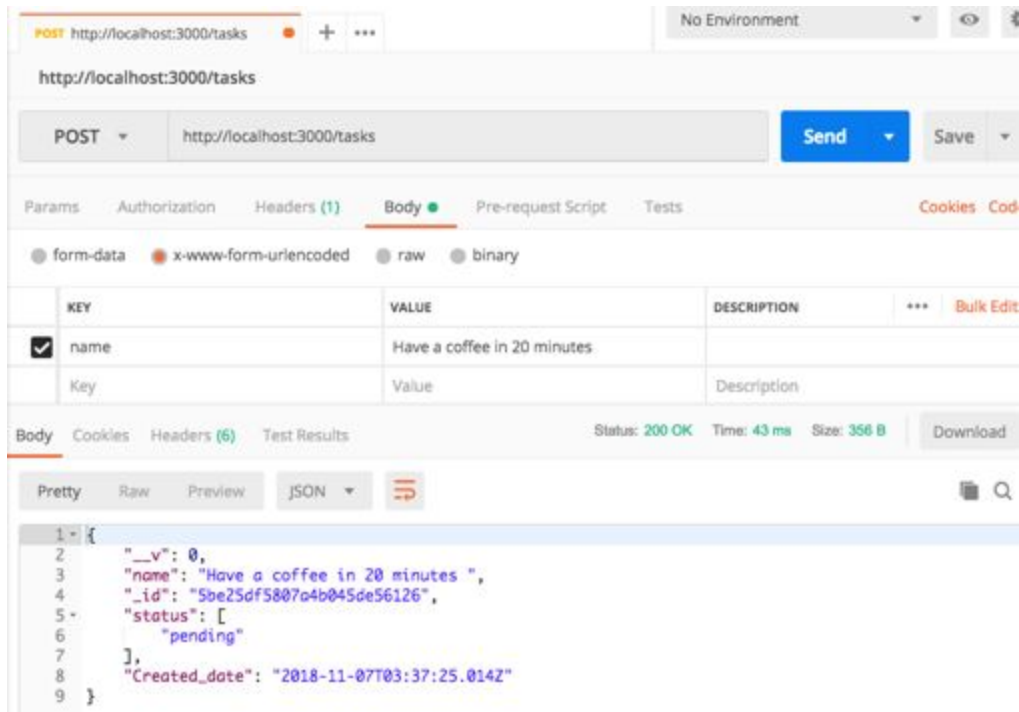


Figure 6: Using POST to add a new task that uses a default created_date which is now

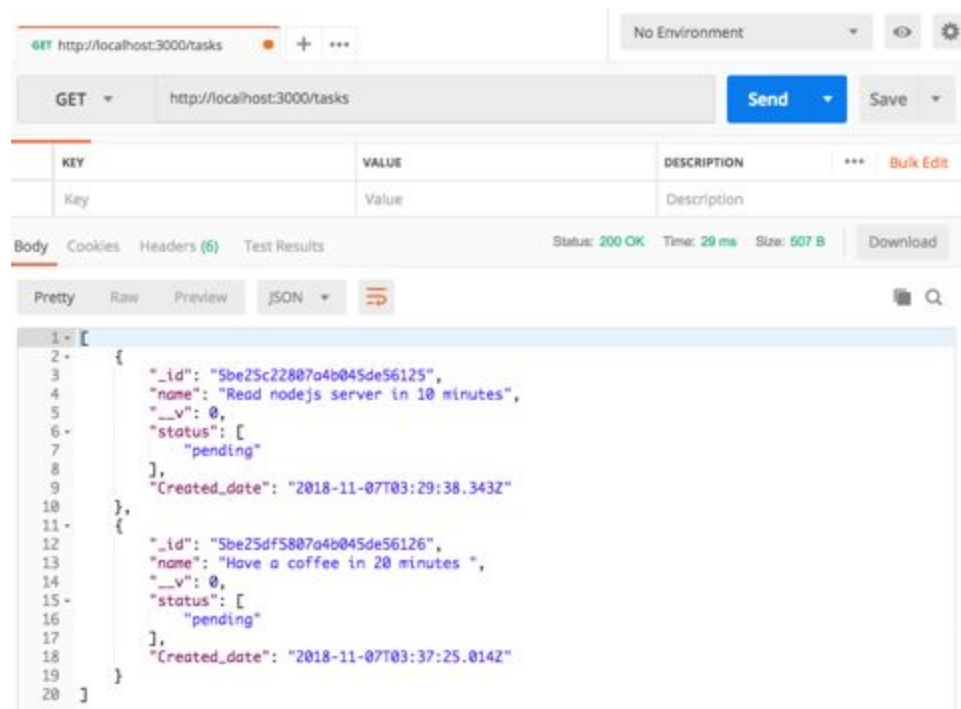


Figure 7: Using GET to display all tasks

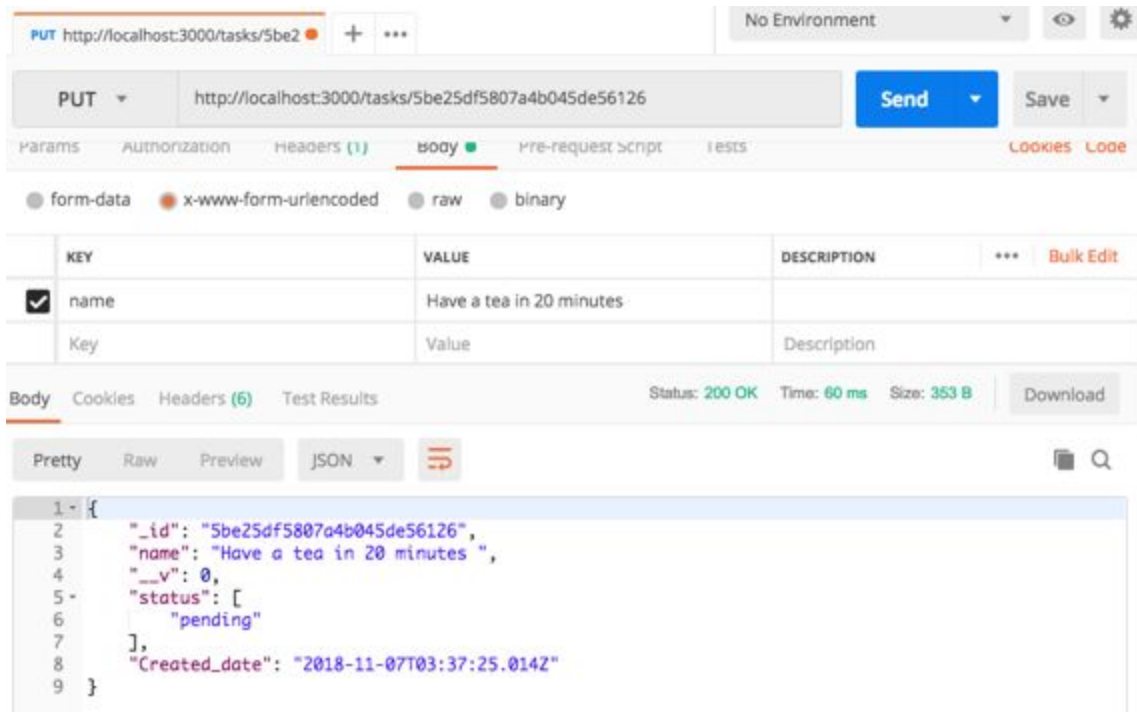


Figure 8: Using PUT to modify the task with the given ID

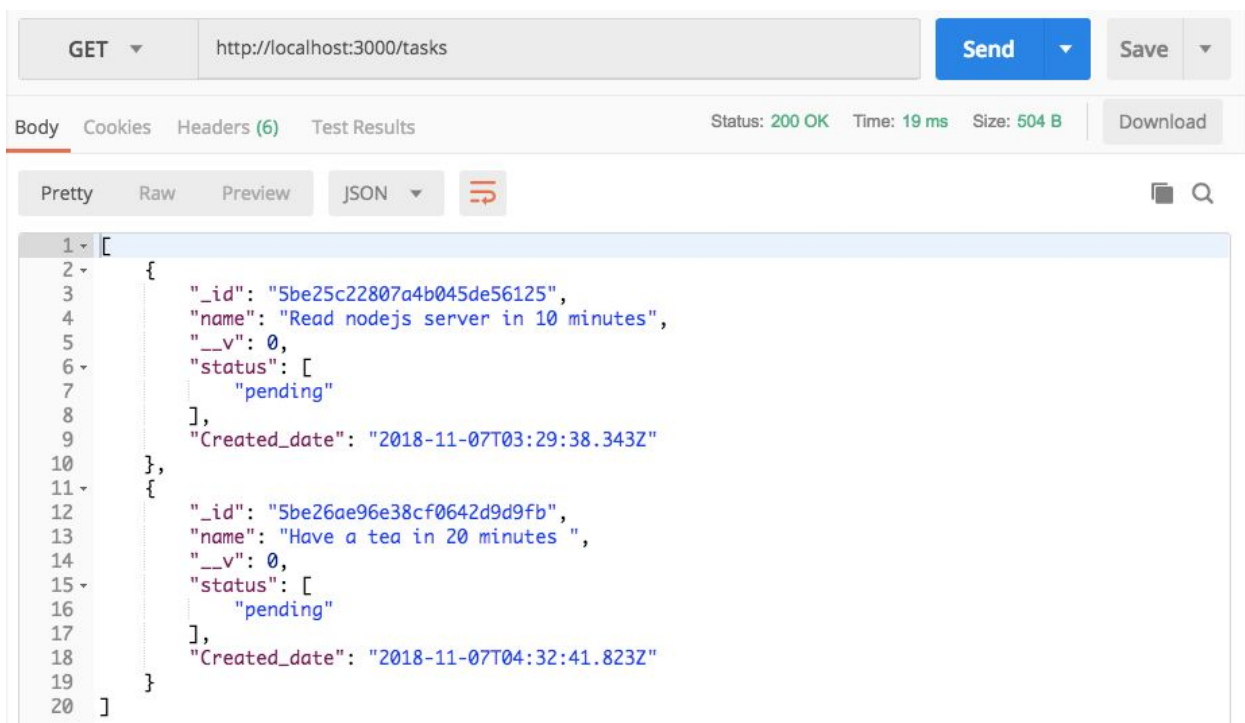


Figure 9: Using GET to display tasks including the updated task

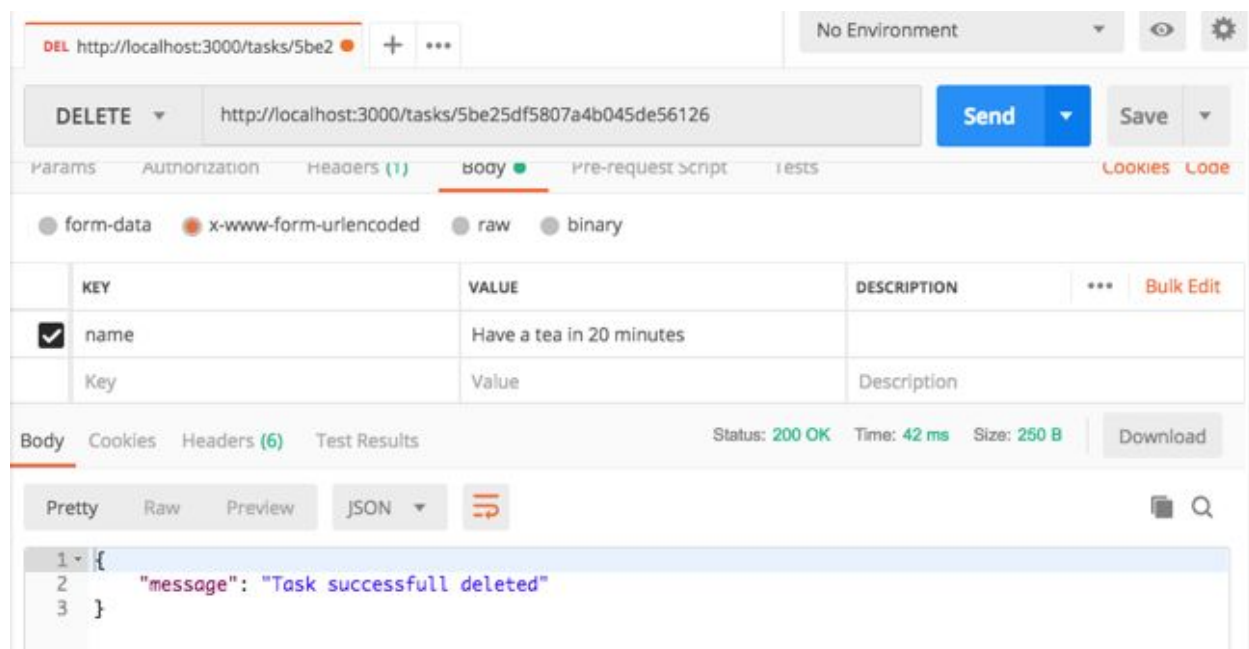


Figure 10: Using DELETE to remove the specified task

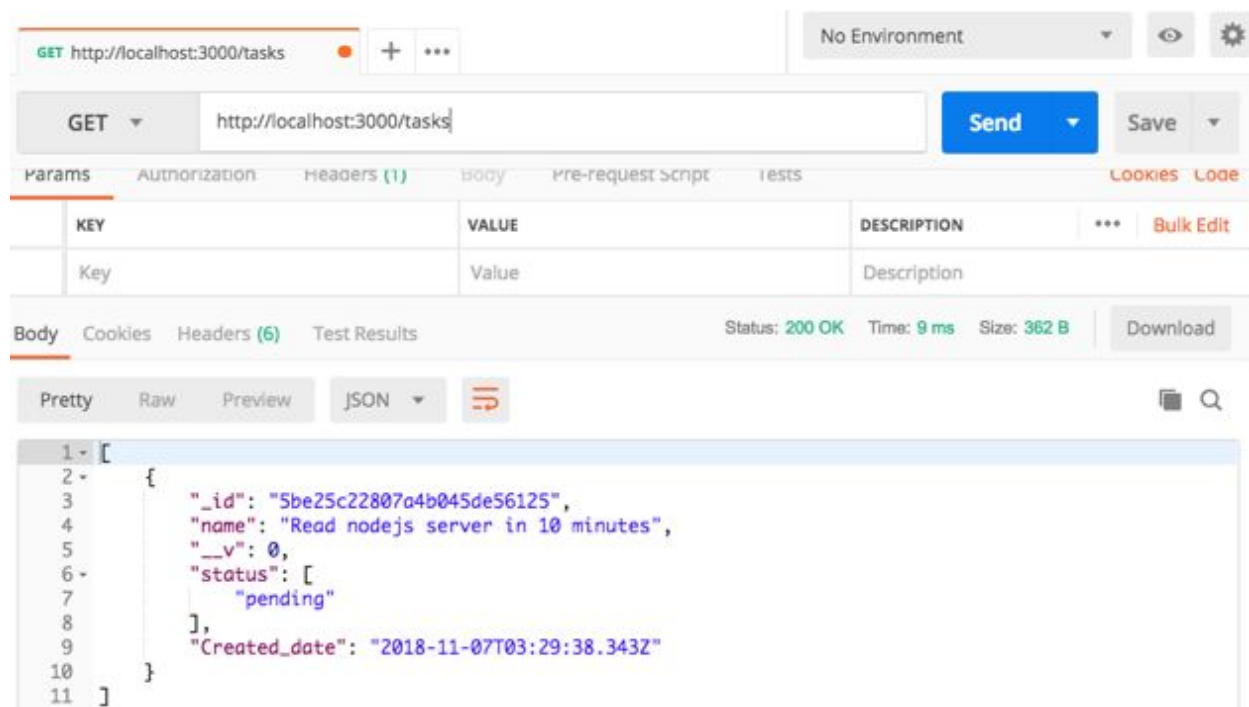


Figure 11: Using GET method to display all tasks which shows that the previous task is successfully deleted

What can be done after lab hours

Problem 3 (2 points) Calling Google Map Directions API

Calling Google Map Direction to know the routes from Khon Kaen to Bangkok as can be specified using this URL

<https://maps.googleapis.com/maps/api/directions/json?origin=Khon%20Kaen&destination=Bangkok&key=<key>>

Then generate the output as shown in Figure 12. Note that the text of each item is from the value of html_instructions of each step and the distance is displayed in blue which its value is obtained from the value of distance text.

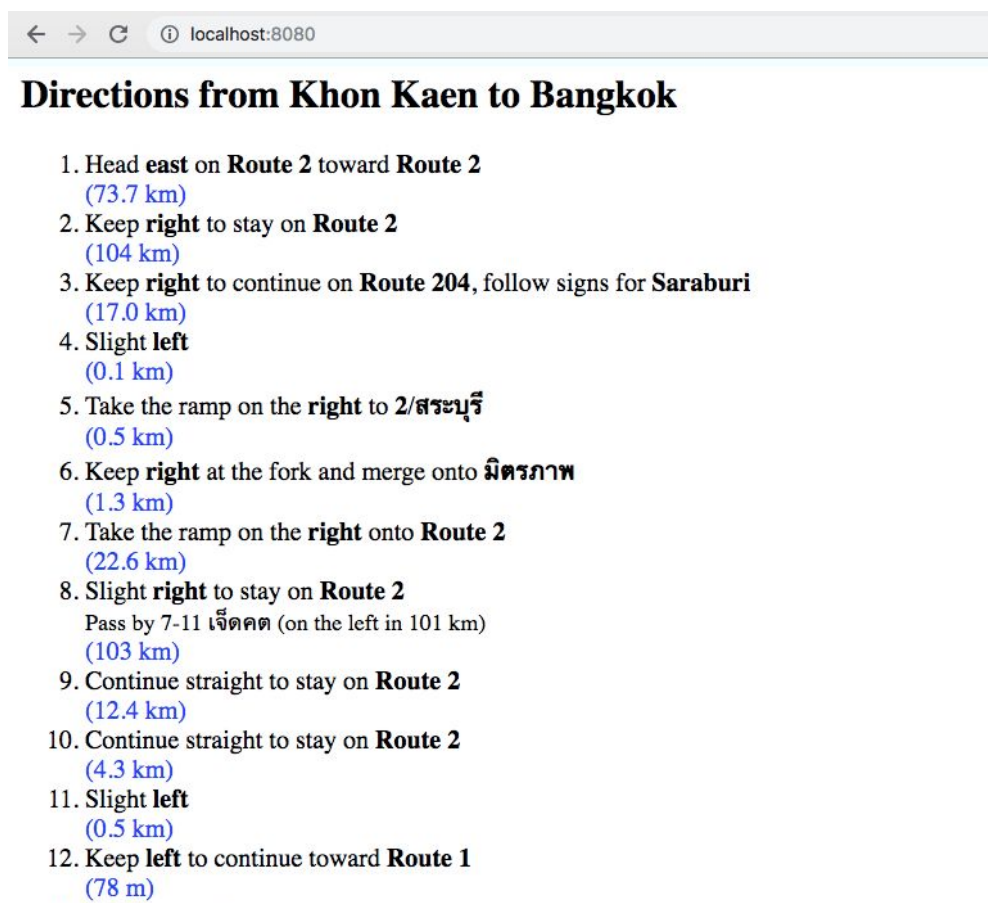


Figure 12: The routes and directions from Khon Kaen to Bangkok

Problem 4 (3 points) Developing Person API

Develop Person API where a person must have name and weight while a person has 'country' as optional with the default value as 'Thailand'

The sample test result with HTTP GET and POST with URL `http://localhost:3000/persons` and HTTP GET, PUT, and DELETE with URL `http://localhost:3000/persons/<personID>`

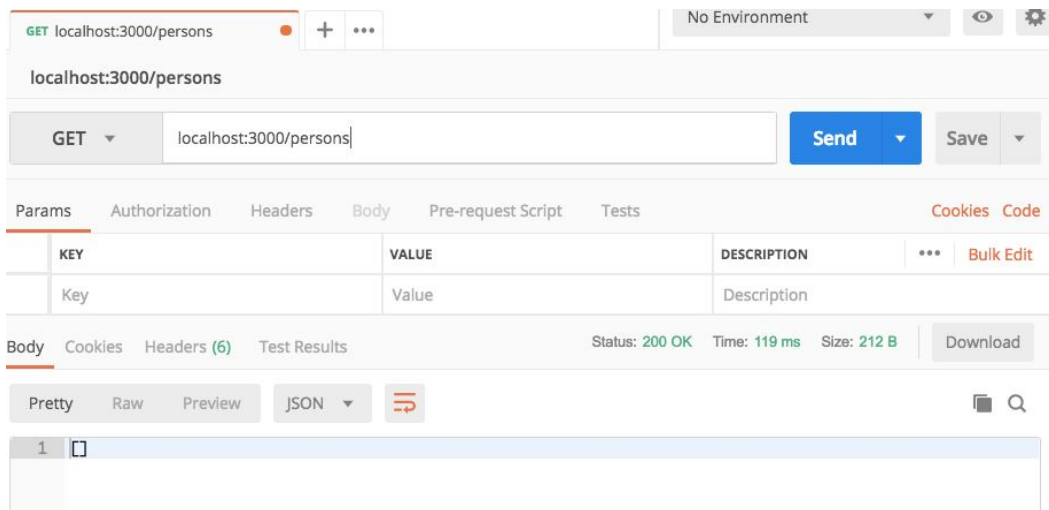


Figure 13: Using GET on <http://localhost:3000/persons> to see the current list of persons

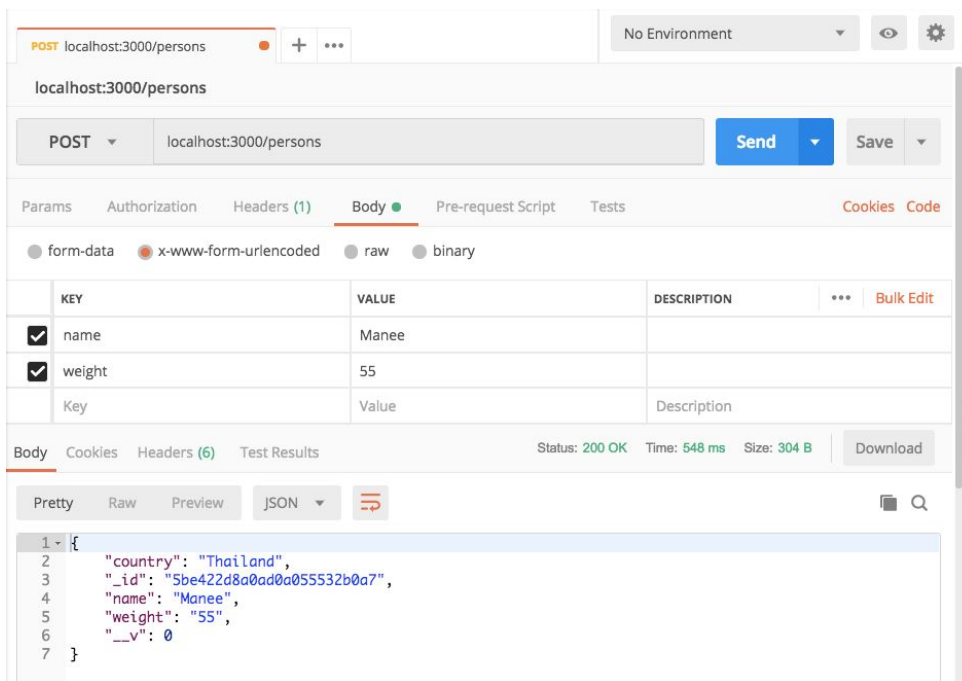


Figure 14: Using POST on <http://localhost:3000/persons> to add the first person

GET localhost:3000/persons

localhost:3000/persons

GET localhost:3000/persons Send Save

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

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 40 ms Size: 306 B Download

Pretty Raw Preview JSON

```

1 [
2   {
3     "country": "Thailand",
4     "_id": "5be422d8a0ad0a055532b0a7",
5     "name": "Manee",
6     "weight": "55",
7     "__v": 0
8   }
9 ]

```

Figure 15: Using GET on <http://localhost:3000/persons> to see the current list of person

GET localhost:3000/persons

localhost:3000/persons

GET localhost:3000/persons Send Save

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

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 14 ms Size: 396 B Download

Pretty Raw Preview JSON

```

1 [
2   {
3     "country": "Thailand",
4     "_id": "5be422d8a0ad0a055532b0a7",
5     "name": "Manee",
6     "weight": "55",
7     "__v": 0
8   },
9   {
10    "country": "Japan",
11    "_id": "5be42304a0ad0a055532b0a8",
12    "name": "Mana",
13    "weight": "70",
14    "__v": 0
15  }
16 ]

```

Figure 15: Using GET on <http://localhost:3000/persons> after adding two persons

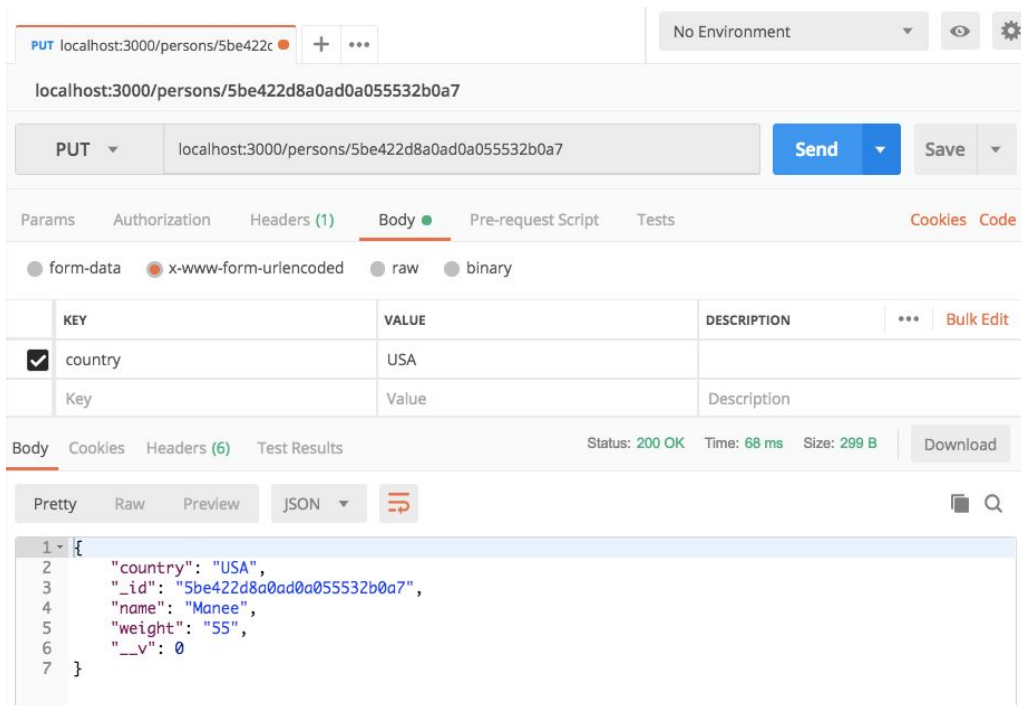


Figure 16: Using PUT on <http://localhost:3000/persons/<personID>> to update the country of a person

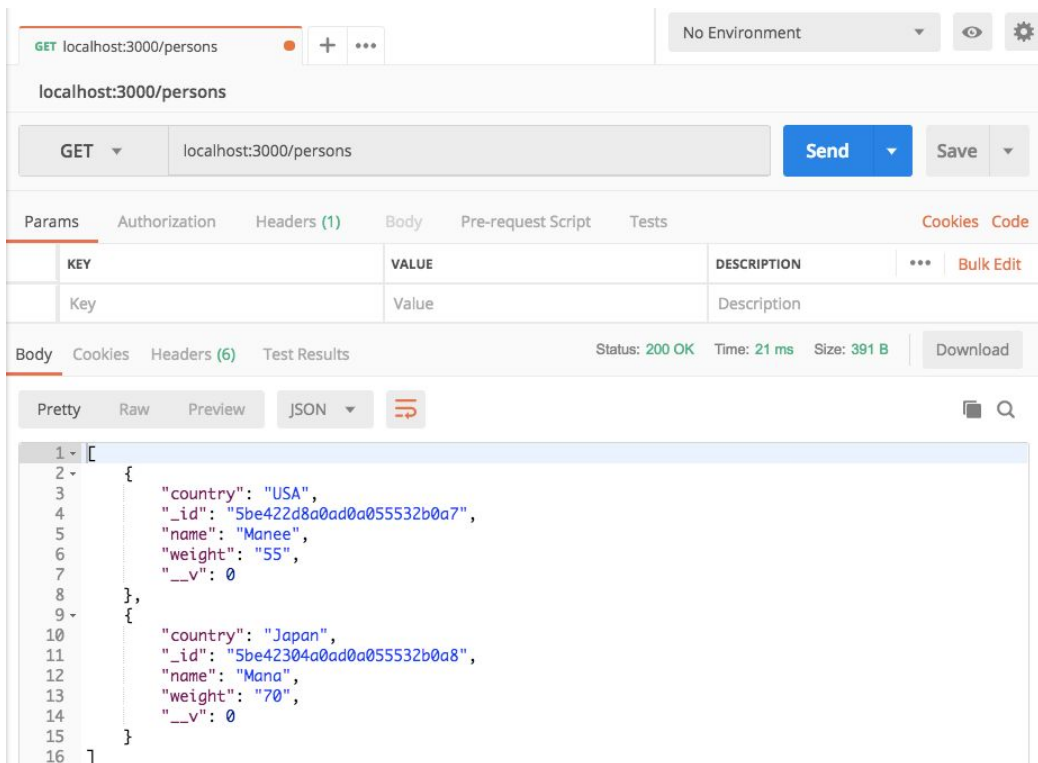


Figure 17: Using GET on <http://localhost:300/persons> to see the current list

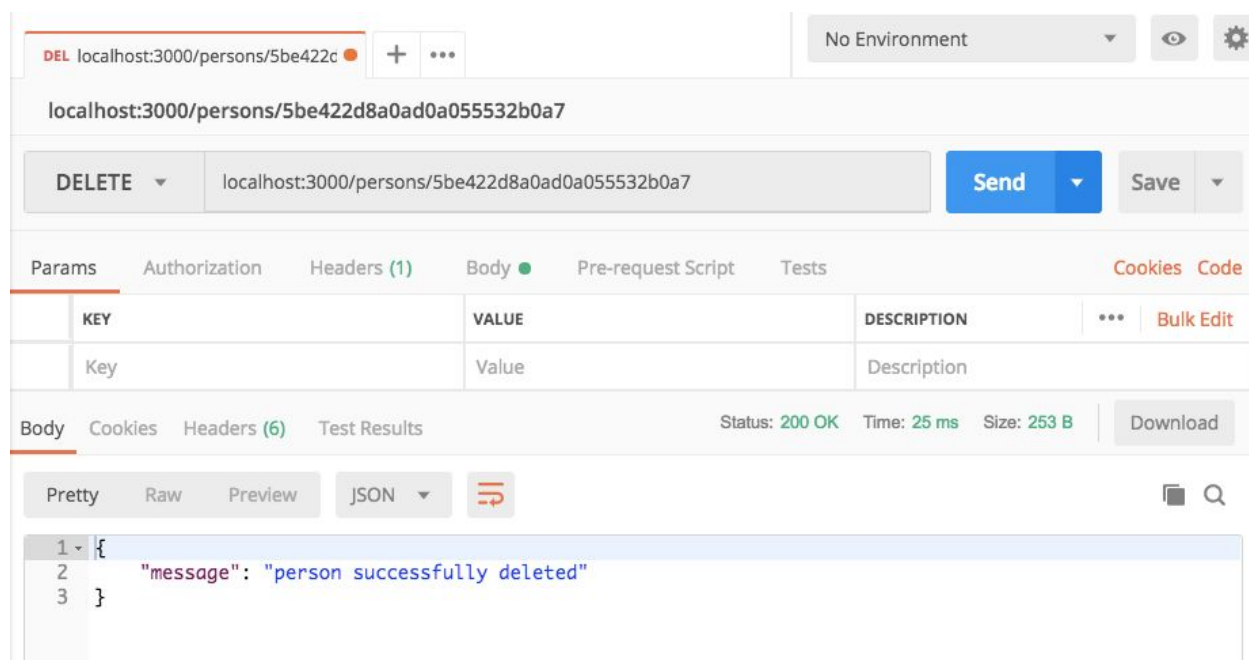


Figure 18: Using DELETE on <http://localhost:3000/persons/<personID>> to remove the person

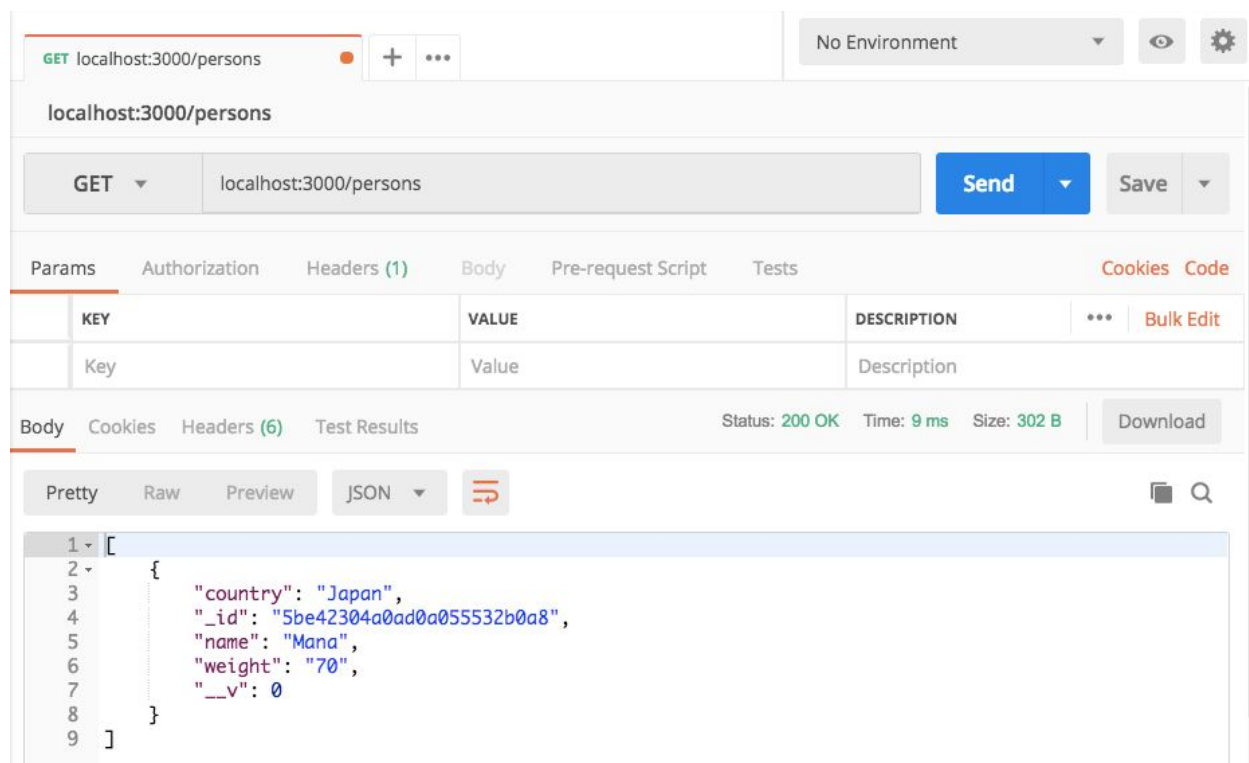


Figure 19: Using GET on <http://localhost:3000/persons> to see the current list of persons