

# IOCO Assessment

## Invoice-API

### Testing Document

This is just a document, illustrating simple steps to test the functionality of my Invoice-API, A Spring Boot REST API application. We will be using Postman to achieve this

#### Step 1

##### Create an Invoice:

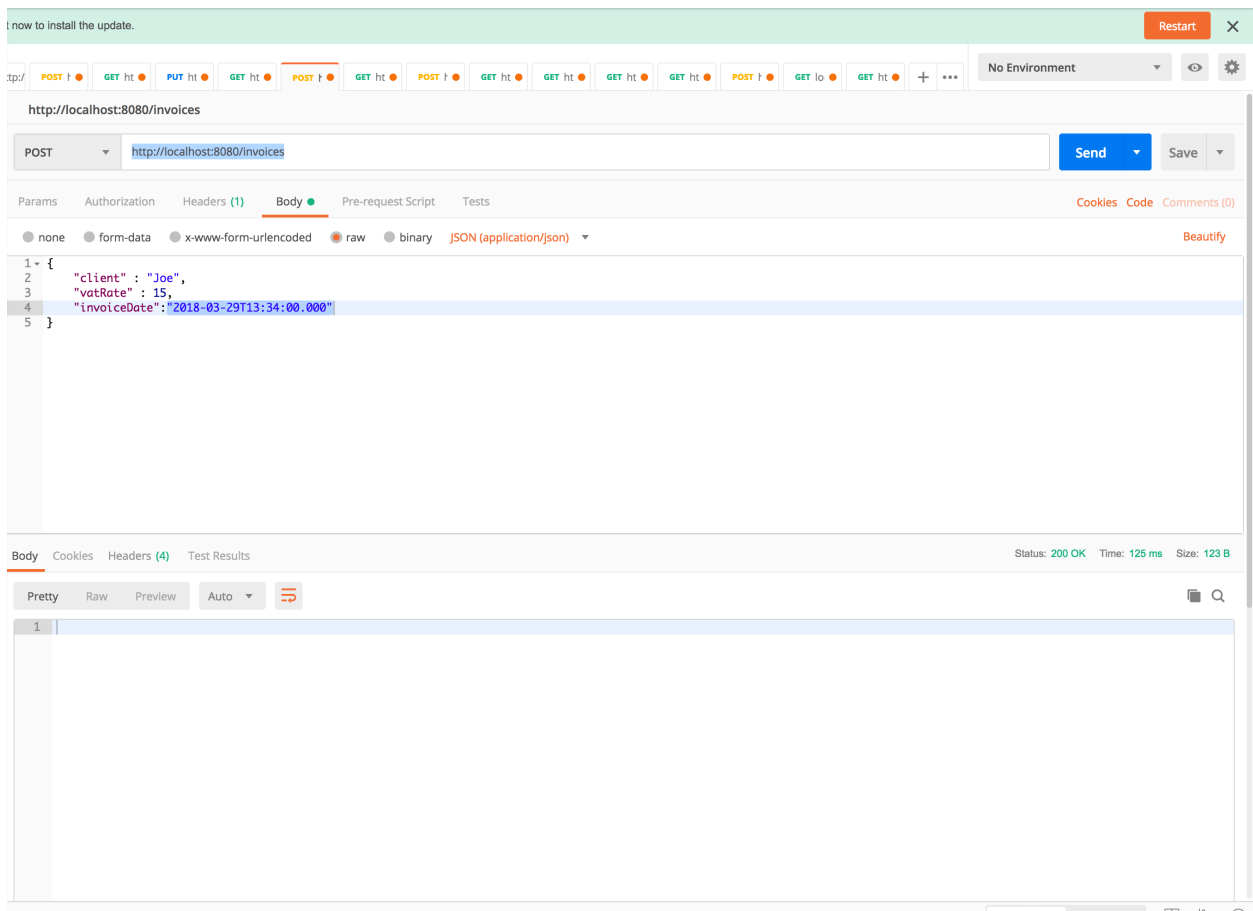
Using our REST End Point:

Method: POST

URL: <http://localhost:8080/invoices>

Send the following JSON on the Request Body,  
in the Request Header set Content-Type = application/json  
as displayed in screen shot below

```
{  
  "client" : "Joe"  
  "vatRate": 15  
  "invoiceDate": "2018-03-29T13:34:00.000"  
}
```



## Step 2

### Create a 2nd Invoice (Same as Step1)

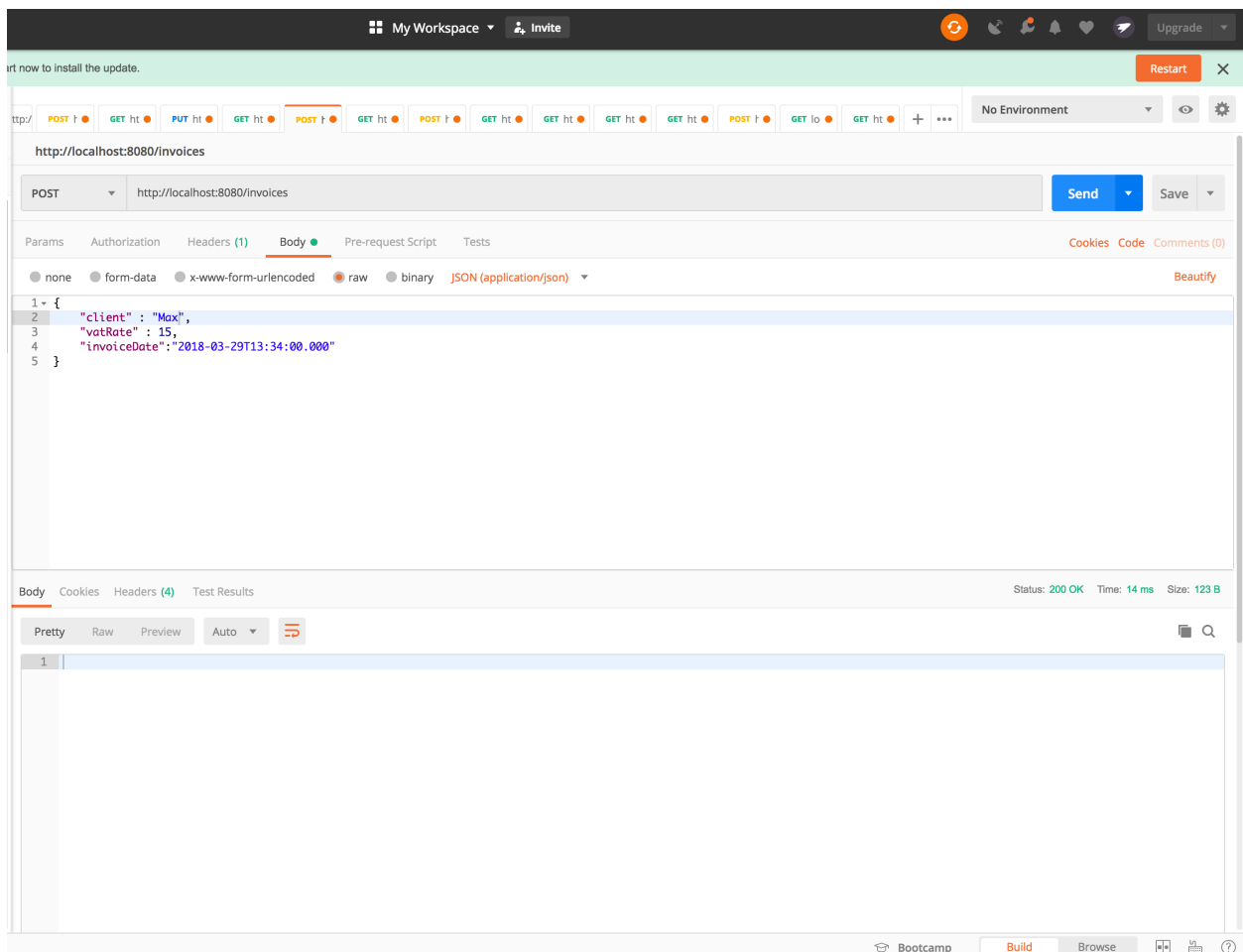
Using our REST End Point:

Method: POST

URL: <http://localhost:8080/invoices>

Send the following JSON on the Request Body,  
in the Request Header set Content-Type = application/json  
as displayed in screen shot below

```
{
  "client" : "Max"
  "vatRate": 15
  "invoiceDate": "2018-03-29T13:34:00.000"
}
```



### Step 3

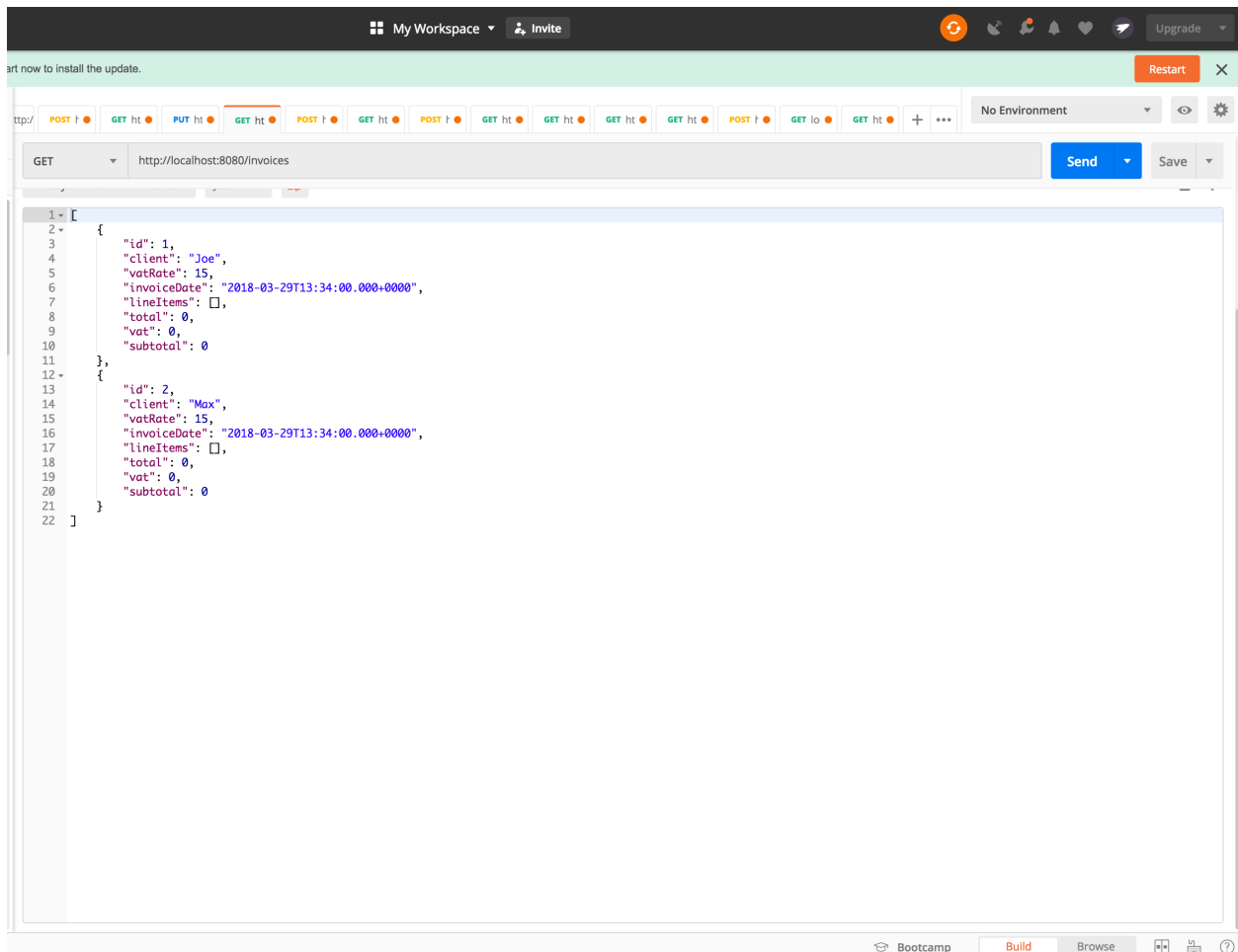
#### Check if our Invoices are created

Using our REST End Point:

Method: GET

URL: <http://localhost:8080/invoices>

As done in the screen shot below:



## Step 4

### Add a Line Item to Invoice - id= 1 Client = Joe

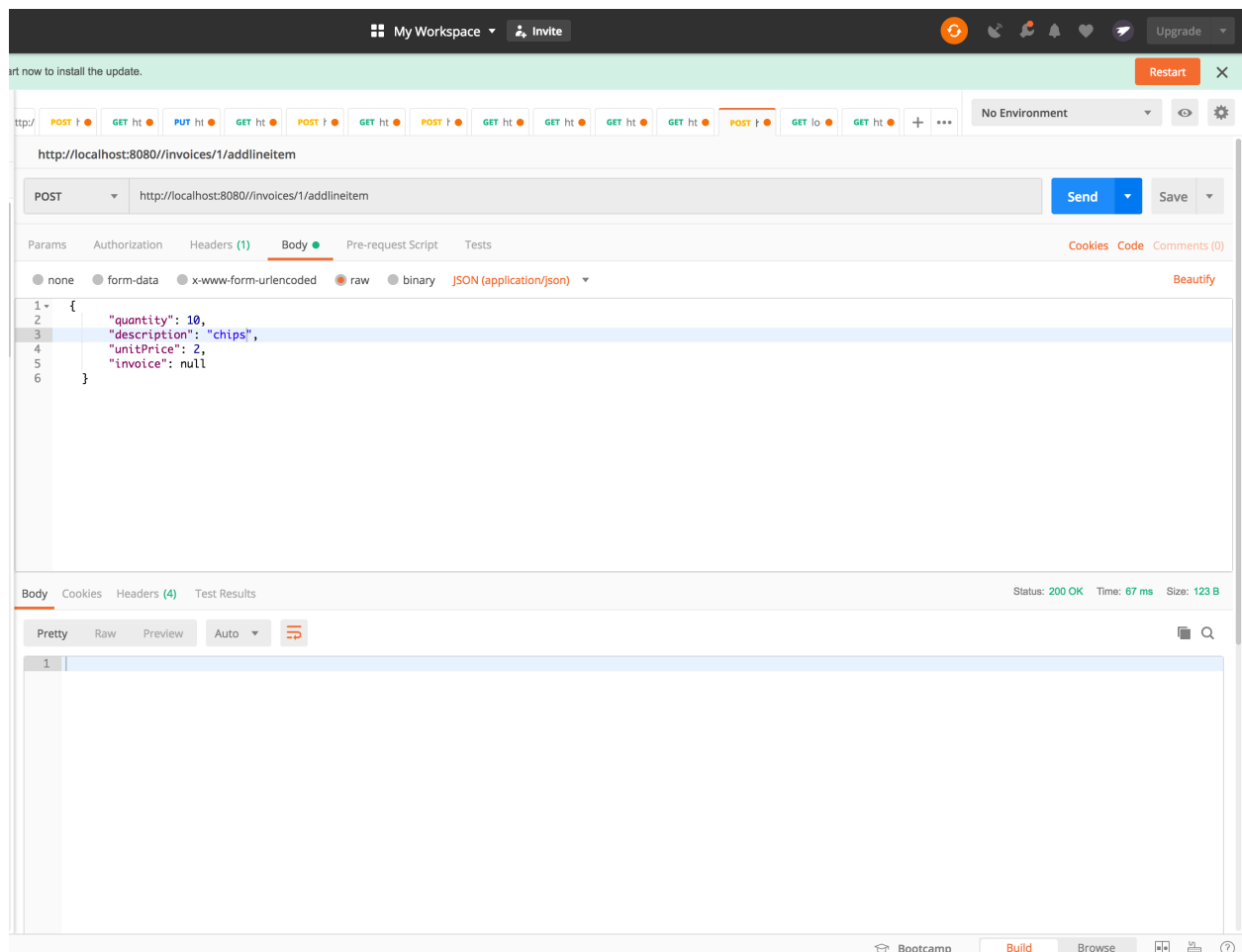
Using our REST End Point:

Method: POST

URL: <http://localhost:8080/invoices/{invoiceId}/addlineitem>

Send the following JSON on the Request Body,  
in the Request Header set Content-Type = application/json  
as displayed in screen shot below

```
{
  "quantity": 10,
  "description": "chips",
  "unitPrice": 2,
  "invoice": null
}
```



## Step 5

### Add another Line Item to Invoice - id= 1 Client = Joe (same as Step 4)

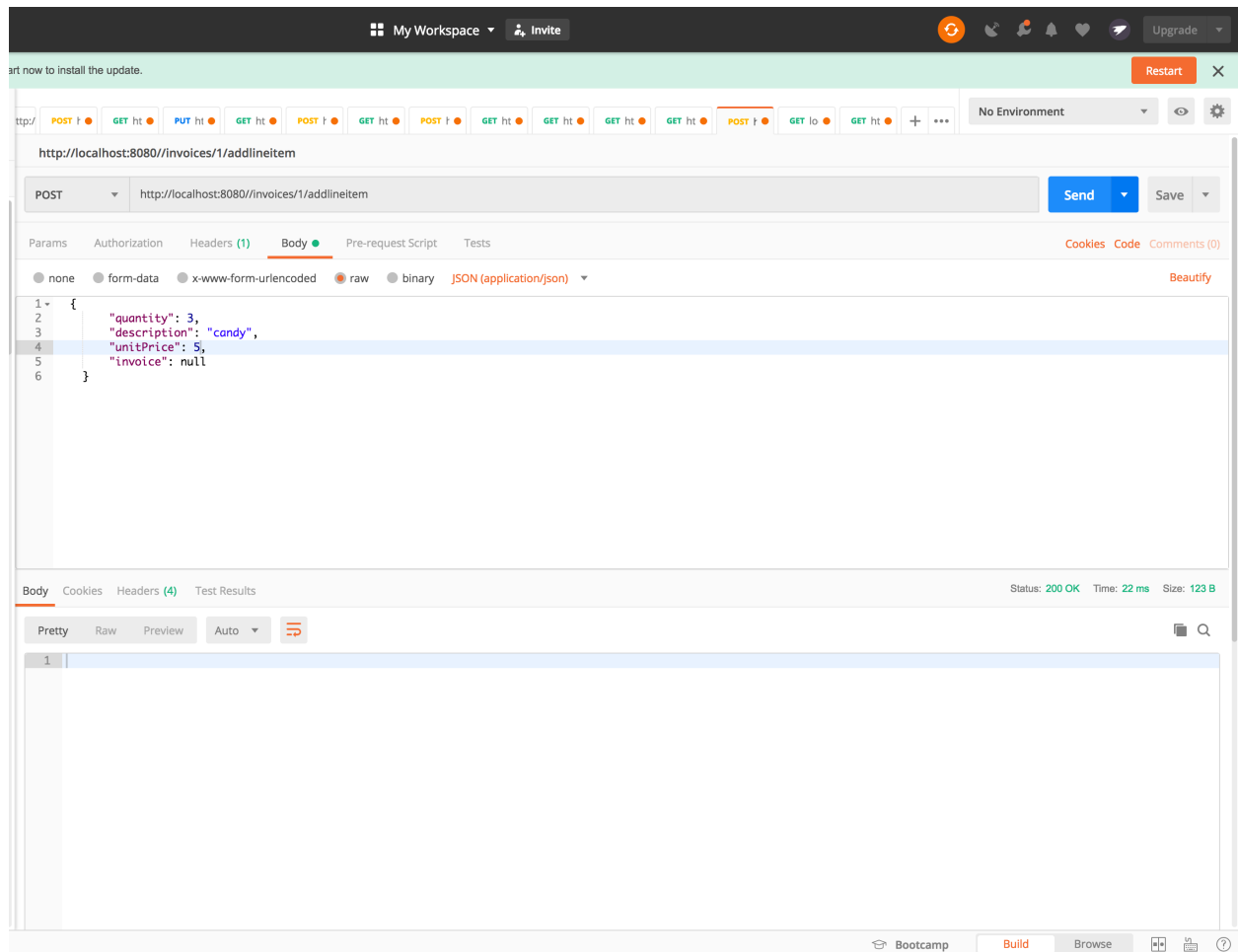
Using our REST End Point:

Method: POST

URL: <http://localhost:8080/invoices/{invoiceId}/addlineitem>

Send the following JSON on the Request Body,  
in the Request Header set Content-Type = application/json  
as displayed in screen shot below

```
{
  "quantity": 10,
  "description": "chips",
  "unitPrice": 2,
  "invoice": null
}
```



## Step 6

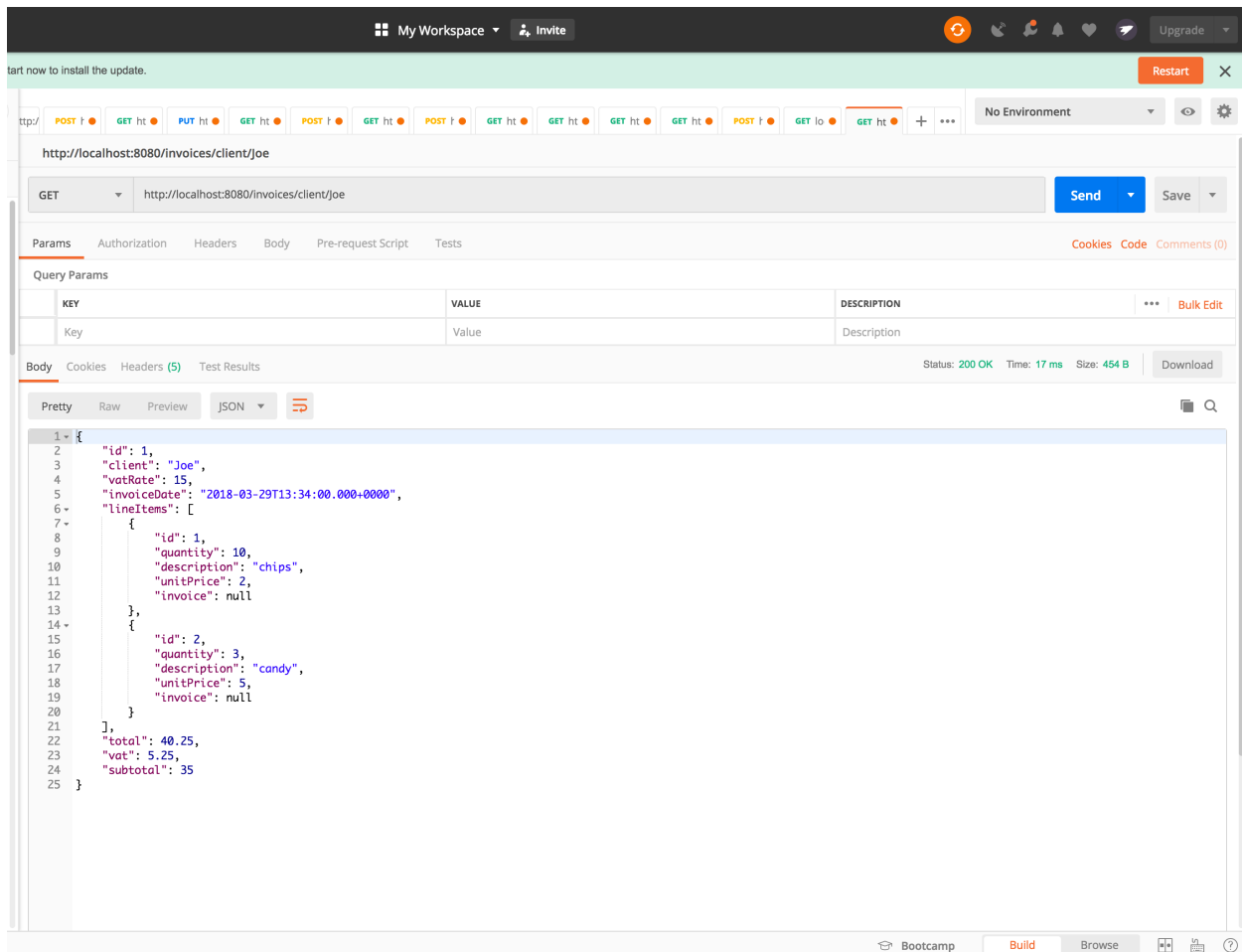
**View Invoice - id = 1 Client = Joe**  
**To see our LineItems added**

Using our REST End Point:

Method: GET

URL: <http://localhost:8080/invoices/client/{client}>

As done in the screen shot below:



### Note on Calculations:

#### Sub Total = 35

Line Item - id= 1, desc = Chips, quantity = 10 , unitPrice = 2

quantity \* unitPrice = 10 \* 2 = 20

Line Item - id= 2, desc = Candy, quantity = 3 , unitPrice = 5

quantity \* unitPrice = 3 \* 5 = 15

20 + 15 = 35

**Vat = Sub Total \* vat Rate = 35 \* 15% = 5.25**

**Total = Sub Total + Vat = 35 + 5.25 = 40.25**

## Step 7

### Add a Line Item to Invoice - id= 2 Client = Max

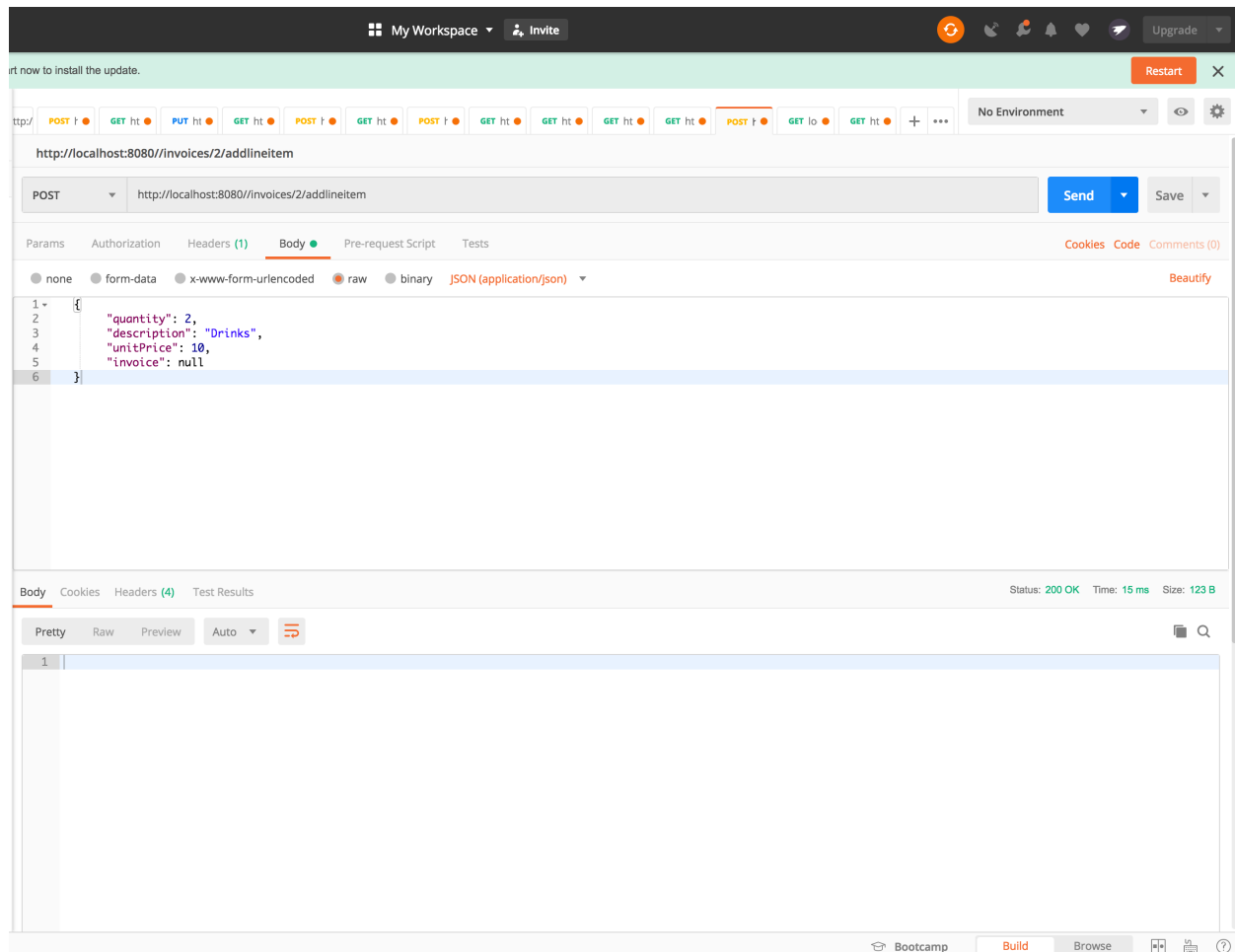
Using our REST End Point:

Method: POST

URL: <http://localhost:8080/invoices/{invoiceId}/addlineitem>

Send the following JSON on the Request Body,  
in the Request Header set Content-Type = application/json  
as displayed in screen shot below

```
{
  "quantity": 2,
  "description": "Drinks",
  "unitPrice": 10,
  "invoice": null
}
```



## Step 8

### Add another Line Item to Invoice - id= 2 Client = Max (same as Step 7)

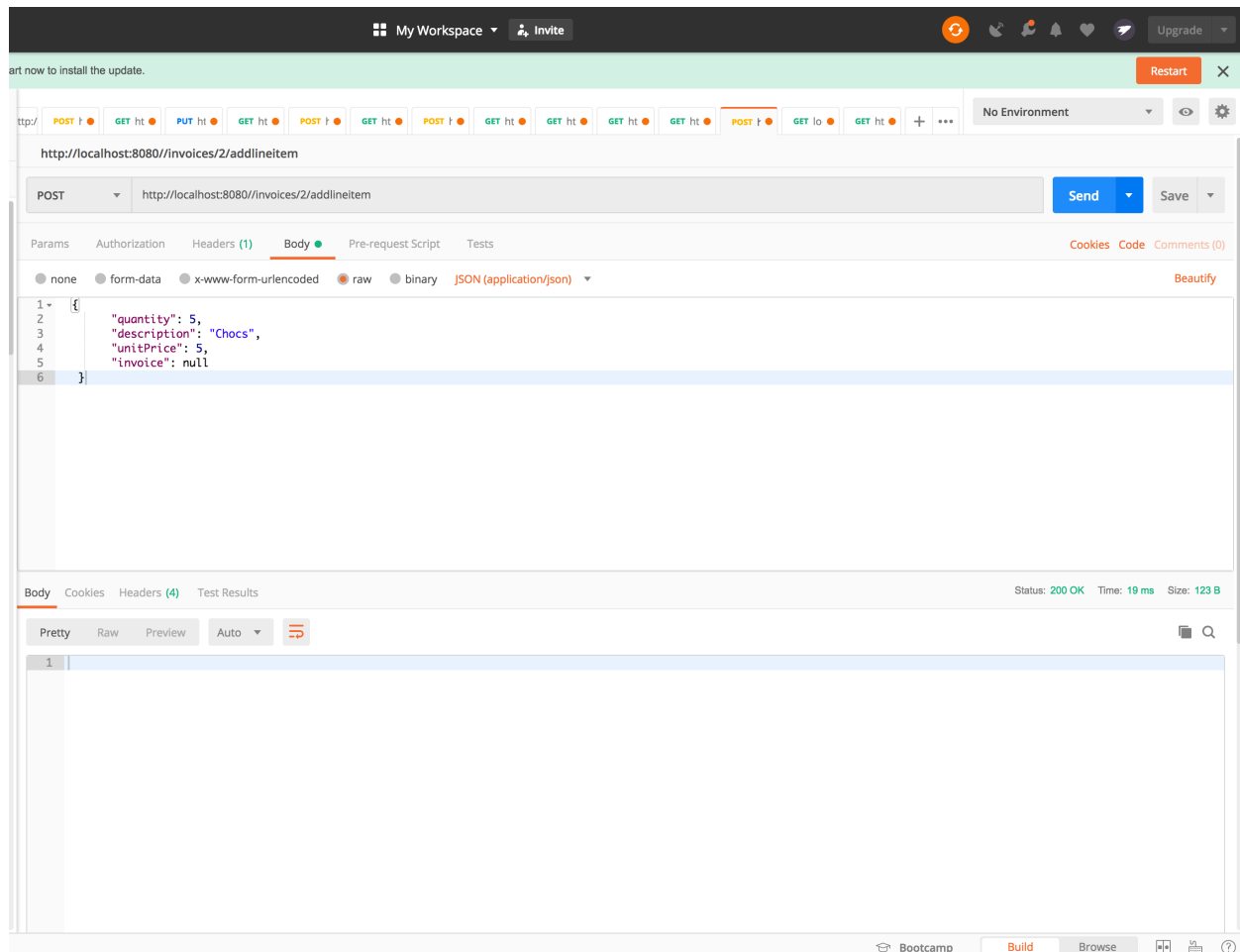
Using our REST End Point:

Method: POST

URL: <http://localhost:8080/invoices/{invoiceId}/addlineitem>

Send the following JSON on the Request Body,  
in the Request Header set Content-Type = application/json  
as displayed in screen shot below

```
{
  "quantity": 5,
  "description": "Chocs",
  "unitPrice": 5,
  "invoice": null
}
```





## Step 9

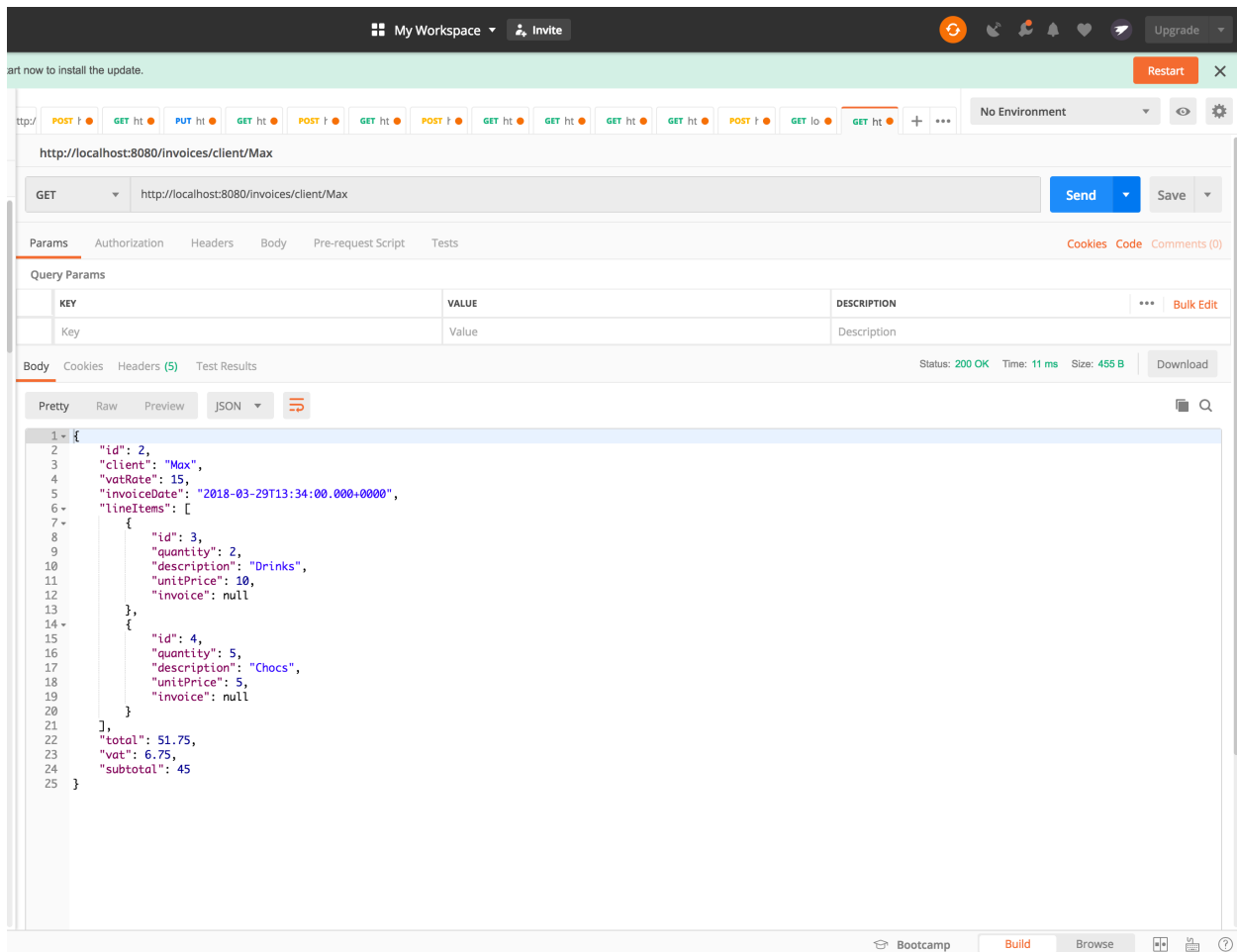
### View Invoice - id = 2 Client = Max To see our LineItems added

Using our REST End Point:

Method: GET

URL: <http://localhost:8080/invoices/client/{client}>

As done in the screen shot below:



### Note on Calculations:

#### Sub Total = 45

Line Item - id= 3, desc = Drinks, quantity = 2 , unitPrice = 10  
quantity \* unitPrice = 2 \* 10 = 20  
Line Item - id= 4, desc = Chocs, quantity = 5 , unitPrice = 5  
quantity \* unitPrice = 5 \* 5 = 25  
20 + 25 = 45

**Vat** = Sub Total \* vat Rate = 45 \* 15% = 6.75

**Total** = Sub Total + Vat = 45 + 6.75 = 51.75

## Step 10

### View all Invoices

Using our REST End Point:

Method: GET

URL: <http://localhost:8080/invoices>

Note All LinelItems are added to their invoice and the calculations for Sub Total, Vat Total and Total are displayed for each Invoice

As done in the screen shot below:

The screenshot shows the Postman application interface. At the top, there's a notification bar for Postman updates. Below it, the URL bar shows 'http://localhost:8080/invoices' with the method 'GET'. The 'Params' tab is active, showing an empty table for query parameters. The 'Body' tab is also active, showing a JSON response in 'Pretty' format. The response is a JSON array of two invoice objects. The first invoice is for 'Joe' with a total of 40.25, and the second is for 'Max' with a total of 51.75. Both invoices have a list of line items with quantities, descriptions, and unit prices.

```
1- [
2-   {
3-     "id": 1,
4-     "client": "Joe",
5-     "vatRate": 15,
6-     "invoiceDate": "2018-03-29T13:34:00.000-0000",
7-     "lineItems": [
8-       {
9-         "id": 1,
10-        "quantity": 10,
11-        "description": "chips",
12-        "unitPrice": 2,
13-        "invoice": null
14-      },
15-      {
16-        "id": 2,
17-        "quantity": 3,
18-        "description": "candy",
19-        "unitPrice": 5,
20-        "invoice": null
21-      }
22-    ],
23-    "total": 40.25,
24-    "vat": 5.25,
25-    "subtotal": 35
26-  },
27-   {
28-     "id": 2,
29-     "client": "Max",
30-     "vatRate": 15,
31-     "invoiceDate": "2018-03-29T13:34:00.000-0000",
32-     "lineItems": [
33-       {
34-         "id": 3,
35-         "quantity": 2,
36-         "description": "Drinks",
37-         "unitPrice": 10,
38-         "invoice": null
39-       },
40-       {
41-         "id": 4,
42-         "quantity": 5,
43-         "description": "Choco",
44-         "unitPrice": 5,
45-         "invoice": null
46-       }
47-     ],
48-     "total": 51.75,
49-     "vat": 6.75,
50-     "subtotal": 45
51-   }
52- ]
```