

# PHP - Sessions

A web session is the time duration between the time a user establishes connection with a server and the time the connection is terminated. Along with the cookies, the session variables make the data accessible across the various pages of an entire website.

During a session, the website maintains information about the user's actions and preferences. The session data is populated in a superglobal associative array **\$\_SESSION**.

To start a new session in PHP, you need to call the **session\_start()** function.

## Starting a Session

In order to enable access to session data, `session_start()` function must be invoked. `session_start()` creates a session or resumes the current one based on a session identifier passed via a GET or POST request, or passed via a cookie.

```
session_start(array $options = []): bool
```

This function returns **true** if a session was successfully started, otherwise **false**.

PHP first creates a unique identifier for that particular session which is a random string of 32 hexadecimal numbers.

The **session\_id()** function sets or retrieves a unique session ID.

```
session_id(?string $id = null): string|false
```

PHP will generate a random session ID, if the **\$id** parameter is not given. You may specify your own ID instead. The function returns the session id for the current session or the empty string if there is no current session. On failure, it returns **false**.

## Example

Take a look at the following example –

```
</>
```

[Open Compiler](#)

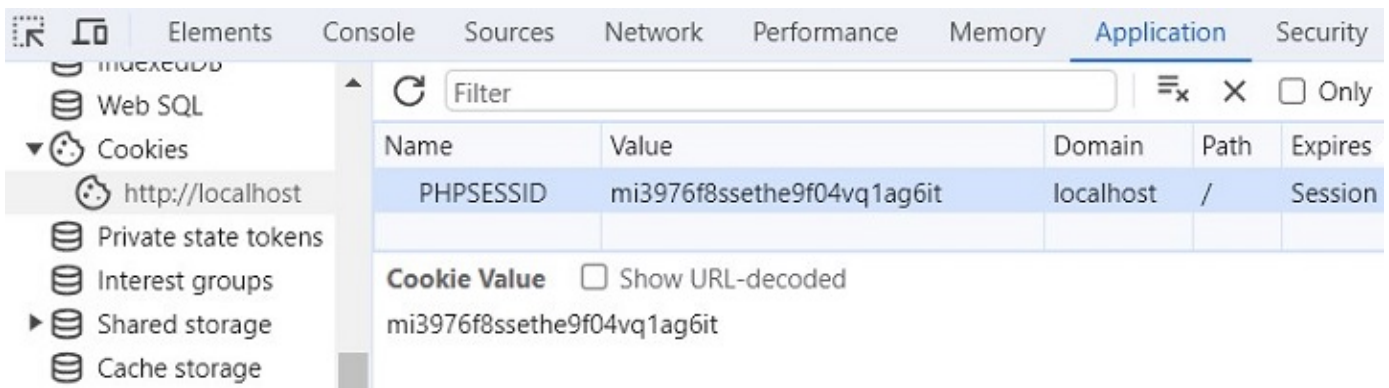
```
<?php
// Starting the session
session_start();
```

```
$id = session_id();  
echo "Session Id: ".$id ;  
?>
```

The browser will show a random string as the **output** –

Session Id: mi3976f8ssethe9f04vq1ag6it

A cookie called **PHPSESSID** is automatically sent to the user's computer to store unique session identification string.



A session creates a file in a temporary directory on the server where registered session variables and their values are stored. This data will be available to all pages on the site during that visit.

The location of the temporary file is determined by a setting in the "php.ini" file called "session.save\_path".

## Handling Session Variables

Session variables are stored in associative array called `$_SESSION[]`. These variables can be accessed during lifetime of a session.

To create a new session variable, add a key-value pair in the **`$_SESSION`** array –

```
$_SESSION[ "var" ]=value;
```

To read back the value of a session variable, you can use echo/print statements, or `var_dump()` or `print_r()` functions.

```
echo $_SESSION[ "var" ];
```

To obtain the list of all the session variables in the current session, you can use a foreach loop to traverse the `$_SESSION` –

```
foreach ($_SESSION as $key=>$val)
echo $key . "=>" . $val;
```

## Example

The following example starts a session then register a variable called counter that is incremented each time the page is visited during the session.

Use the **isset() function** to check if a session variable is already set or not.

The following PHP script starts a session when it runs for the first time, and sets a session variable named counter. When the client revisits the same URL again, since the session variable is already set, the counter is incremented.

&lt;/&gt;

Open Compiler

```
<?php
    session_start();
    if( isset( $_SESSION['counter'] ) ) {
        $_SESSION['counter'] += 1;
    } else {
        $_SESSION['counter'] = 1;
    }
    $msg = "Number of visits in this session: ". $_SESSION['counter'];
?>
<?php
    echo "$msg";
?>
```

Refresh the browser multiple times to simulate repeated visits. The browser displays the counter –

Number of visits in this session: 5

Explore our [latest online courses](#) and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

## Destroying a PHP Session

A PHP session can be destroyed by `session_destroy()` function. This function does not need any argument and a single call can destroy all the session variables. If you want to destroy a single session variable then you can use `unset()` function to unset a session variable.

Here is an example to **unset a single variable** –

```
<?php
    unset($_SESSION['counter']);
?>
```

Here is the call which will **destroy all the session variables** –

```
<?php
    session_destroy();
?>
```

You don't need to call `start_session()` function to start a session when a user visits your site if you can set `session.auto_start` variable to 1 in `php.ini` file.

## Example

The following PHP script renders a HTML form. The form data is used to create three session variables. A hyperlink takes the browser to another page, which reads back the session variables.

```
<html>
<body>
    <form action="<?php echo $_SERVER['PHP_SELF'];?>" method="post">
        <h3>User's ID: <input type="text" name="ID"/></h3>
        <h3>User's Name: <input type="text" name="name"/></h3>
        <h3>User Type: <input type="text" name="type"/></h3>
        <input type="submit" value="Submit" />
    </form>

    <?php
        session_start();
        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $_SESSION['ID'] = $_POST['ID'];
            $_SESSION['Name'] = $_POST['name'];
            $_SESSION['type'] = $_POST['type'];

            echo "<h2>Following Session variables Created</h2>";
            foreach ($_SESSION as $key=>$val) {
```

```
        echo "<h3>" . $key . "=>" . $val . "</h3>";
    }
    echo "<a href='test.php'><b>Click Here</b></a>";
}
?>
</body>
</html>
```

Save this code as "hello.php" in the document root folder, and open it in a client browser.

**User's ID:**

**User's Name:**

**User Type:**

Press the **Submit** button. The browser will show the session variables created –

## Following Session variables Created

**ID=>A001**

**Name=>Aman Gupta**

**type=>admin**

[Click Here](#)

The browser navigates to another page by following the link shown. It reads back the session variables.

## Following Session variables Read

**ID=>A001**

**Name=>Aman Gupta**

**type=>admin**