

**In this assignment, we would learn about different types of HTTP requests and Responses in Django. Django uses request and response objects to pass state through the system.**

**When a client requests a resource, an HttpRequest object is created, and correspond view function is called that returns the HttpResponse object.**

**To handle request and response, Django provides HttpRequest and**

**HttpResponse/JsonResponse, classes**

**Some More details can be found here <https://docs.djangoproject.com/en/3.1/ref/request-response/>**

**This assignment will cover 4 HTTP request types i.e GET, POST, PUT, DELETE, and 2 response types HttpResponse and JsonResponse.**

**At a High level, the Django application will perform different HTTP requests to perform some logical operations.**

**Problem Statement Create a Django Application with the Name "simpleapp". Following are some use cases**

**Calculate the square of a given Number**

**Check whether a string is a palindrome**

**Add all the numbers in an array**

**Template 1: Calculate the square of a Number**

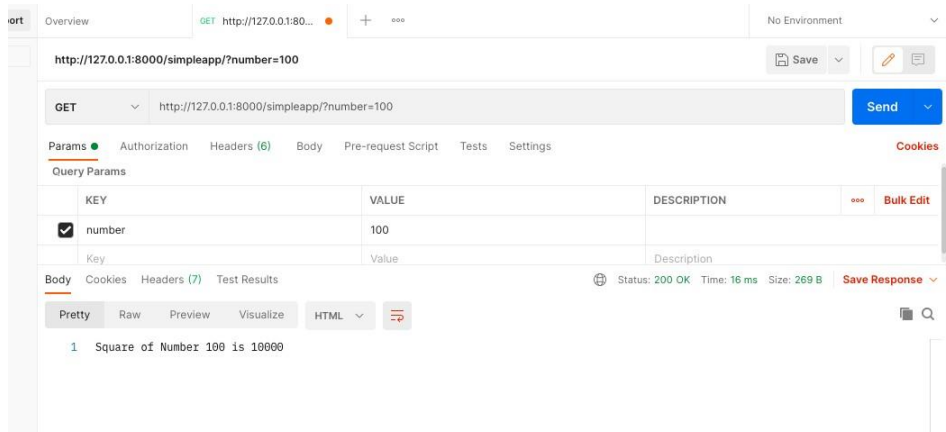
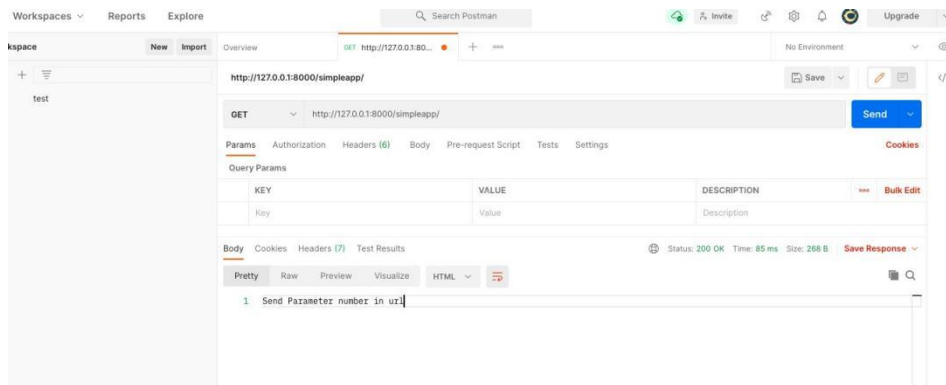
**With following HTTP requests types GET, PUT, POST, DELETE Calculate the square of a Number passed**

**Django URL name: array\_addition Parameter Name: number**

**GET:**

**# With No parameter passed/simpleapp/Response -> "Send Parameter number in url"  
/simpleapp/?number=10Response -> Square of Number 10 is 100"**

**Incorrect parameter Passed /simpleapp/?no.=10 Response -> Square of Number 10 is 100"**

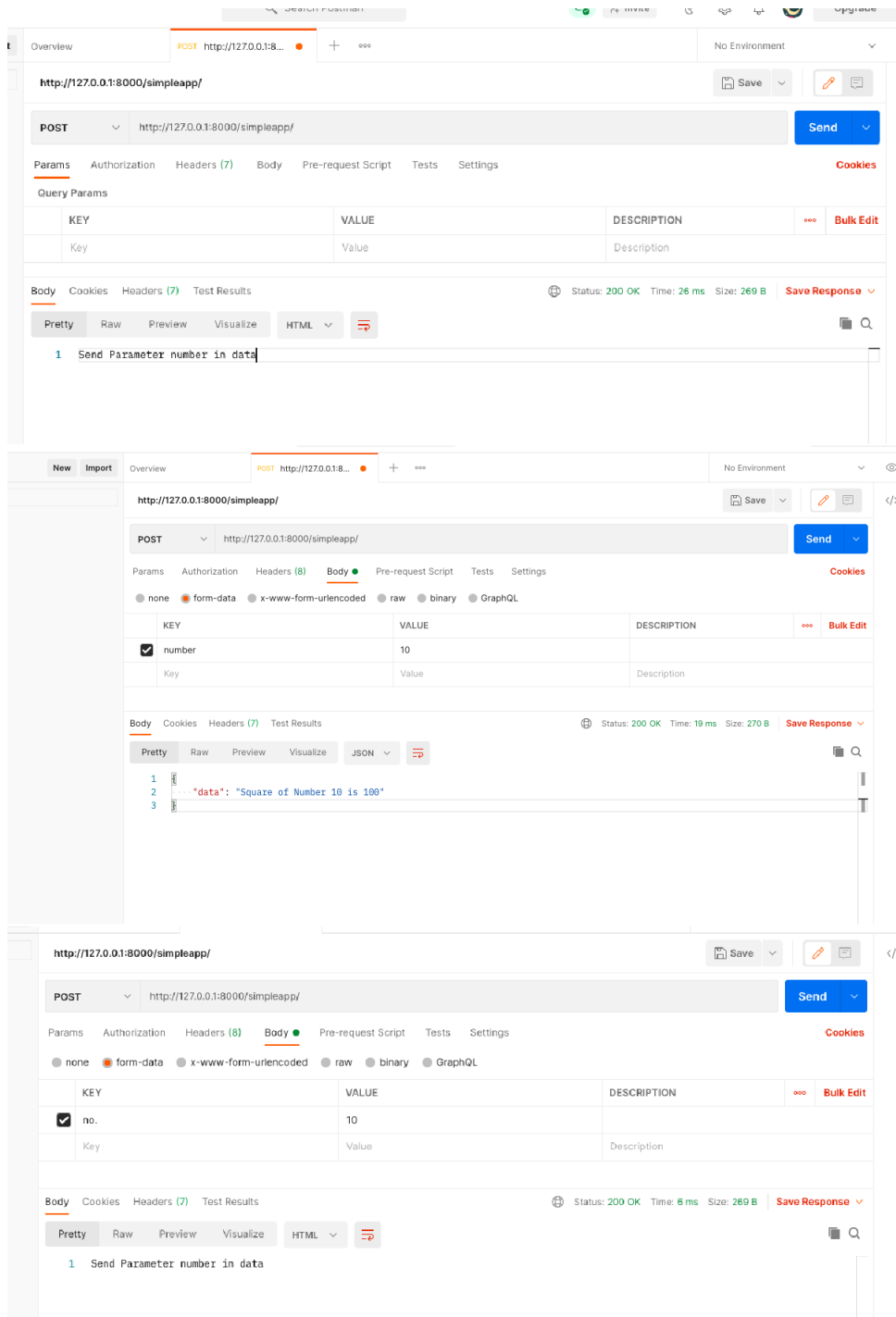


## POST:

# With No data passed /simpleapp/Response -> "Send Parameter number in data"

/simpleapp/ -d "number=15"Response -> {"data": "Square of Number 15 is 225"}

Incorrect parameter Passed/simpleapp/ -d "no.=15"Response -> "Send Parameter number in data"



**PUT :**  
**# With No parameter passed/simpleapp/Response -> "Send Parameter as /simpleapp/<number>"**  
**/simpleapp/15Response -> {"data": "Square of Number 15 is 225"}**

The image displays two screenshots of a REST client interface, likely Postman, showing the process of sending a PUT request and receiving a response.

**Top Screenshot:**

- URL:** `http://127.0.0.1:8000/simpleapp/`
- Method:** PUT
- Body:** Send Parameter as `/simpleapp/<number>`
- Status:** 200 OK, Time: 5 ms, Size: 277 B

**Bottom Screenshot:**

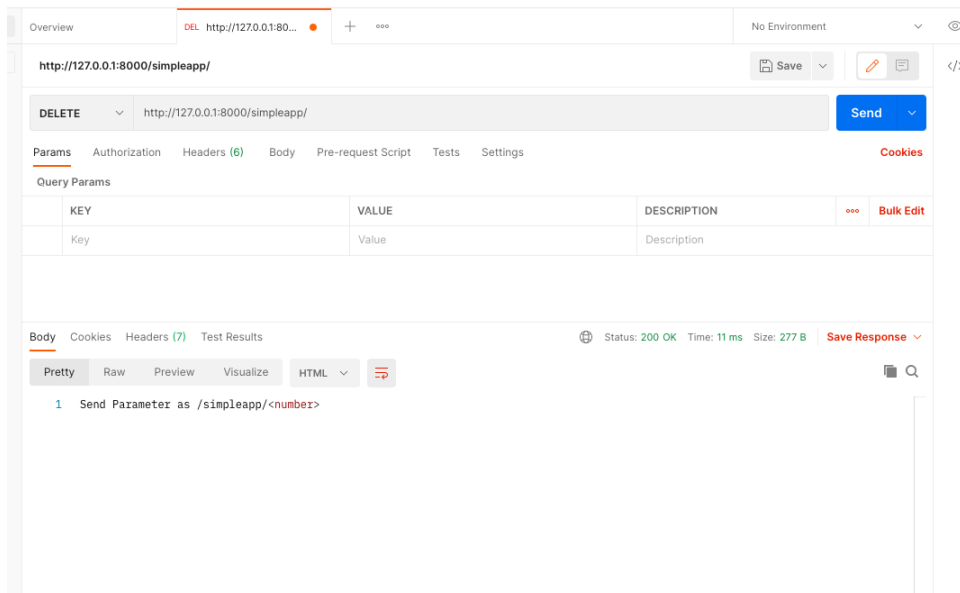
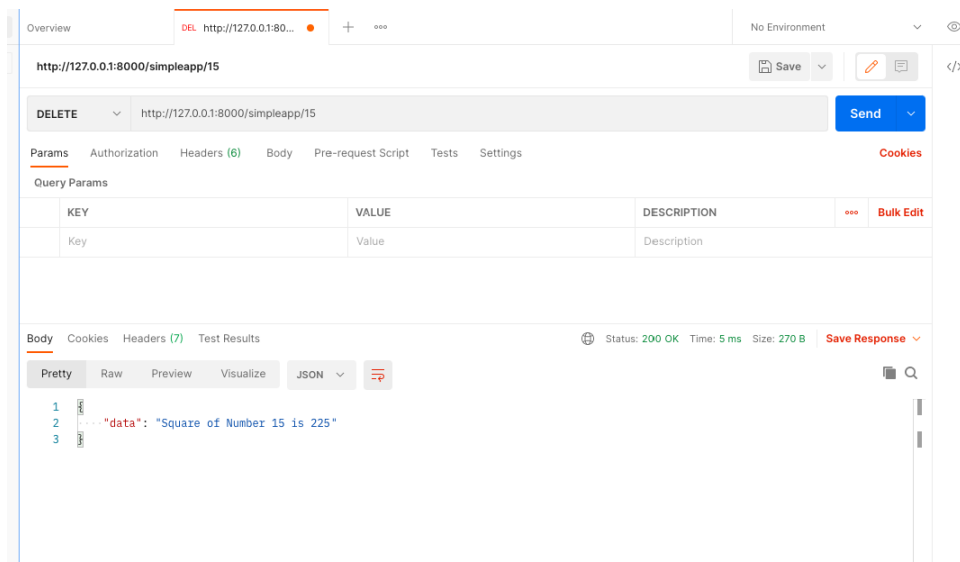
- URL:** `http://127.0.0.1:8000/simpleapp/15`
- Method:** PUT
- Body:**

```
{
  "data": "Square of Number 15 is 225"
}
```
- Status:** 200 OK, Time: 16 ms, Size: 270 B

## DELETE :

# With No parameter passed/simpleapp/Response -> "Send Parameter as /simpleapp/<number>"

/simpleapp/15Response -> {"data": "Square of Number 15 is 225"}



## Example Curl requests

**curl -X POST "http://127.0.0.1:8000/simpleapp/" -d "number=10" {"data": "Square of Number 10 is 100"}**

## Template 2 : Check whether a string is a palindrome

### GET

**# With No parameter passed/palindrome\_check/Response -> "Send Parameter string in URL to check palindrome"**

**/palindrome\_check/?string=calenderResponse -> "<b>calender</b> is not a palindrome"**

**Incorrect parameter Passed/palindrome\_check/?text=helloResponse -> "Send Parameter string in url to check palindrome"**

Overview GET http://127.0.0.1:8000/palindrome\_check/ No Environment

http://127.0.0.1:8000/palindrome\_check/ Save Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION		Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 5 ms Size: 288 B Save Response

Pretty Raw Preview Visualize HTML

```
1 Send Parameter string in url to check palindrome
```

Overview GET http://127.0.0.1:8000/palindrome\_check/?string=calender No Environment

http://127.0.0.1:8000/palindrome\_check/?string=calender Save Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION		Bulk Edit
<input checked="" type="checkbox"/> string	calender			
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 13 ms Size: 275 B Save Response

Pretty Raw Preview Visualize HTML

```
1 <b>calender</b> is not a palindrome
```

Overview GET http://127.0.0.1:8000/palindrome\_check/?text=hello No Environment

http://127.0.0.1:8000/palindrome\_check/?text=hello Save Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION		Bulk Edit
<input checked="" type="checkbox"/> text	hello			
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 14 ms Size: 288 B Save Response

Pretty Raw Preview Visualize HTML

```
1 Send Parameter string in url to check palindrome
```

## POST

# With No data passed/palindrome\_check/Response -> "Send Parameter string in data to check palindrome"

**/palindrome\_check/ -d "string=abbbba"Response -> {"result": "abbbba is a palindrome"}**

**/palindrome\_check/ -d "text=jjj"Response -> "Send Parameter string in data to check palindrome"**

Overview

POST http://127.0.0.1:8000/palindrome\_check/

No Environment

http://127.0.0.1:8000/palindrome\_check/

POST http://127.0.0.1:8000/palindrome\_check/ Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

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

This request does not have a body

Body Cookies Headers (7) Test Results Status: 200 OK Time: 9 ms Size: 289 B Save Response

Pretty Raw Preview Visualize HTML Send

1 Send Parameter string in data to check palindrome

Overview

POST http://127.0.0.1:8000/palindrome\_check/

No Environment

http://127.0.0.1:8000/palindrome\_check/

POST http://127.0.0.1:8000/palindrome\_check/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

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

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> string	abbbba	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 22 ms Size: 268 B Save Response

Pretty Raw Preview Visualize JSON Send

1 {  
2 "result": "abbbba is a palindrome"  
3 }

Overview

POST http://127.0.0.1:8000/palindrome\_check/

No Environment

http://127.0.0.1:8000/palindrome\_check/

POST http://127.0.0.1:8000/palindrome\_check/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

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

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> text	jjj	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 6 ms Size: 289 B Save Response

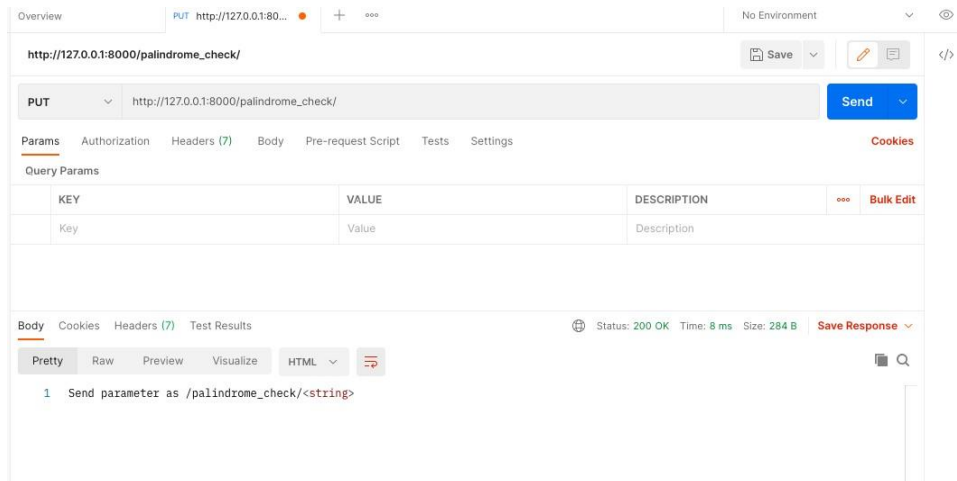
Pretty Raw Preview Visualize HTML Send

1 Send Parameter string in data to check palindrome

**PUT**

**# With No parameter passed/palindrome\_check/Response -> "Send parameter as /palindrome\_check/<string>"**

**/palindrome\_check/abbbbaResponse -> '{"result": "abbbba is a palindrome"}'**



## **DELETE**

**# With No parameter passed /palindrome\_check/ Response -> "Send parameter as /palindrome\_check/<string>"**

**/palindrome\_check/abbbbaResponse -> '{"result": "abbbba is a palindrome"}'**



The image displays two screenshots of a REST client interface, likely Postman, showing HTTP requests and responses for a palindrome check API.

**Top Screenshot:**

- URL:** `http://127.0.0.1:8000/palindrome_check/`
- Method:** DELETE
- Status:** 200 OK, Time: 7 ms, Size: 284 B
- Body:** Send parameter as `/palindrome_check/<string>`

**Bottom Screenshot:**

- URL:** `http://127.0.0.1:8000/palindrome_check/abbbba`
- Method:** DELETE
- Status:** 200 OK, Time: 5 ms, Size: 268 B
- Body:**

```
{
  "result": "abbbba is a palindrome"
}
```

### Example Curl Requests

**curl -X PUT "http://127.0.0.1:8000/palindrome\_check/a"**

**{"result": "a is a palindrome"}**

**curl -X POST "http://127.0.0.1:8000/palindrome\_check/" -d "string=12"**

**{"result": "12 is not a palindrome"}**

**Template 3 : Add all the numbers separated by Comma**

**With the following HTTP requests types GET, POST add all the numbers separated by Comma**

**Django URL name: index Parameter Name: array**

## GET

# With No parameter passed      /array\_addition/      Response -> "Send Parameter array as comma seperated numbers 2,3,4,5"

/array\_addition/?array=100,200,300,1000Response -> "<h1>Sum is 1600</h1>"

/array\_addition/?array=1,-1,-3,3    Response -> "<h1>Sum is 0</h1>"

Incorrect Parameter/array\_addition/?list=1,-1,-3,3Response -> "Send Parameter array as comma seperated numbers 2,3,4,5"

The screenshot displays a REST client interface with two test cases. The top test case is a GET request to `http://127.0.0.1:8000/array_addition/`. The response status is 200 OK, and the body contains the text "Send Parameter array as comma seperated numbers 2,3,4,5". The bottom test case is a GET request to `http://127.0.0.1:8000/array_addition/?array=100,200,300,1000`. The response status is 200 OK, and the body contains the HTML response `<h1>Sum is 1600</h1>`.

Overview GET http://127.0.0.1:80... + ... No Environment

http://127.0.0.1:8000/array\_addition/ Save </>

GET http://127.0.0.1:8000/array\_addition/ Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (7) Test Results Status: 200 OK Time: 6 ms Size: 295 B Save Response

Pretty Raw Preview Visualize HTML

1 Send Parameter array as comma seperated numbers 2,3,4,5

Overview GET http://127.0.0.1:80... + ... No Environment

http://127.0.0.1:8000/array\_addition/?array=100,200,300,1000 Save </>

GET http://127.0.0.1:8000/array\_addition/?array=100,200,300,1000 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> array	100,200,300,1000		
Key	Value	Description	

Body Cookies Headers (7) Test Results Status: 200 OK Time: 7 ms Size: 260 B Save Response

Pretty Raw Preview Visualize HTML

1 <h1>Sum is 1600</h1>

The image displays two screenshots of a REST client interface, likely Postman, showing the configuration and response of an HTTP GET request.

**Top Screenshot:**

- Request:** GET `http://127.0.0.1:8000/array_addition/?array=1,-1,-3,3`
- Params:** Query Params table with one entry: 

KEY	VALUE	DESCRIPTION
array	1,-1,-3,3	
- Body:** Pretty view shows the response: `<h1>Sum is 0</h1>`
- Status:** 200 OK, Time: 6 ms, Size: 257 B

**Bottom Screenshot:**

- Request:** GET `http://127.0.0.1:8000/array_addition/?list=1,-1,-3,3`
- Params:** Query Params table with one entry: 

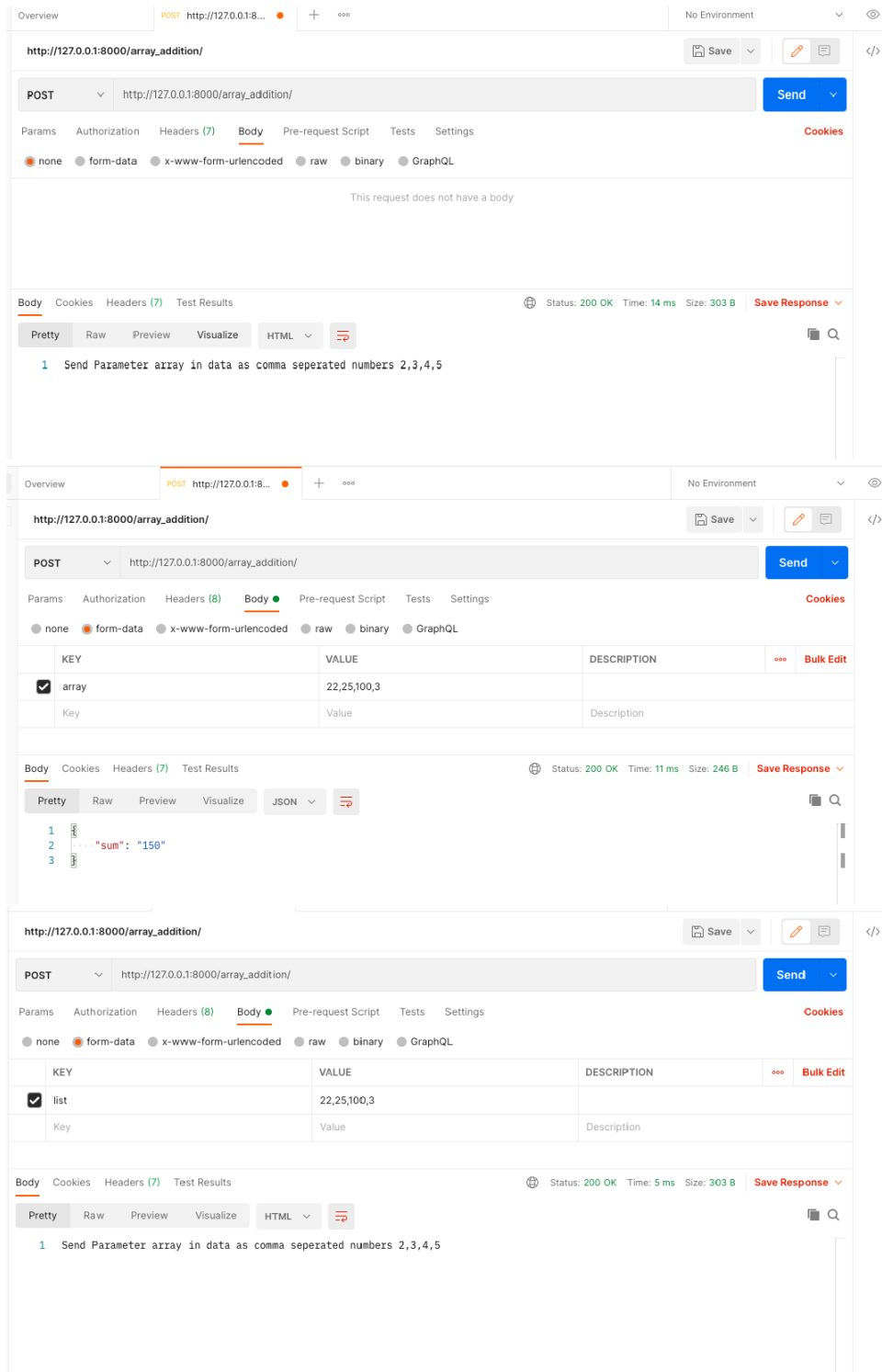
KEY	VALUE	DESCRIPTION
list	1,-1,-3,3	
- Body:** Pretty view shows the response: `Send Parameter array as comma seperated numbers 2,3,4,5`
- Status:** 200 OK, Time: 5 ms, Size: 295 B

## POST

**#With No data passed      /array\_addition/      Response -> "Send Parameter array in data as comma seperated numbers 2,3,4,5"**

**/array\_addition/ -d "array=22,25,100,3"Response -> {"sum": "150"}**

**Incorrect Parameter/array\_addition/ -d "list=22,25,100,3"Response -> {"sum": "150"}**



## Installation Steps

**Open up your Terminal / Command Line**

**git clone the repository**

**cd into the directory of the step (the one you just git cloned)**

**make sure you have python3 installed**

**Install virtualenv by following the steps**

**Mac**

```
python3 -m pip install --user virtualenv  
python3 -m venv envsource env/bin/activate
```

**Windows**

```
py -m pip install --user virtualenv  
py -m venv env.\env\Scripts\activate
```

**Run**

```
python -m pip install -r requirements/requirements.txt
```

**No Database setup needs to be done. Do not edit Database settings in settings.py**  
**If everything runs fine till here, create the application using the command**  
**python manage.py startapp simpleapp**

**Now uncomment the following lines**

**Line 21 in first\_assignment/urls.py -> path('', include(('simpleapp.urls', 'simpleapp'), namespace='simpleapp')) ,Line 40 in first\_assignment/settings.py -> 'simpleapp',**

**Now Create File simpleapp/urls.py and add lines**

```
from django.urls import pathfrom . import viewsurlpatterns = []
```

**Run**

```
python manage.py makemigrations
```

**Now Run the Server using command**

```
python manage.py runserver
```

**Great! Start Coding the assignment for above usecase**

**After Finishing the assignment, you can test them locally using command**  
**coverage run --source='.' manage.py test --no-input**

**Push the code into the repo using git, Check Pull Requests Tab.**

**Check if all the tests are Passed**