

Introduction

We have already installed the **XAMPP environment**. We also know the **php.ini** file and the **phpinfo** function. So now we are going to explore the content of the variable **\$_SERVER** which is injected to the running script, in it we can find information about the server environment and the client.


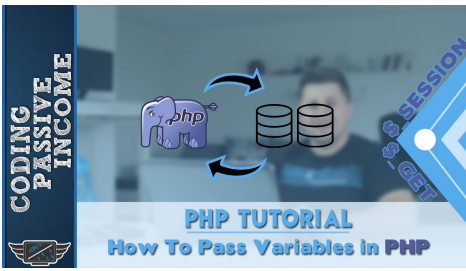



For instance, we can know the browser and the IP address of the client. Check the complete [list of available variables](#).

```
[REMOTE_ADDR] => 192.168.64.1
[HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; Intel Mac OS X
10_15_5) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.97 Safari/537.36
```

1. General analysis

1.1 Watch this videos about PHP functions:

	<h3>1. How to Start a Session in PHP</h3> <p>In this video you will learn how to start a PHP session</p> <p>https://www.youtube.com/watch?v=3CS-eQdcMLU</p>
	<h3>2. How to pass variables in PHP using Sessions and GET Method</h3> <p>In this PHP Tutorial, I will show you How To Pass Variables in PHP.</p> <p>https://www.youtube.com/watch?v=hNppPXJYJ60</p>
	<h3>3. PHP \$_SERVER SuperGlobal Array</h3> <p>In this video you will learn how to see information that is in the server variable</p> <p>https://www.youtube.com/watch?v=wD_V72pU15M</p>

1.2. Make an analysis of the local PHP server variable \$_SERVER

Create a php script called **server_variables.php** on the htdocs folder and dump out the content of the **\$_SERVER** variable with the **print_r** function. Once done, point your web browser to the local server url which is provided by XAMPP as we've already seen.

- Example URL:
 - http://localhost/server_variables.php

You will see something like this. You can add before the `print_r` line a `echo` `<pre>` ; to print it pretty.

```
Array
(
    [UNIQUE_ID] => XuIvUb-WZjgFV8ANKqoPEQAAAAM
    [HTTP_HOST] => 192.168.64.2
    [HTTP_CONNECTION] => keep-alive
    [HTTP_CACHE_CONTROL] => max-age=0
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.97 Safari/537.36
    [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
    [HTTP_ACCEPT_ENCODING] => gzip, deflate
    [HTTP_ACCEPT_LANGUAGE] => es-ES,es;q=0.9,en;q=0.8,fr;q=0.7,pt;q=0.6,ca;q=0.5,nl;q=0.4,da;q=0.3,gl;q=0.2,mt;q=0.1,la;q=0.1
    [PATH] => /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
    [LD_LIBRARY_PATH] => /opt/lampp/lib:/opt/lampp/lib
    [SERVER_SIGNATURE] =>
    [SERVER_SOFTWARE] => Apache/2.4.43 (Unix) OpenSSL/1.1.1g PHP/7.4.6 mod_perl/2.0.8-dev Perl/v5.16.3
    [SERVER_NAME] => 192.168.64.2
    [SERVER_ADDR] => 192.168.64.2
    [SERVER_PORT] => 80
    [REMOTE_ADDR] => 192.168.64.1
    [DOCUMENT_ROOT] => /opt/lampp/htdocs
    [REQUEST_SCHEME] => http
    [CONTEXT_PREFIX] =>
    [CONTEXT_DOCUMENT_ROOT] => /opt/lampp/htdocs
    [SERVER_ADMIN] => you@example.com
    [SCRIPT_FILENAME] => /opt/lampp/htdocs/server_variables.php
    [REMOTE_PORT] => 52407
    [GATEWAY_INTERFACE] => CGI/1.1
    [SERVER_PROTOCOL] => HTTP/1.1
    [REQUEST_METHOD] => GET
    [QUERY_STRING] =>
    [REQUEST_URI] => /server_variables.php
    [SCRIPT_NAME] => /server_variables.php
    [PHP_SELF] => /server_variables.php
    [REQUEST_TIME_FLOAT] => 1591881553.962
    [REQUEST_TIME] => 1591881553
)
```

1.3. Make an analysis of the local PHP server variable `$_SESSION`.

This variable like `$_SERVER` is injected to the **current script** and it holds information about the **client session**.

When we **browse an application**, we open it, then we do something there, and finally we close it. This can be called a **Session** with an application.

So the computer knows who we are. It knows when we **started** the **application** and when we **left** it.

Because the **HTTP address** is **stateless** the **web server** does not know who we are or what we were doing on the application.

To solve this problem **session variables** were added, they store **user information** to be used across multiple pages(**Login, home, profile..etc**). The duration of a session can be configured and by default it lasts until the **browser is closed**.

So we are going to **start a session** and dump the content of the **\$_SESSION** variable. Let's create a script called **session_variables.php** and add this.

```
<?php
session_start();

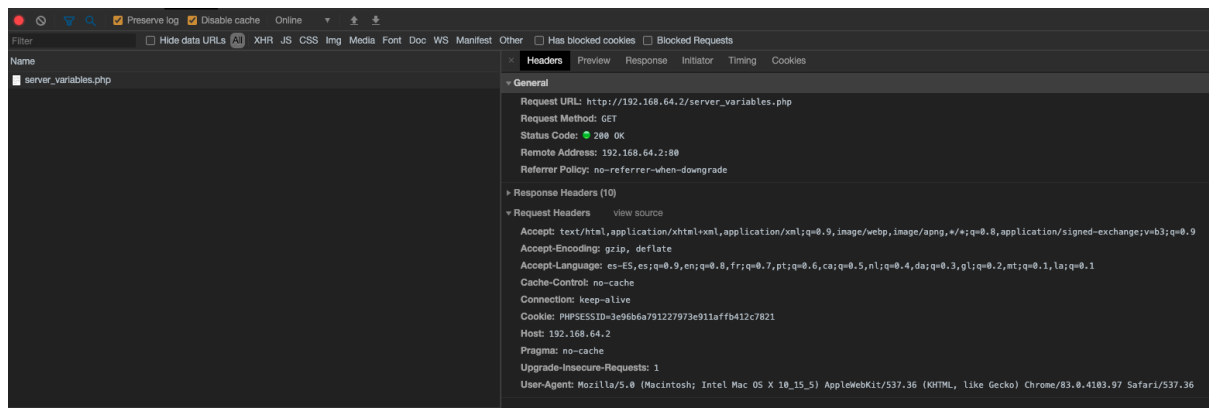
$_SESSION['browser'] = $_SERVER['HTTP_USER_AGENT'];
$_SESSION['ip'] = $_SERVER['REMOTE_ADDR'];
$_SESSION['time'] = time();
echo '<pre>';
print_r($_SESSION);
```

You will see a result similar to this:

```
Array
(
    [browser] => Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36
    [ip] => 192.168.64.1
    [time] => 1591884511
)
```

What happened here is that with **session_start()**, we have **started a session** which means that **PHP** is sending to the **browser** an identifier in the **request headers** as a **cookie** in order to **recognise** that **browser** or **client** the next time it connects again.

You can see the **request header** by opening the **dev tools** of your favourite browser and over the tab network **click on the script URL** and take a look at the **request headers**.



Cookie: PHPSESSID=3e96b6a791227973e911affb412c7821

Now you can identify a **client browser** and do amazing things like login and specialise the user **browsing experience**.

1.4. Configure the server and edit the php.ini file

Finally it is necessary that you **locate** and **edit** the file "**php.ini**".

You should also investigate what **PHP function** allows you to **find the path of your php.ini file**.

- Change the time zone to **Europe / Madrid**.
- Maximum execution time "**max_execution_time**" from **30** to **60** seconds.
- Add a **comment** at the **end** of the **file**.

Once the changes have been applied, you must call the **function** necessary to see the **PHP configuration** through a **PHP custom file**.

1.5. Session exercise

As a last exercise you must **create a login** which will allow you to access another **PHP file** and **view the content only** if you are **logged in**.

The **login** will ask for **email** and **password** (the **correct data** with which to compare the entered inputs will be in the **PHP code**).

Therefore you must create **four PHP files**:

- **"index.php"** : this will be the **login page with user and password fields and a login button**. This fields must be in a form pointing its action to the validation script. In case you **log in** or are **already logged in**, you must be automatically redirected to the panel.
- **"validate.php"**: All **login validations** must be done in this file receiving the **form data** by the **POST method** and this should **redirect the user** depending on whether the **login is correct or not**. You can use a simple string comparison or anything you want for deciding if the login is correct or not.
- **"panel.php"**: this file should **indicate** on the screen whether the **user** who is trying to access is **logged in or not**, in case the **user is logged**, it must also contain a **button to log out, destroy the current session** and then **redirect** you to the **"index.php"** file. **If not logged** you must **alert** the user that this is a **private section**.
- **"close_session.php"**: This is the last file which will be responsible for **destroying the session** and **redirecting to the login page**.

2. Pill Organization

It's important to document your own work in each pill or project you do. In this way you can easily after a time read your own resume of what you did in each project.

So since we use a repository we have a **README.md** file which must be used to document how our application works. In the resources section there are some links about github guidelines to document your project.

3. Deliverables

The following deliverables will be necessary to evaluate the project:

- Repository with the code

4. Resources

- \$_SERVER: <https://www.php.net/manual/en/reserved.variables.server.php>
- php.ini: <https://www.php.net/manual/es/configuration.file.php>
- Apache: <https://httpd.apache.org/>
- PHP ini loaded file:
<https://www.php.net/manual/es/function.php-ini-loaded-file.php>

4.1. Multimedia resources

	1. How to configure php.ini file? This video explains what is this file and how to configure it https://www.youtube.com/watch?v=ZvvUD6kQm8k
	2. How to login logout in PHP This video explains how to create a login and logout using PHP sessions. https://www.youtube.com/watch?v=O0Ky0tKvsJ8