


# Interacting with Invoices: A Step-by-Step Process

22 Steps    [View most recent version on Tango.us](#) 

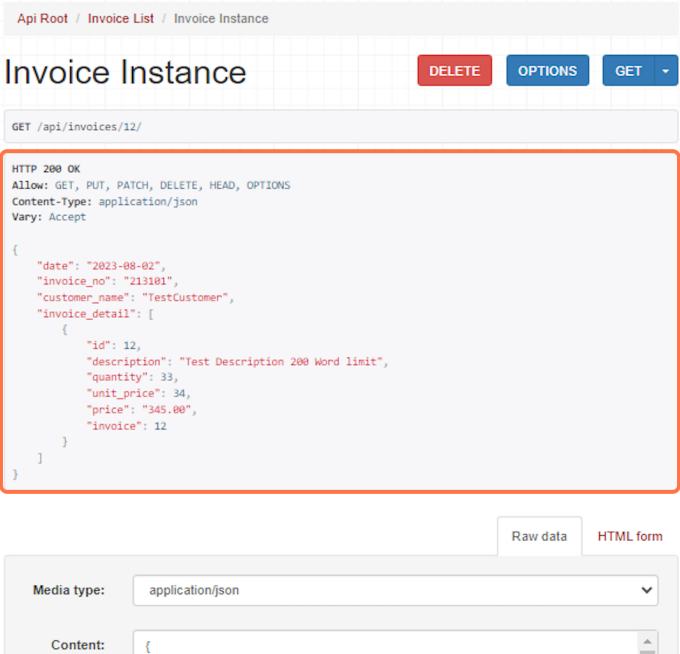
---

Created by	Creation Date	Last Updated
Rahul M.	August 2, 2023	August 2, 2023

## STEP 1

<http://127.0.0.1:8000/api/invoices/>

For Access all invoices or invoice list you can use this url



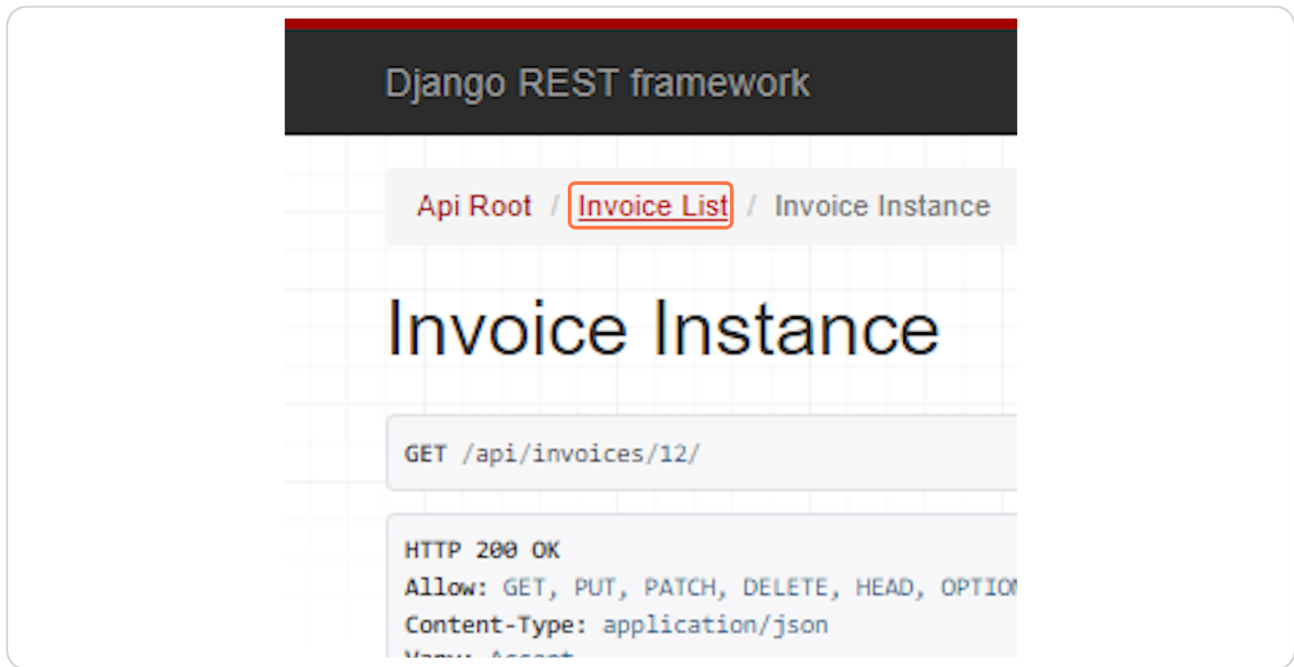
The screenshot displays a REST client interface for the endpoint `GET /api/invoices/12/`. The response is an `HTTP 200 OK` with `Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS`, `Content-Type: application/json`, and `Vary: Accept`. The JSON body is highlighted with an orange border and contains the following data:

```
{
  "date": "2023-08-02",
  "invoice_no": "213101",
  "customer_name": "TestCustomer",
  "invoice_detail": [
    {
      "id": 12,
      "description": "Test Description 200 Word limit",
      "quantity": 33,
      "unit_price": 34,
      "price": "345.00",
      "invoice": 12
    }
  ]
}
```

Below the response, there are tabs for `Raw data` and `HTML form`. The `Media type` is set to `application/json` and the `Content` field shows the start of the JSON object: `{`.

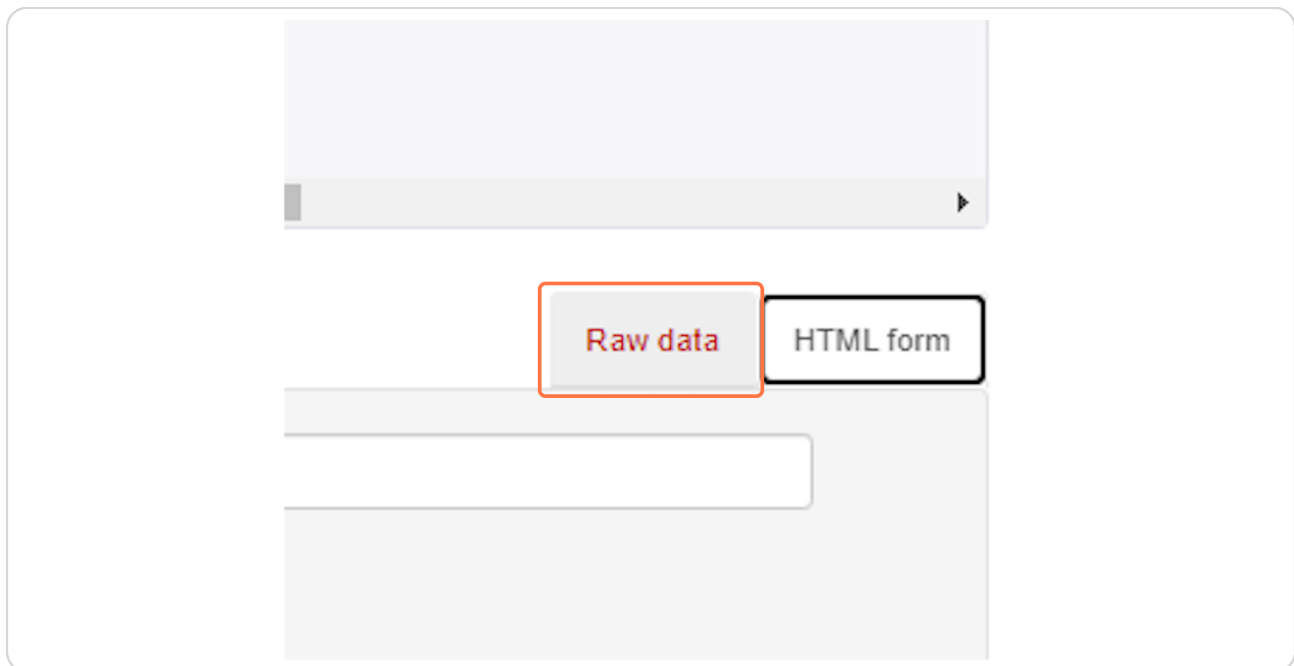
## STEP 2

Click on Invoice List



## STEP 3

Click on Raw data



## STEP 4

Click on application/json from Media type:

The screenshot shows a REST client interface. At the top, a text area contains a JSON object: 

```
{  "description": "Test Description 200 Word limit",  "quantity": 33,  "unit_price": 34,  "price": "345.00",  "invoice": 12}
```

. Below this, there are two tabs: "Raw data" and "HTML form". Under the "Raw data" tab, there is a dropdown menu labeled "type:" with "application/json" selected. Below the dropdown, there is a text area labeled "Content:" containing a JSON object: 

```
{  "customer_name": "",  "invoice_detail": []}
```

. A red rectangle highlights the "type:" dropdown and the "Content:" text area.

## STEP 5

Click on Content:

The screenshot shows a REST client interface. At the top, a text area contains a JSON object: 

```
{  "description": "Test Description 200 Word limit",  "quantity": 33,  "unit_price": 34,  "price": "345.00",  "invoice": 12}
```

. Below this, there are two tabs: "Raw data" and "HTML form". Under the "Raw data" tab, there is a dropdown menu labeled "type:" with "application/json" selected. Below the dropdown, there is a text area labeled "Content:" containing a JSON object: 

```
{  "customer_name": "",  "invoice_detail": []}
```

. A red rectangle highlights the "Content:" text area.

## STEP 6

Click on Content:

The screenshot shows a REST client interface. At the top, a code editor displays a JSON object: 

```
{  "description": "Test Description 200 Word limit",  "quantity": 33,  "unit_price": 34,  "price": "345.00",  "invoice": 12}
```

. Below the code editor, there are two tabs: "Raw data" (selected) and "HTML form". Under the "Raw data" tab, there is a dropdown menu labeled "type:" with "application/json" selected. Below the dropdown, there is a text area labeled "Content:" which is currently empty and highlighted with a red border. A blue button is visible at the bottom right of the interface.

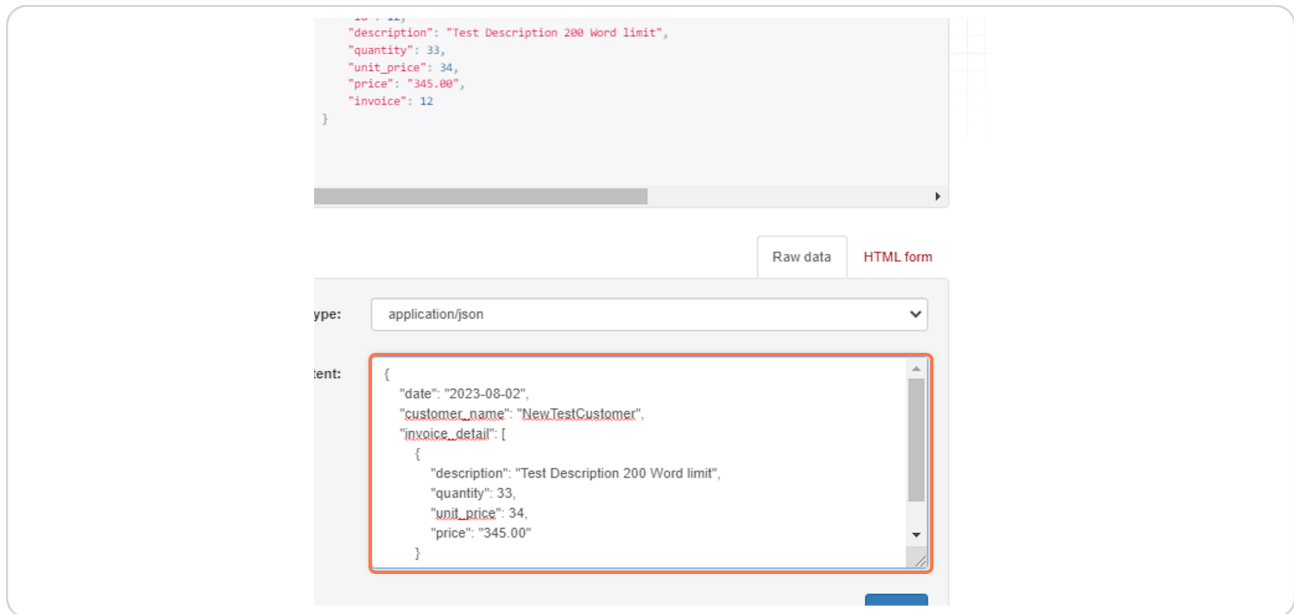
## STEP 7

Click on this content textarea and type objects date into json format

This screenshot is identical to the one in Step 6. It shows the same REST client interface with the JSON object in the code editor, the "Raw data" tab selected, and the "Content:" text area highlighted with a red border. The interface elements, including the dropdown menu and the blue button, are the same as in the previous step.

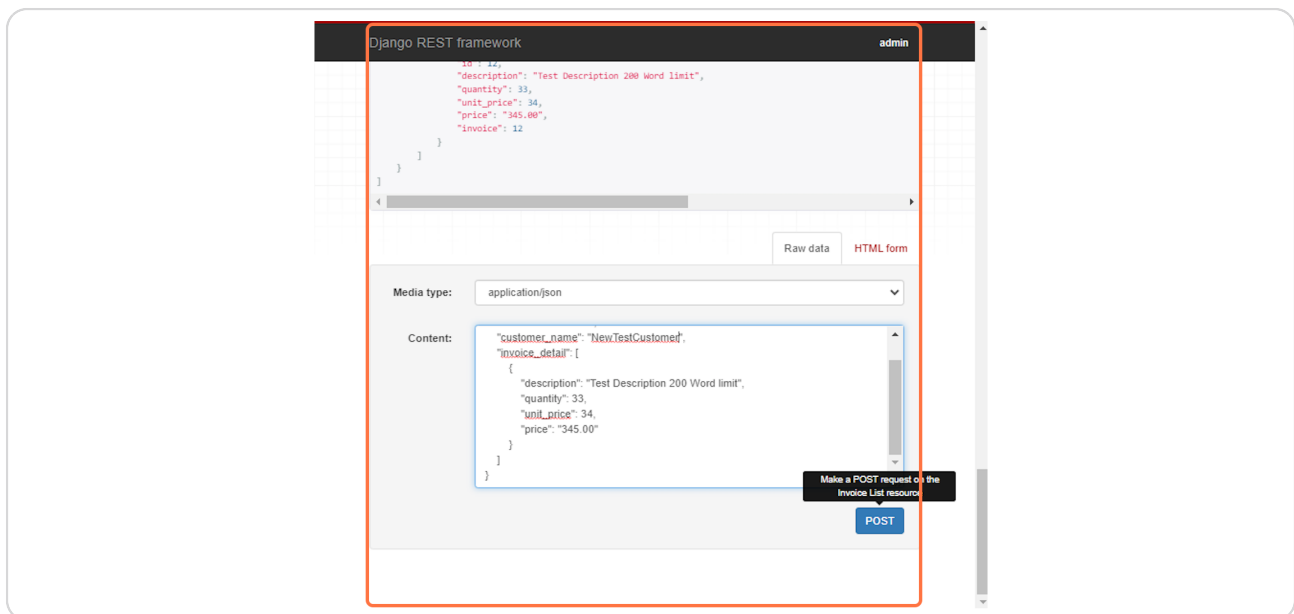
## STEP 8

Type "{ "date": "2023-08-02", "customer\_name": "NewTestCustomer", "invoice\_detail": [ { "description": "Test Description 200 Word limit", "quantity": 33, "unit\_price": 34, "price": "345.00" } ]}"



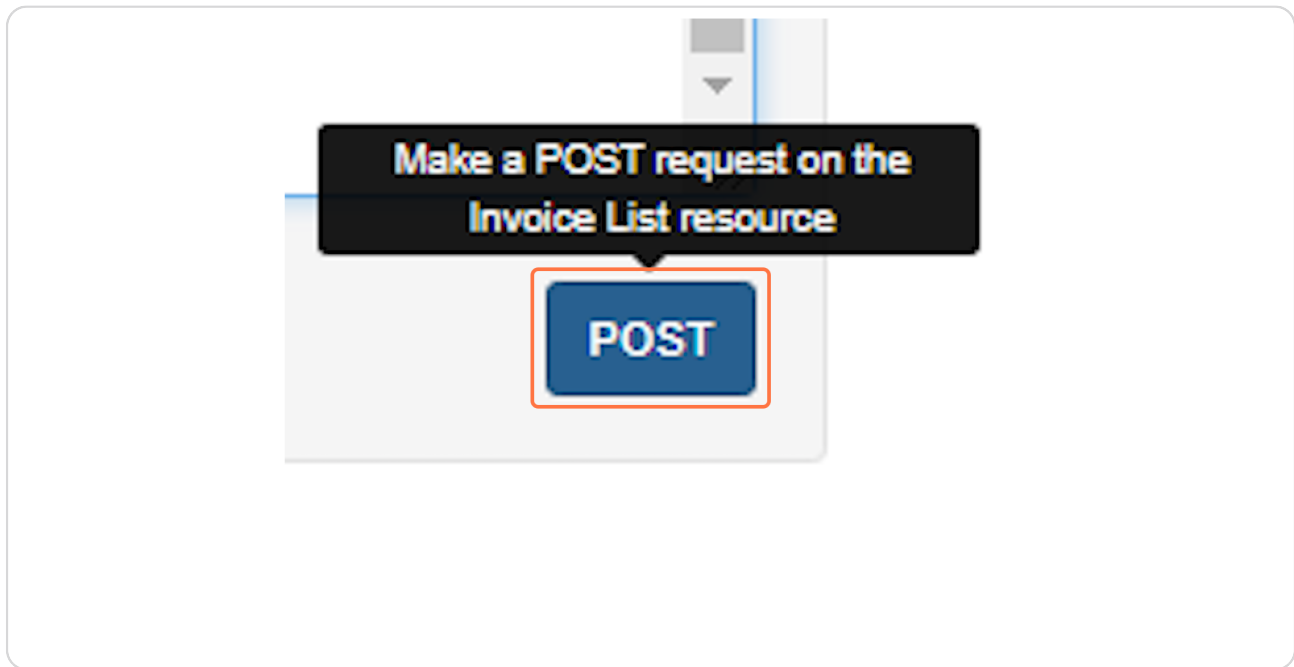
## STEP 9

Click on content



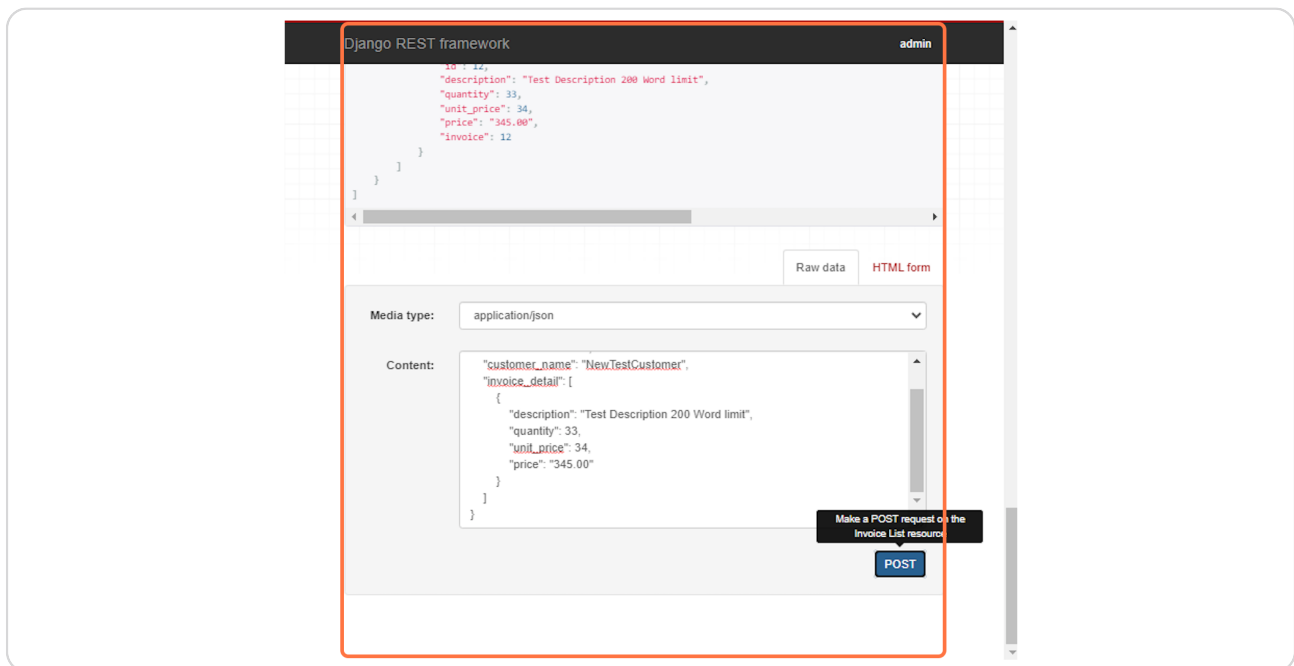
## STEP 10

Click on POST



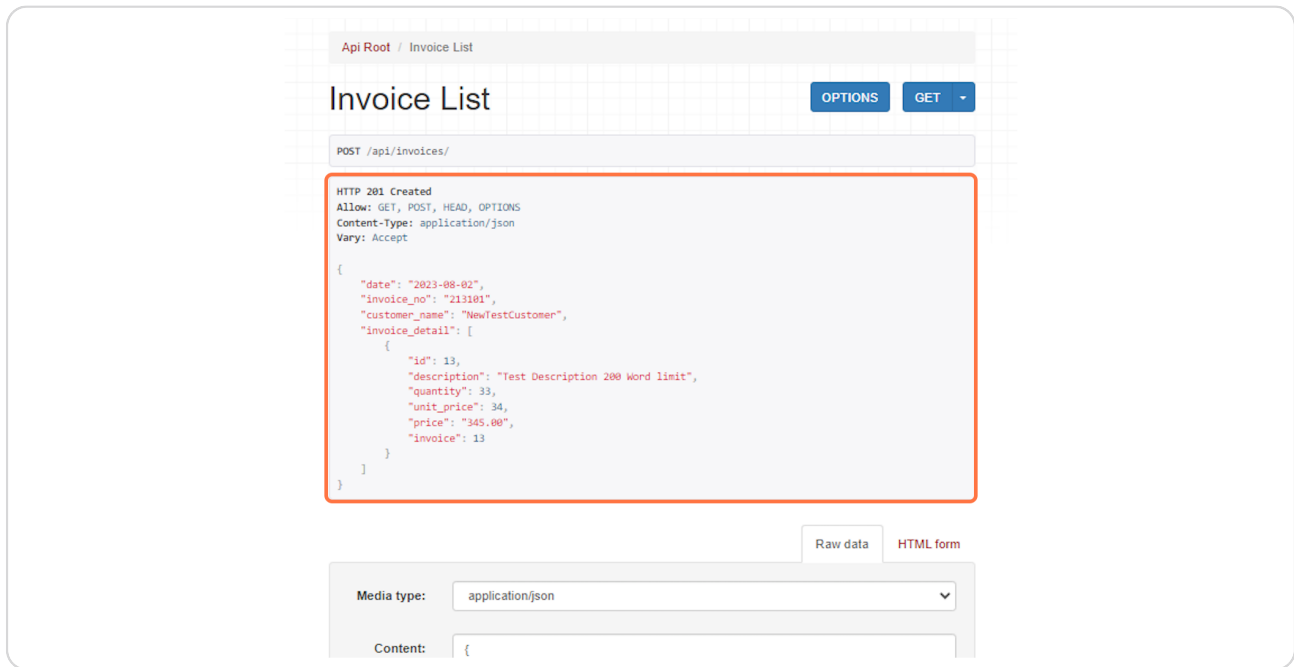
## STEP 11

Submit content



## STEP 12

### Now out New Invoice are create



The screenshot shows a REST client interface for the 'Api Root / Invoice List' endpoint. The 'POST /api/invoices/' method is selected. The response is highlighted with a red box and shows a successful 'HTTP 201 Created' status. The response headers include 'Allow: GET, POST, HEAD, OPTIONS', 'Content-Type: application/json', and 'Vary: Accept'. The response body is a JSON object representing a newly created invoice.

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "date": "2023-08-02",
  "invoice_no": "213101",
  "customer_name": "NewTestCustomer",
  "invoice_detail": [
    {
      "id": 13,
      "description": "Test Description 200 Word limit",
      "quantity": 33,
      "unit_price": 34,
      "price": "345.00",
      "invoice": 13
    }
  ]
}
```

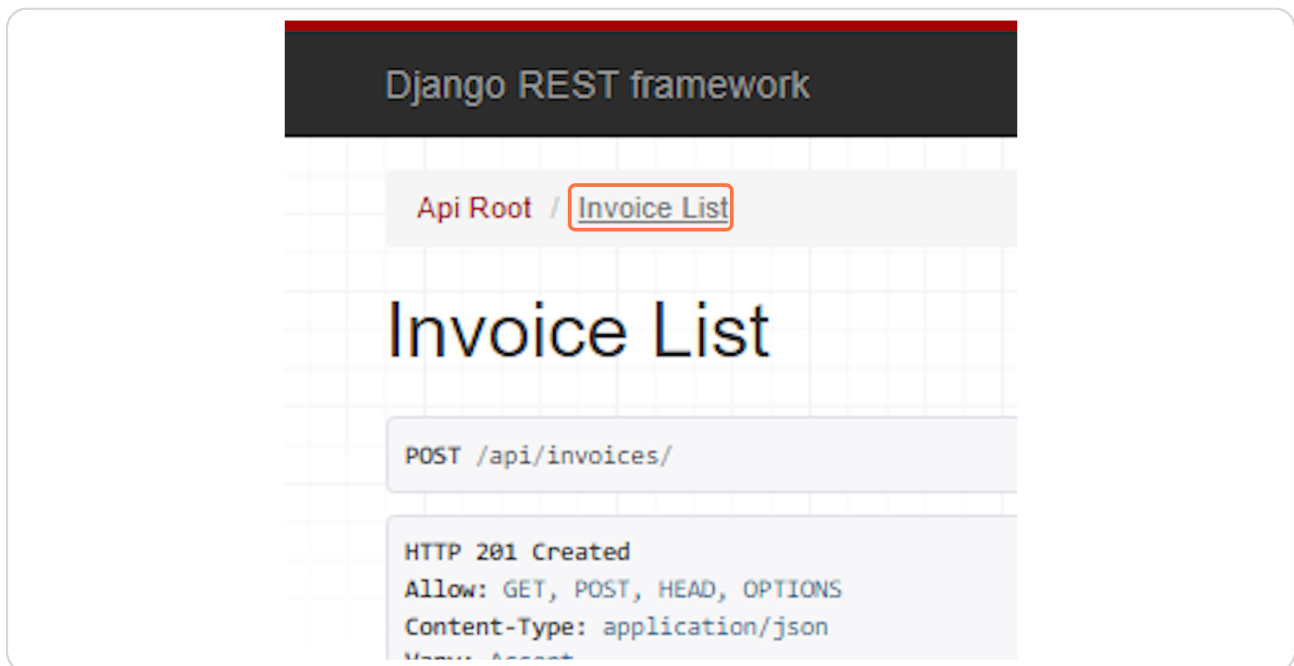
Raw data HTML form

Media type: application/json

Content: {

## STEP 13

### Click on Invoice List



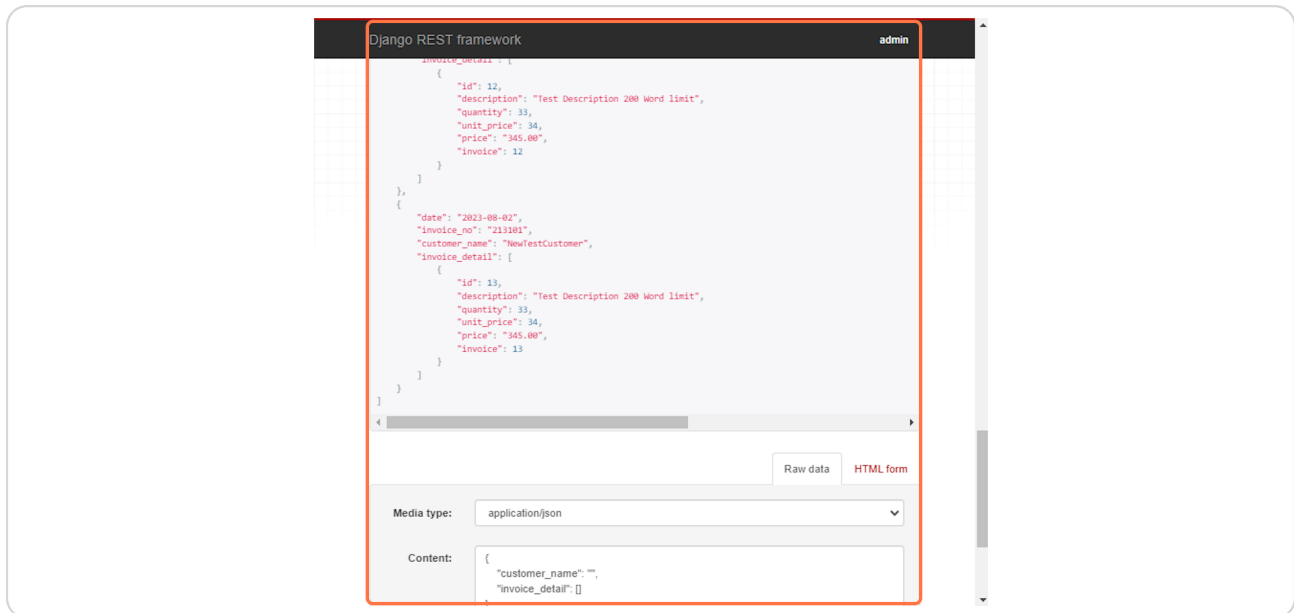
The screenshot shows a REST client interface for the 'Django REST framework' API. The 'Api Root / Invoice List' endpoint is selected. The 'POST /api/invoices/' method is selected. The response is shown as 'HTTP 201 Created' with headers 'Allow: GET, POST, HEAD, OPTIONS' and 'Content-Type: application/json'.

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```



## STEP 14

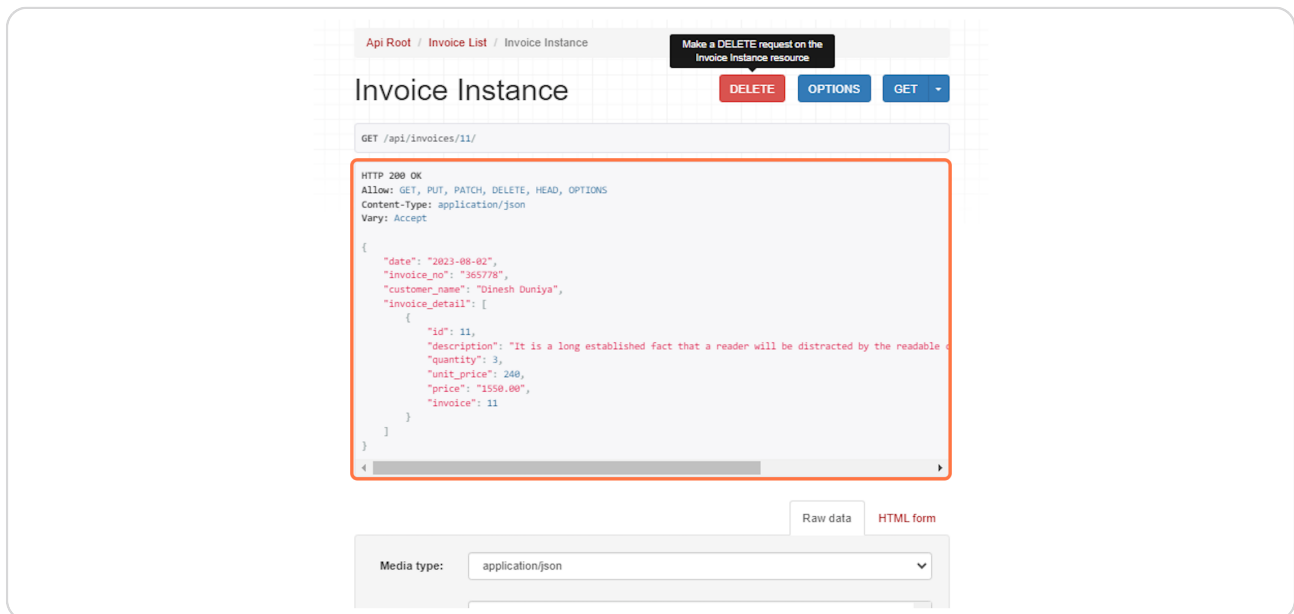
### Click on response info



## STEP 15

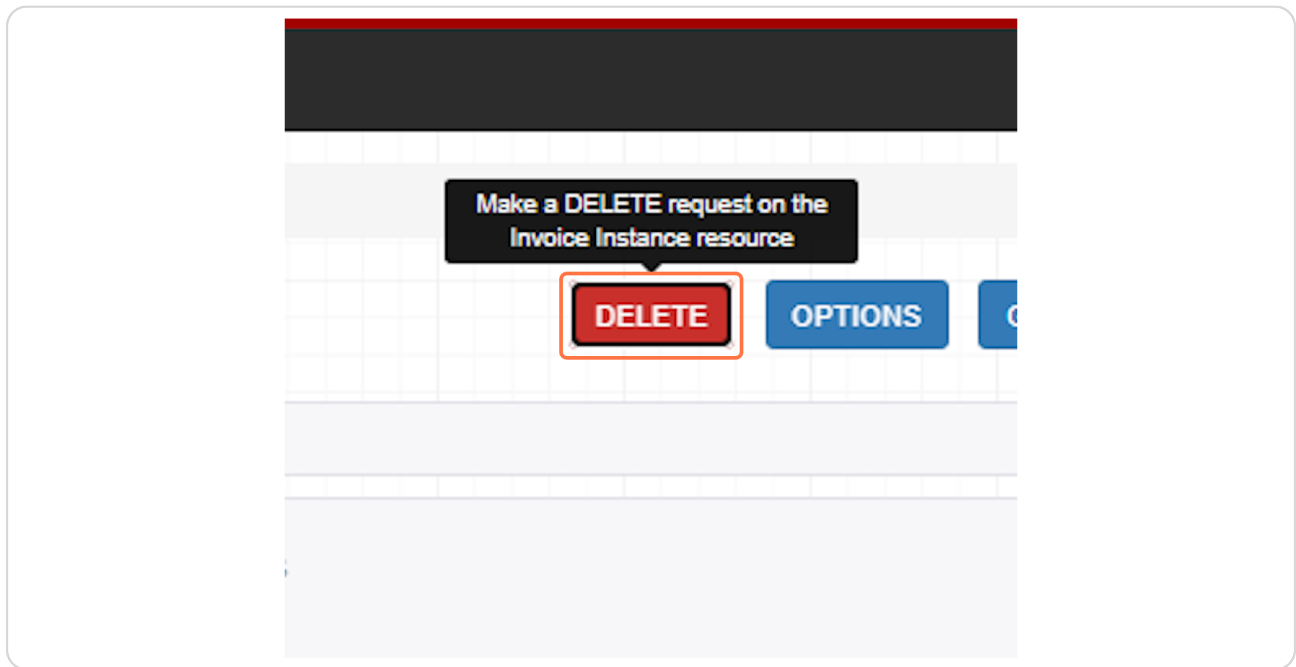
### <http://127.0.0.1:8000/api/invoices/11/>

Here We learn how can we delete invoice specific for this we have to use this url that i was using on 11 you can use the invoice id which one you want to delete for access that particular invoice



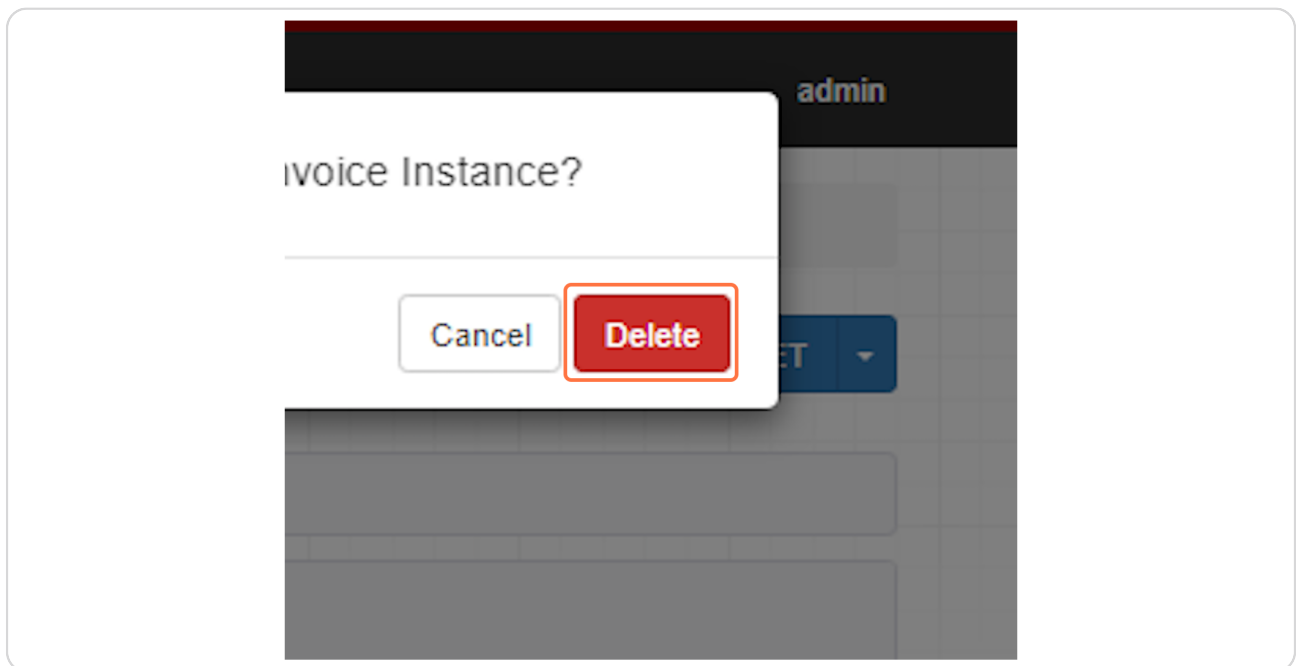
STEP 16

Click on DELETE



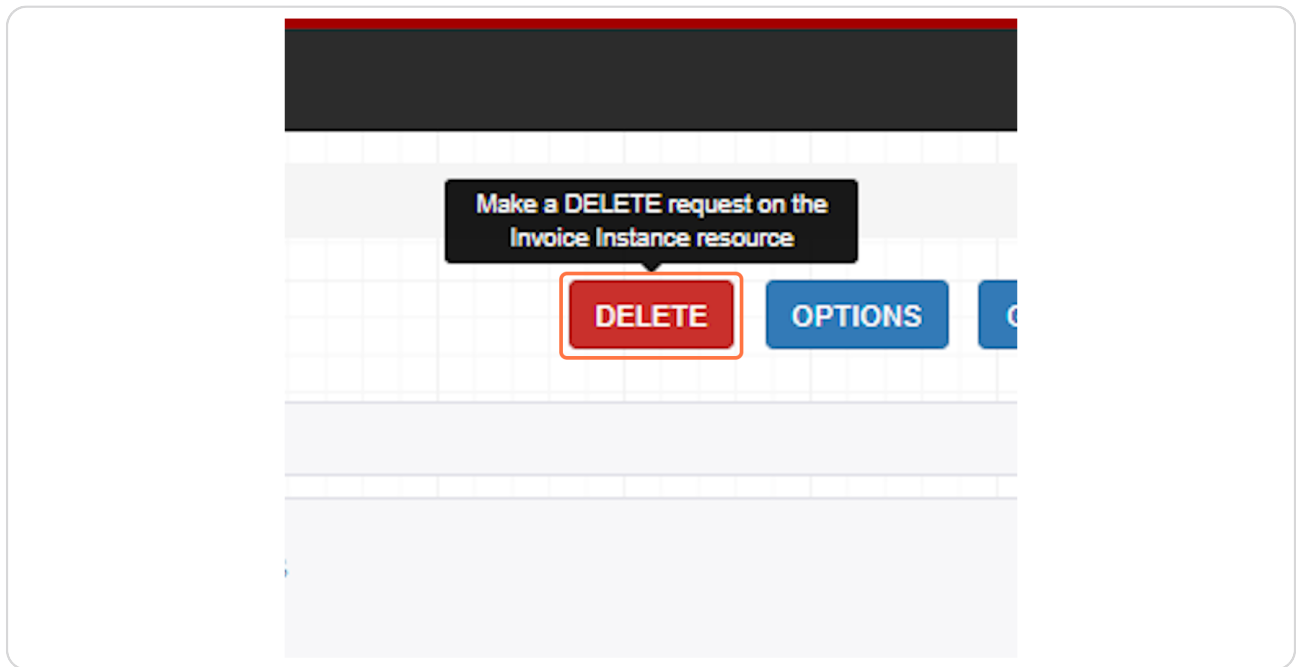
STEP 17

Click on Delete



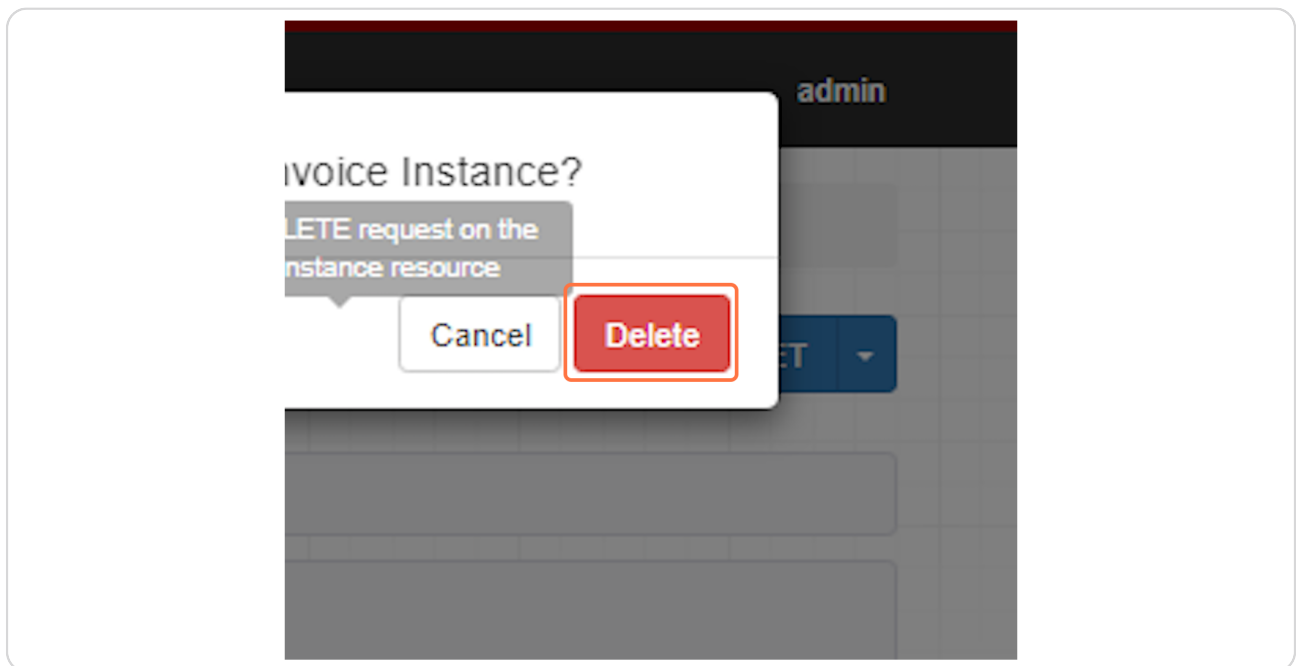
STEP 18

Click on DELETE



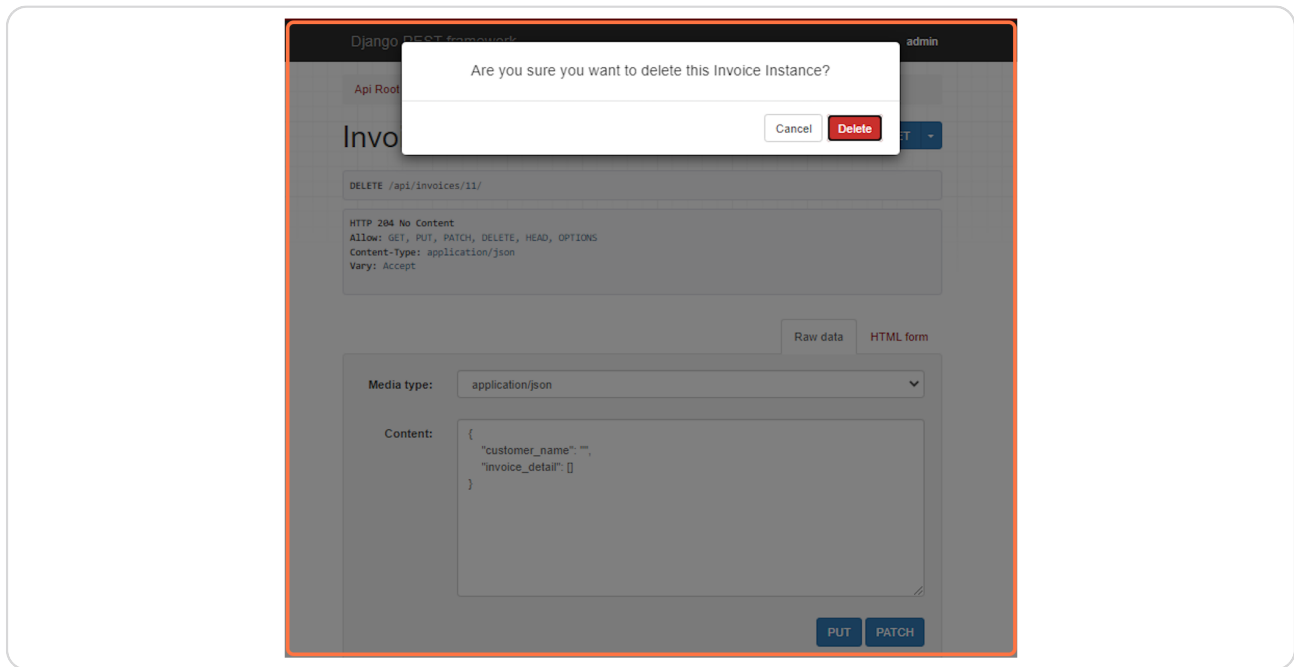
STEP 19

Click on Delete



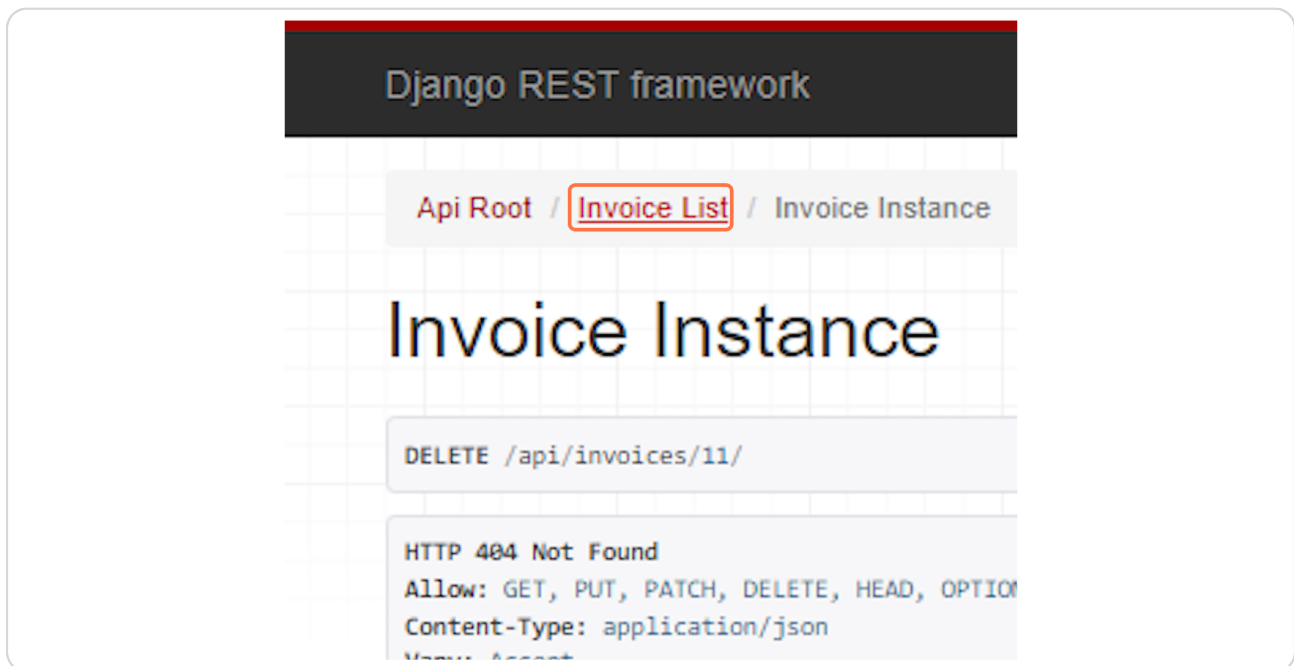
## STEP 20

### Submit dialog



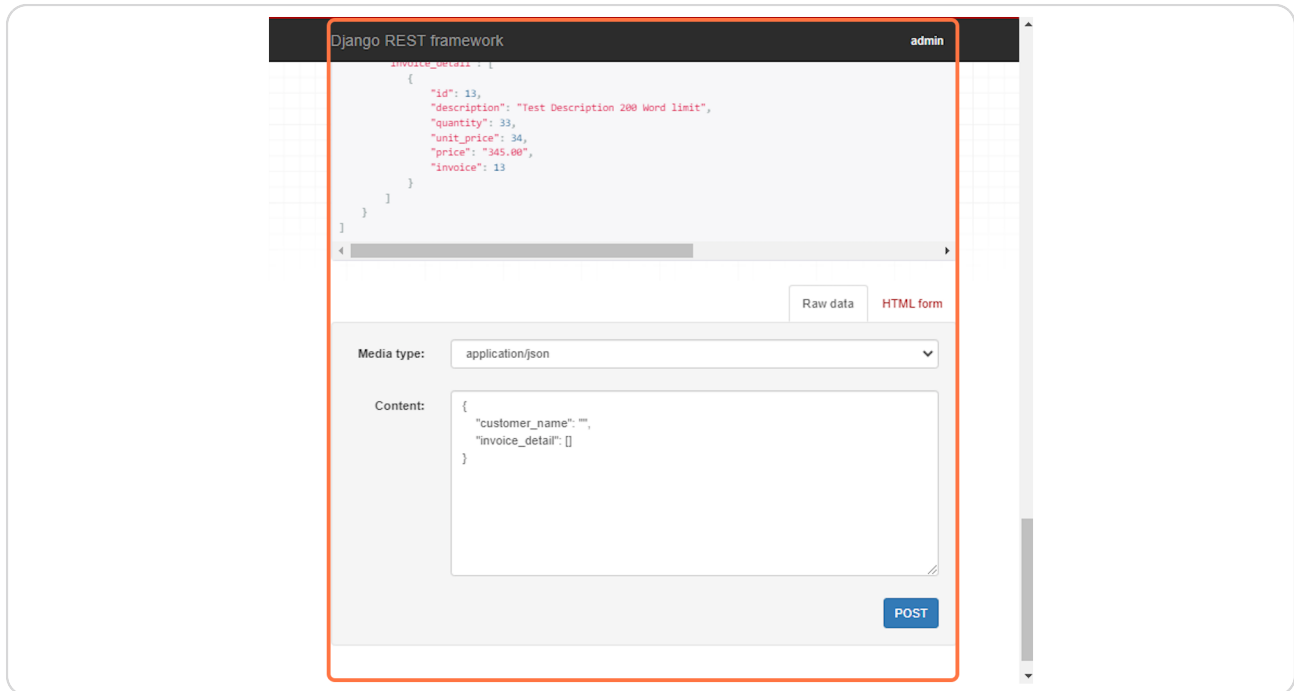
## STEP 21

### Click on Invoice List



## STEP 22

### Click on response info



*Tango*

Never miss a step again. Visit [Tango.us](https://tango.us)