**Adding 2 numbers in Django**

In our before lecture we have created a dynamic page and did some branding i.e., add bgcolor cyan to it. Now we will look at performing basic operation that is adding two numbers on our site. Let’s take a look at how to do it.

* First we need to create form using which we can add 2 numbers. This form contains 2 text fields(input of numbers) as we need to add 2 numbers. br will make it appear in a new line. We need a button which has the name “**submit**”. Our home page should have the html code as shown below. Remember this is just a html code which doesn’t do any operation.

A screenshot of a computer

Description automatically generated with medium confidence

If you do the refresh of the page we will see the output as shown below.

Graphical user interface

Description automatically generated

* When you try to add two numbers and hit submit we will see the below screenshot and observe the url, we are hitting or pointing to the **add** as highlighted below.

Graphical user interface, text, application, email

Description automatically generated

* To handle this we need to add a path, will perform the mapping in **urls.py** as shown below, yes, since we haven’t created a add method we will get the error.

A screenshot of a computer

Description automatically generated with medium confidence

* Now let’s **add the add function in our views.py** where the actual operation takes place as shown below. We will print the result in **result.html** which is a new page

Text

Description automatically generated

* To print the result we will create a new page in templates folder with the name **“result.html”** and we will add the normal text result….

Graphical user interface, text, application

Description automatically generated

* If you give 2 values and print the output it will be like below as we are not printing the values, however we are just printing the result… by rendering the result.html page as shown below

Graphical user interface

Description automatically generated

* To add the values we need to modify our add function in the **views.py** page as shown below using the python code. Recollect that we need to fetch the values from num1 and num2. This is where we will use the request object, when a client send a request to a server, the client is sending an object and that object will have some values and here the values which we are passing are num1 and num2 is in that particular object. When the server gives another value it will be in the form of response. We will use GET http method as highlighted below and calculate the result and send it to the **result.html** as mentioned in the curly brackets highlighted below.

A screenshot of a computer

Description automatically generated with medium confidence

* Update the result.html with the below highlighted code as we are calling the result

A screenshot of a computer screen

Description automatically generated with medium confidence

* Now if we add 5 and 6 we will get the result 56 as shown below.

Graphical user interface, application

Description automatically generated

* How to fix the issue?

Recap: In the **home.html** we got a form and we are asking for the two values, the moment you click on **submit** the request will go to action **add.** Now as per the mapping **url.py** it will call the views.**add** functionwhich will trigger the **add** function in views.py with the values. It is in **result.html** we are printing the added value.

**GET vs POST Django**

HTTP protocol has multiple methods to work with and we are dealing with request and response.

Graphical user interface, application

Description automatically generated

Below image gives an idea about some of the request methods, **our focus is on GET and POST**

Diagram

Description automatically generated

Whenever you want to fetch a resource from a server, something like “hey I want an image that is a GET request” because we are expecting something from the server.

Graphical user interface, application

Description automatically generated

What if you want to send something to the server, may be you are filling a form and submit the data, which is **POST** method.

A screenshot of a computer

Description automatically generated with medium confidence

**So GET is to get information and POST is to submit the data**. We have other methods as well for example: **PUT** method is to update the resource or update the data on the server, **DELETE** method is to delete a resource or data on the server.

* In this scenario of adding two numbers we are sending the data to the server which should be **POST**. But in our code we had used **GET**

Make sure your webpage works as expected by checking the output.

Graphical user interface, text, application, chat or text message

Description automatically generated

* Whenever you want to fetch some data we need to pass some data as well, check the results on google page and observe the url we can see the keywords in the address bar. Since this is on google it might be fine.
* Think of this what if you are sending the username and password in the form, as an example we are sending 4 and 5 in the URL which is not secure. **We don’t want to send our data in the address bar.**

Graphical user interface, text

Description automatically generated

* Hence we need to use **POST** another reason is what if you have a form with multiple fields, we don’t want everything to be coming on the address bar and technically we should always use **POST**.
* Let’s modify our **home.html** add **method=POST**, by default if we don’t set the method it will be set to **GET** if you are mentioning specifically then it will consider it as **POST**
* Now if you try to add the two numbers, it will give the result as shown below and we can notice that atleast now the numbers are no more appearing in the URL. Let’s take a look at CSRF verification failed message.

Graphical user interface, text, application

Description automatically generated

* What happens is in the world of internet we have many resources, services along with huge amounts of user data, since we have something valuable online which needs to be protected, we should acknowledge the adversaries as well. Hackers or attackers and these attackers will have various attacking techniques. There are many techniques and one of them is CSRF(Cross site request forgery).

How CSRF works?

Donate for social cause [click here](http://www.google.com):

Diagram

Description automatically generated

* As a web developer we need to understand how various attacks works so that we can eliminate them before deploying the code to production. Here in this context to avoid this CSRF what Django gives you by default is **csrfviewMiddleware** in **settings.py** – ‘which means it says hey developer by default I am providing you the middleware please use it’.

Text

Description automatically generated

* However, in our code in **home.html** we are not using it. In order to use it, we will make use of jinja code and mention the **csrf token** as highlighted below. Update the **views.py** with **POST** instead of **GET**

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

* After updating the two files, when we try to run the addition operation we should get the output with no details in the address bar as shown below. It is done with the help of cookies. Not the cookie which we eat.

Graphical user interface, text, application, chat or text message

Description automatically generated

If you want to know more about cookies, <https://www.kaspersky.com/resource-center/definitions/cookies> this might help.