SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

COURSE NAME : 19SB602 FULL STACK DEVELOPMENT FOR NEXT GENERATION IOT

III YEAR / VI SEMESTER

Unit III- **CORE PHP WITH MODEL–VIEW–CONTROLLER**

Topic :PHP JSON Functions, code igniter4

JSON

JSON stands for **JavaScript Object Notation,** and is a syntax for **storing and exchanging data.**

JSON is a **text format for storing and transporting data**

JSON is "**self-describing" and easy to understand**

JSON is a **lightweight data-interchange format**

JSON is **plain text written in JavaScript object notation**

JSON is used to **send data between computers**

JSON is **language independent ***

JSON Example

This example is a JSON string:

'{"name":"John", "age":30, "car":null}‘

It defines an object with 3 properties:

name
age
car

Each property has a value.

If you parse the JSON string with a JavaScript program, you can access the data as an object:

```
let personName = obj.name;
let personAge = obj.age;
```

PHP - json_encode()

The json_encode() function is **used to encode a value to JSON format.**

Example

This example shows how to encode an associative array into a JSON object:

```
<?php
$age = array("Peter"=>35, "Ben"=>37, "Joe"=>43);

echo json_encode($age);
?>
```

```php
<?php
$myObj = new stdClass();
$myObj->name = "John";
$myObj->age = 30;
$myObj->city = "New York";

$myJSON = json_encode($myObj);

echo $myJSON;
?>
```

```
{"name":"John","age":30,"city":"New York"}
```

PHP - json_decode()

The json_decode() function is **used to decode a JSON object** into a PHP object or an associative array.

Example

This example decodes JSON data into a PHP object:

```
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

var_dump(json_decode($jsonobj));
?>
```

The **json_decode() function returns** an **object by default.** The json_decode() function has a second parameter, and when set to true, JSON objects are decoded into associative arrays.

Example

This example decodes JSON data into a PHP associative array:

```
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

var_dump(json_decode($jsonobj, true));
?>
```

PHP - Accessing the Decoded Values

Here are two examples of how to access the decoded values from an object and from an associative array:

Example

This example shows how to access the values from a PHP object:

```
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

$obj = json_decode($jsonobj);

echo $obj->Peter;
echo $obj->Ben;
echo $obj->Joe;
?>
```

PHP - Looping Through the Values

You can also loop through the values with a foreach() loop:

Example

This example shows how to loop through the values of a PHP object:

```
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

$obj = json_decode($jsonobj);

foreach($obj as $key => $value) {
  echo $key . " => " . $value . "<br>";
}
?>
```

Code Igniter4

CodeIgniter a very popular and **powerful MVC framework** which is used to **develop web applications**. CodeIgniter known for its simplicity and ease of use.

Hello world example - CodeIgniter framework

In this example we will discuss about Hello world example CodeIgniter framework PHP.

We use two file for hello world example.

1. Hello.php (CodeIgniter\application\controllers\Hello.php)
2. hello_world.php (CodeIgniter\application\views\hello_world.php)

Hello.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Hello extends CI_Controller {

    public function index()
    {
        $this->load->view('hello_world');
    }
}
?>
```

hello_world.php

```
<!DOCTYPE html>
<html>
<head>
<title>Hello World !</title>
</head>
</body>

<h1>Hello World !</h1>

</body>
</html>
```

Remove index.php from URL in CodeIgniter using .htaccess

Create a .htaccess file in root folder and add this code.

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php/$1 [L]
```

Include header footer in CodeIgniter

Include header and footer is very easy in CodeIgniter.First of all create a common file in view folder.In this example i create a file name as start.php .

Start.php(View)

```
<?php
        include_once "header.php";
        include_once $pageName.'.php';
        include_once "footer.php";
?>
```

Home.php(view)

```
<!DOCTYPE html>
<html>
<head>
<title>Hello World !</title>
</head>
<body>

<h1>Main Content</h1>

</body>
</html>
```

header.php(View)

```
<h3>Header</h3>
```

footer.php(View)

```
<h3>Footer</h3>
```

Welcome.php

```
<?php
defined('BASEPATH') OR exit('No direct script access
allowed');

class Welcome extends CI_Controller {

        /**
         * Index Page for this controller.
         *
         * Maps to the following URL
         *
        http://example.com/index.php/welcome
         *        - or -
         *
```

```
http://example.com/index.php/welcome/index
	 *		- or -
	 * Since this controller is set as the default controller in
	 * config/routes.php, it's displayed at http://example.com/
	 *
	 * So any other public methods not prefixed with an
underscore will
	 * map to /index.php/welcome/<method_name>
	 * @see https://codeigniter.com/user_guide/general/urls.html
	 */
	public function index()
	{
		$data['pageName']= 'home';
		$this->load->view('start',$data);
	}
}
```

Session with example CodeIgniter framework

In CodeIgniter or any other framework session is used to store information (in variables) and used it through out the application.

Initializing Session

To store data in session first of all we need to initialize the session.

In PHP we initialize the session by simply write the session_start(); function.

But in CodeIgniter We can do that by executing the following line in constructor.

$this->load->library('session');

Add data to session

In php we simply use $_SESSION super global variable to add value to a session.

$_SESSION['key']=value;

Same thing can be done in CodeIgniter as shown below.

Remove Session Data

In PHP, we remove the data by using unset() function.
unset($_SESSION[‘some_name’]);

But in CodeIgniter we use unset_userdata() function that will remove data from session.

$this->session->unset_userdata(‘some_name');

Fetch Session Data

After set a data to a session we also retrieve it for our use. Userdata() function will be used for this purpose.

$name = $this->session->userdata('name');

Any Query????

Thank you……