Experiment No. 8

Title: REST APIs IN PHP

Batch: A3 **Roll No.:16010421073** **Experiment No.:8**

Aim: REST APIs IN PHP

Resources needed: Windows OS, Web Browser, Editor, XAMPP Server

---

Pre Lab/ Prior Concepts:
Students should have prior knowledge of HTML/CSS/Basic Programming, web services.

Theory:

HTTP is designed considering the REST (Representational State Transfer) architecture which defines how to develop and consume the web service. REST API provides easy to implement without less complexity and stateless web service. SOAP (Simple Object Access Protocol architecture) based communication is another approach web services can adopt.
REST API supports multiple data formats such as Command Separated Value (CSV)
, JavaScript Object Notation (JSON), Extensible Markup Language (XML) as compare to SOAP [1][2].

Many REST APIs use JSON (or JavaScript Object Notation) to carry responses from REST API endpoints. PHP natively supports converting data to JSON format from PHP variables and vice versa through its json extension [1].

To get a JSON representation of a PHP variable, use json_encode():

```
$data = array(1, 2, "three");
$jsonData = json_encode($data);
echo $jsonData;
```

Result will be : [1, 2, "three"]

Similarly, if you have a string containing JSON data, you can turn it into a PHP variable using json_decode():

```
<?php
$jsonData = "[1, \"KJSCE\", 2001]";
$data = json_decode($jsonData);
print_r($data);
?>
```

Result will be:

```
Array(
[0] => 1
[1] => KJSCE
[2] => 2001
```

)

There is no direct translation between PHP objects and JSON objects, these objects are an associative array.

Retrieving resources:

To retrieve the resource, GET method of HTTP is used. It uses the **curl** extension to format an HTTP request, set parameters on it, send the request, and get the returned information.
To retrieve information about the resource, first construct a URL representing the endpoint for that resource; it initializes a curl resource and provides the constructed URL to it. Finally, the curl object is executed, which sends the HTTP request, waits for the response, and returns it as demonstrated in the below code snippet.

```
<?php
$empId = '1';
$url = "http://localhost/test/rest1.php/{$empId}";
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
$response = curl_exec($ch);
$resultInfo = curl_getinfo($ch);
curl_close($ch);
// decode the JSON
$empJson = json_decode($response);
Print_r($empJson);
?>
```

In this case, the response is JSON data, which is decoded and handed off for future processing. The PUT method of HTTP is used to update the resource and DELETE method of HTTP is used to delete the resource.

Activity:

- Implement REST API to retrieve information of a resource
- Accept the required resource details from the user and using REST API retrieve information of the required resource.

Code:

Rest_Api.php

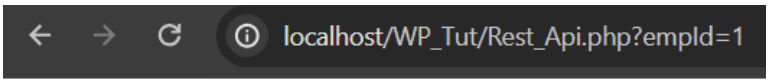
```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Rest Api</title>
</head>
<body>
    <form method="get">
        <input type="text" name="empId" id="empId" placeholder="Enter Employee ID">
        <input type="submit" value="Submit">
    </form>
    <?php
    if (isset($_GET['empId'])) {
        $empId = $_GET['empId'];
        $url = "http://localhost/WP_Tut/rest.php/?id=" . $empId;
        $client = curl_init();
        curl_setopt_array($client, [
            CURLOPT_RETURNTRANSFER => 1,
            CURLOPT_URL => $url
        ]);
        $result = curl_exec($client);
        curl_close($client);
        echo "Response from the server";
        echo "<pre>" . print_r($result, true) . "</pre>";
    }
    ?>
</body>
</html>
```

Rest.php

```php
<?php
$data = array(
    array("id" => 1, "name" => "Keyur", "age" => 20),
    array("id" => 2, "name" => "Harsh", "age" => 22),
    array("id" => 3, "name" => "Aditya", "age" => 24),
    array("id" => 4, "name" => "Elvish", "age" => 27),
    array("id" => 5, "name" => "bhavya", "age" => 32)
);
$id = $_GET['id'];

echo json_encode($data[$id - 1], JSON_PRETTY_PRINT);
?>
```

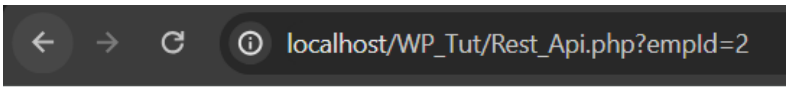
Output(Code with result Snapshot)



Enter Employee ID Submit

Response from the server

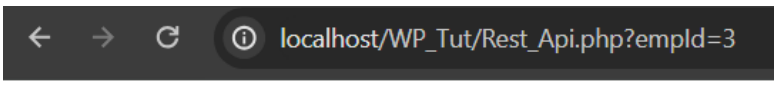

```
{
    "id": 1,
    "name": "Keyur",
    "age": 20
}
```

Enter Employee ID Submit

Response from the server

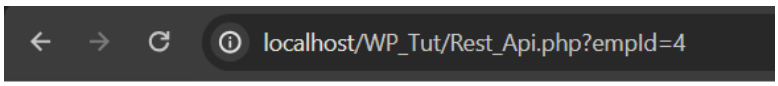

```
{
    "id": 2,
    "name": "Harsh",
    "age": 22
}
```

Enter Employee ID Submit

Response from the server

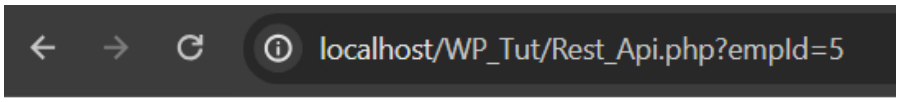

```
{
    "id": 3,
    "name": "Aditya",
    "age": 24
}
```

Enter Employee ID Submit

Response from the server

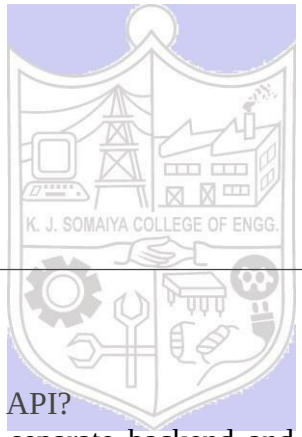

```
{
    "id": 4,
    "name": "Elvish",
    "age": 27
}
```

Enter Employee ID Submit

Response from the server


```
{
    "id": 5,
    "name": "bhavya",
    "age": 32
}
```



Questions:-

1. What is the purpose of REST API?

   Ans: Rest API allows us to separate backend and frontend and allow cross connection between various tech stacks, it formats the data into json or XML which can then be converted to the types specific to that particular language. REST API is powerful as a php backend can be connected to a react frontend. Thus allowing greater opportunity for improving tech stack if needed.

2. What are the other alternatives to REST API?

   Ans: Alternatives are

   - GraphQL
   - gRPC
   - websockets
   - Event driven architecture
   - FALCOR

Outcomes:

CO 4 : Demonstrate the use advanced features such as REST API, email handling, localization and internationalization in PHP.

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

Implemented and called REST API made in this assignment.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books:

1. Thomson PHP and MySQL Web Development Addison-Wesley Professional, 5th Edition 2016.
2. Peter MacIntyre, Kevin Tatroe Programming PHP O'Reilly Media, Inc, 4th Edition 2020
3. Frank M. Kromann Beginning PHP and MySQL: From Novice to Professional, Apress 1st Edition, 2018