# Working with Sessions

Annapurna Agrawal
AUTHOR

@annapurna_23 linkedin.com/in/annapurna-agrawal

## Overview

Purpose of sessions

Starting, using and destroying a session

Working with session variables

Using sessions to preserve data

Storing session data in MySQL database using SessionHandlerInterface

## Sessions

Tool for managing the state of an application

Global variable stored on the web server

Stored in PHP super global $_SESSION array variable

Each session is associated with unique id, called Session ID or SID

## Sessions

Must be started before any output to the browser

Data on the server and the cookie in the browser work together

Server remembers you while working on an application using session

Automatically deleted when the browser is closed

## Why Sessions?

- To store data persistently for an application
- For frequently referred to data
- To store important information securely
- For more storage and smaller request size

## How Session Works?

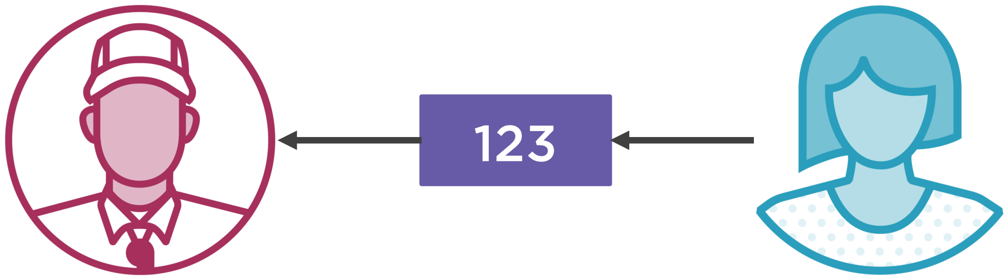
**HTTP is stateless**

**To preserve the application state**

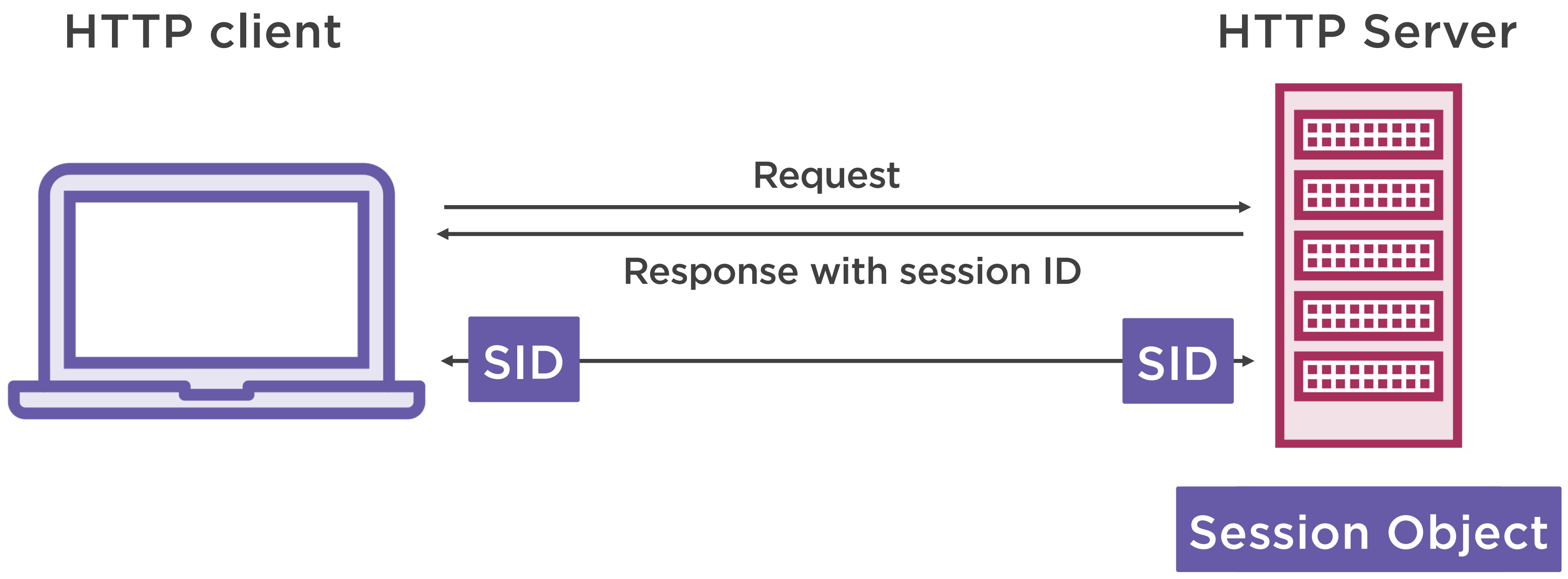
- Cookies shares user data between server and client
- Sessions uses server storage to store current user information

## How Session Works?

Unique Token



## How Session Works?



## Session Data & Session ID

**Session data can be stored in file or database**

**Session ID can be propagated using**

- Cookies
- URL parameters

**Using cookies to transfer the SID is recommended**

## Session Configurations

- php.ini file
- Apache configuration file, httpd.conf
- .htaccess file
- ini_set()

## Session Configuration Options

session.auto_start: automatically start session

Default is off

session.name: change name of current session & session cookie

Default session name is PHPSESSID

session.save_path: path where sessions data is stored on file system

Defaults is the server's tmp directory

## Session Configuration Options

session.gc_maxlifetime: specifies time in seconds for which session data is valid, default is 1440 seconds or 24 minutes

session.cookie_lifetime: specifies the lifetime of the cookie sent to browser, default is 0

session.cookie_path: sets the cookie path,  default is ‘/’

session.cookie_secure: specifies if cookie should be sent only over a secure http connection,  off by default

## Session Configuration Options

session.use_strict_mode: If enabled, rejects session ID that isn't initialized by server, disabled by default

session.cookie_httponly: access cookie through HTTP protocol, disabled by default

session.use_cookies: specifies if SID will be stored in cookies on client side, enabled by default

## Session Configuration Options

session.use_only_cookies: forces session to use only cookie for SID, works in conjunction with session.use_cookies

session.use_trans_sid: controls the use of transparent SID, disabled by default

session.cache_limiter: specifies cache control method used for session pages, default is nocache

session.cookie_samesite: controls the accessibility of the cookie in cross-domain requests, should be Lax or Strict

## Session Functions

session_start(): starts the session and makes $_SESSION available

session_name(): changes the name of session, same as session.name

session_id(): get and/or set the current session id

set new session ID: session_id('<new SID>')

session_destroy(): Destroys all data registered to a session

```php
session_start();

$_SESSION['username'] = 'Anna';

echo isset($_SESSION['username’]) ? $_SESSION['username’] : ‘Not available’;
```

## Initializing Session

**session_start()**

**Include in all scripts that needs access to session data**

**Checks if there is an existing session. If not, creates a new session**

**Stores data in PHP super global array variable $_SESSION**

## Destroying Session

**Session expires when**

- browser closes
- session timeout
- explicitly expired

**Destroying a session implies**

- clearing all session variables
- clearing server side data
- deleting client side session cookie

## Destroying Session

session_destroy()

unset()

## Destroying Sessions

session_destroy()

Destroy all data registered to a session

Does not unset any $_SESSION variable, or session cookie

Returns a boolean value

## Destroying Sessions

session_unset()

**To free all session variables**

**For deprecated code when $_SESSION is not used**

## Destroying Sessions

unset()

**To unset a session variable**

**For code where $_SESSION is used unset ($_SESSION['username']);.**

```php
$_SESSION = [];
$params = session_get_cookie_params();

$options = array('lifetime' => time() - 60)
setcookie(session_name(), '', $options);

session_destroy();
```

## Destroying a Session

**Clear the $_SESSION variable**

**Delete the session cookie using setcookie()**

**Destroy session using session_destroy()**

## Demo

Login form with sessions

## Session Storage

**Low load sites**

- store session data in file system

**High traffic sites or sites with multiple server**

- store session data in central location
- use database

## Demo

Login form with sessions

Store session data in MYSQL

Custom session handlers

## Overview

Login form with sessions

Store session data in MYSQL

Custom session handlers

Demo

Store session data in MYSQL using custom session handlers

## Custom Session Handler

```
session_set_save_handler (
    $sessionhandler,
    $register_shutdown) : bool
```

```
SessionHandlerInterface {

  public bool close();

  public bool destroy($sid);

  public bool gc($maxlifetime) ;

  public bool open($save_path, $sid);

  public string read($sid);

  public bool write($sid, $sess_data);

}
```

◀ close the session

◀ destroy a session

◀ clean up old session data

◀ open a new session

◀ read information of current session

◀ write the session data

## Summary

**Purpose of session**

**Work flow of session**

**Starting and using session**

**Completely destroy session**

- clear all session variables
- clear server side data
- delete client side session cookie

## Summary

**Session configuration**

**Session functions**

**Storing session data in MySQL database**

- using custom session handlers in PHP, which implements the SessionHandlerInterface

## Up Next:
## Session Security