

Data Access using PHP

Course Code: **CSC 3222**

Course Title: WEB TECHNOLOGIES



Dept. of Computer Science
Faculty of Science and Technology

Lab No:	4	Week No:	4	Semester:	
Lecturer:	<i>Supta Richard Philip & richard@aiub.edu</i>				

Lecture Outline



- 1. Learning outcome**
- 2. Data Access using PHP**
 - I. Data Read / Write in text file**
 - II. PHP SimpleXML - Read /write in xml File**
- 3. PHP Include Files**
- 4. Upload File in PHP**
- 5. Books and References**

Learning outcome



- In this lecture, we will learn read/write data using PHP.
- Discussing the techniques to read and write text files, XML and JSON using PHP script.
- We will also learn, PHP “**include function**” to add script to the other PHP script and file upload techniques in server.

Upload File in PHP



- Suppose we want to upload profile picture or any kind of file.

PROFILE PICTURE



Choose File No file chosen

Submit

```
<form action="upload.php" method="post" enctype="multipart/form-data">
```

Select image to upload:

```
<input type="file" name="myfile">
```

```
<input type="submit" value="Upload Image" name="submit">
```

```
</form>
```



Upload script

- There is one global PHP variable called **`$_FILES`**. This variable is an associate double dimension array and keeps all the information related to uploaded file.
- **`$_FILES['myfile']['tmp_name']`** - the uploaded file in the temporary directory on the web server.
- **`$_FILES['myfile']['name']`** - the actual name of the uploaded file.
- **`$_FILES['myfile']['size']`** - the size in bytes of the uploaded file.
- **`$_FILES['myfile']['type']`** - the MIME type of the uploaded file.
- **`$_FILES['myfile']['error']`** - the error code associated with this file upload.
- The following code upload file.
- `move_uploaded_file($_FILES["myfile"]["tmp_name"], "images/" . $_FILES["myfile"]["name"])`

JSON



- A JSON file is a file that stores simple data structures and objects in JavaScript Object Notation (JSON) format, which is a standard data interchange format.
- It is primarily used for transmitting data between a web application and a server. JSON files are lightweight, text-based, human-readable, and can be edited using a text editor.

JSON vs XML

Similarities:

- Both JSON and XML are "self describing" (human readable)
- Both JSON and XML are hierarchical (values within values)
- Both JSON and XML can be parsed and used by lots of programming languages
- Both JSON and XML can be fetched with an XMLHttpRequest

Differences:

- JSON doesn't use end tag
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays

JSON Example (employee.json)

```
{ "employees": [  
  { "firstName": "John", "lastName": "Doe" },  
  { "firstName": "Anna", "lastName": "Smith" },  
  { "firstName": "Peter", "lastName": "Jones" }  
]}
```

XML Example (employee.xml)

```
<employees>  
  <firstName>John</firstName> <lastName>Doe</lastName>  
</employee>  
<employee>  
  <firstName>Anna</firstName> <lastName>Smith</lastName>  
</employee>  
<employee>  
  <firstName>Peter</firstName> <lastName>Jones</lastName>  
  
</employees>
```


JSON Data Types



- a string { "name":"John" }
- a number { "age":30 }
- an object (JSON object)
 {
 "employee":{ "name":"John", "age":30, "city":"New
York" }
 }
- an array { "employees":["John", "Anna", "Peter"] }
- a Boolean { "sale":true }
- *Null* { "middlename":null }

PHP JSON Functions

- `json_encode()` - The `json_encode()` function is used to encode a value to JSON format.
- `json_decode()` - The `json_decode()` function is used to decode or convert a JSON object to a PHP object.
- `file_put_contents` - The `file_put_contents()` writes data to a file.
- `file_get_contents` - The `file_get_contents()` reads a file into a string.



JSON Write Example

```
//myform.php
<!DOCTYPE html>
<html>
<body>
<form action="process.php" method="POST">
  First name:<br>
  <input type="text" name="firstName">
  <br><br/>
  Last name:<br>
  <input type="text" name="lastName">
  <br><br>

  Email:<br>
  <input type="text" name="email">
  <br><br>

  Mobile:<br>
  <input type="text" name="mobile">
  <br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```



JSON Write Example

process.php

<?php

```
//Get form data
$formdata = array(
    'firstName'=> $_POST['firstName'],
    'lastName'=> $_POST['lastName'],
    'email'=>$_POST['email'],
    'mobile'=> $_POST['mobile']
);
//Convert array to JSON
$jsondata = json_encode($ formdata, JSON_PRETTY_PRINT);

//write json data into data.json file
if(file_put_contents("data.json", $jsondata)) {
    echo 'Data successfully saved';
}
else
    echo "no data saved";
```

?>



JSON Read Example

```
$str_data = file_get_contents("data.json");
$data = json_decode($str_data,true);
foreach($data as $mydata){
    foreach($mydata as $key=>$value)
    {
        echo $key . " is " . $value . "<br>";
    }
}
//or
var_dump( $data );
```



Books

- W3Schools Online Web Tutorials; URL: <http://www.w3schools.com>
- PHP Documentation; URL: <http://www.php.net/docs.php>
- Sams Teach Yourself Ajax JavaScript and PHP All in One; Phil Ballard and Michael Moncur; Sams Publishing; 2010
- JavaScript Phrasebook; Christian Wenz; Sams Publishing; 2007
- PHP and MySQL Web Development, 4/E; Luke Welling and Laura Thomson; Addison-Wesley Professional; 2009
- JavaScript for Programmers Paul J. Deitel and Harvey M. Deitel; Prentice Hall; 2009
- Beginning PHP5, Apache, and MySQL Web Development; Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz and Michael K. Glass; Wiley Publishing; 2005
- XML in a Nutshell, 3/E; Elliotte Rusty Harold and W. Scott Means; O'Reilly Media; 2004



References

1. <http://www.kodecrash.com/javascript/read-write-json-file-using-php/>
2. <https://gist.github.com/recked/7329961>
3. <https://www.kodingmadesimple.com/2016/05/how-to-write-json-to-file-in-php.html>
4. <https://code.tutsplus.com/tutorials/how-to-upload-a-file-in-php-with-example--cms-31763>
5. <https://www.tutorialrepublic.com/php-tutorial/php-file-upload.php>