# PHP Lab 5 GET and POST

An HTML form can be used to create a place for users to enter input to the web server. It can be a .html file without any PHP. The form sends the input to a .php file on the server, where it is processed.

## User Input with GET

Now we'll put up a page that explains we're printing out powers and asks the user how many they would like to see (what number to stop at.) It will ask the user for input, and then call your php script with the number the user has entered.

### Sample HTML for a GET request

Here is sample HTML to create a form with textbox input for a GET request, from <https://www.w3schools.com/html/html_forms.asp> (modified)

<!DOCTYPE html>

<html>

<body>

<h2>Sample Get Form</h2>

<form action="/action\_page.php" method="get">

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname"><br><br>

<input type="submit" value="Submit">

</form>

<p>If you click the "Submit" button, the form-data will be sent to a page called "/action\_page.php".</p>

</body>

</html>

A screenshot of a computer program

Description automatically generated

The label line displays a name for the box, First name: in this case.

The input type="text" line says we will create a textbox and that it will be entered in the variable fname.

The input type="submit" line causes the submit button to be displayed.

If the user puts Ed in the textbox and clicks the submit button, the page will send this to the server (assuming the server is running on localhost):  
localhost/action\_page.php?fname=Ed

There needs to be a file on the server called action\_page.php that is coded to handle the entry for fname.

### Modify the sample HTML

Copy the sample HTML code and paste it into a file on your server in /var/www/html. Edit the sample HTML code for this lab. You'll probably want to change the H2, action\_page.php, fname, and label to suit your needs. Instead of first name, you will want to have the user input the number powers they want to compute. It was probably what you put in the $stop variable in the last lab.

Once your page is complete, you can test it by putting a number in the textbox and clicking submit. You haven't done the PHP part yet so you will get an error, but the URL in the browser should change to   
localhost/powersGet.php?stop=10  
(Your names may be different, but it should call your PHP page and give a number for the variable.

A screenshot of a computer

Description automatically generated

### Create the PHP file

Copy your PHP code from the last lab (powers) into a new file. Name the new file to match the name in the HTML form you created (I used powers-get.php, but you can use whatever you like.)

Change the line in your powers script where you determine how many times to run the loop. Your line may be something like:  
$stop = 5;

Replace it with something like this:  
$stop = $\_GET['name']; where 'name' is what you chose for <input type="text" id="fname" name="fname">.

When you run your new form, input a number, and click submit, it should call your powers script. Can you see the parameter for number of times being passed in the URL? That is how the GET method works.

Open your browser's web developer app and go to the Network section. When you click on the submit button you should be able to see what the browser sent to the server, and how the server replied.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## User Input with POST

Now, change your web page (or copy your HTML for GET into a new page for POST) so it uses the POST method instead of GET. Your parameter is not being passed in the URL this time, so you will have to use the browser webdev or use Wireshark to see how the parameter for number of times is passed to your php script.

The only change you must make to your HTML is to change GET to POST in this line:  
<form action="/action\_page.php" method="get"> becomes  
<form action="/action\_page.php" method="post">

In your PHP code, you will change \_GET to \_POST.  
$stop = $\_GET['name']; becomes  
$stop = $\_POST['name'];

Note that the ?stop=10 is no longer in the browser bar.  
A screenshot of a computer

Description automatically generated

Instead, it is in the body of the request.  
A screenshot of a computer

Description automatically generated

The server response is unchanged.

If you look at the request using Wireshark, you will see that the stop=10 has moved to the data at the end of the packet.  
A screenshot of a computer

Description automatically generated

If you want to use Wireshark on your VM, install it with:  
sudo apt install wireshark

You will need to run wireshark as root if you want to take packet captures. From a terminal, type  
sudo wireshark

Use the Loopback:lo interface for your capture.  
A screenshot of a computer

Description automatically generated

# Hand In

1. Submit your code (HTML and PHP) for the POST request.
2. This code suffers from a Denial of Service (DoS) vulnerability. You can make inputs that will keep the server busy for a long time; it could also flood the browser with data. What is an input that would cause this? How can you alter the code to fix it?