



گزارش پروژه درس یکپارچه سازی
پیاده سازی وب سرویس مبتنی بر REST

نام دانشجو:

احسان بیکی فندخت

۹۷۱۲۳۲۵۰۱۵

نام استاد:

حسین مرادی

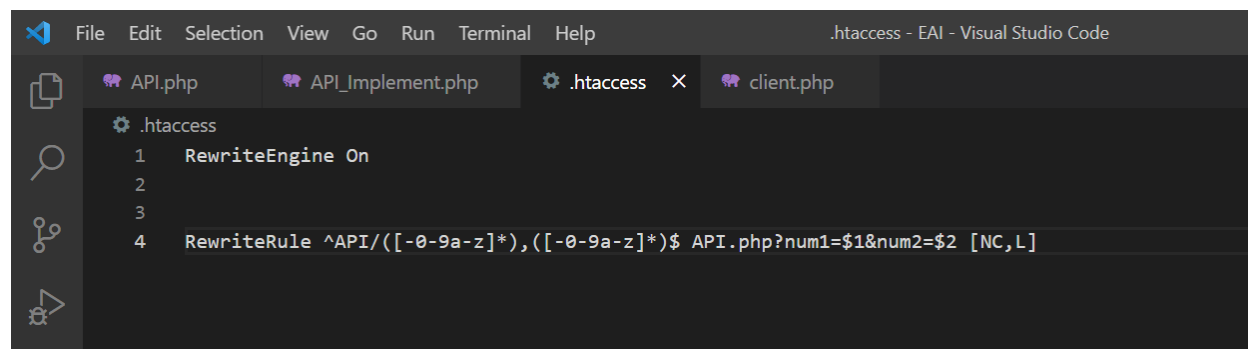
تیر ۱۴۰۲

فهرست مطالب

۲ htaccess
۳ API_Implement.php
۳ API.php
۵ client.php
۷ بررسی متد GET با ابزار POSTMAN
۸ بررسی متد POST با ابزار POSTMAN
۹ فراخوانی API با کمک رابط کاربری

در این پروژه به پیاده سازی یک وب سرویس ساده پرداخته شده است که دو ورودی گرفته و حاصل ضرب آن دو را محاسبه و در پاسخ برمیگرداند. در این پروژه چهار فایل مهم (htaccess, API.php, client.php, API_Implement.php) و تعدادی فایل برای استایل دهی به رابط کاربری (موجود در پوشه Style) وجود دارد که در ادامه به شرح آن ها خواهیم پرداخت.

.htaccess



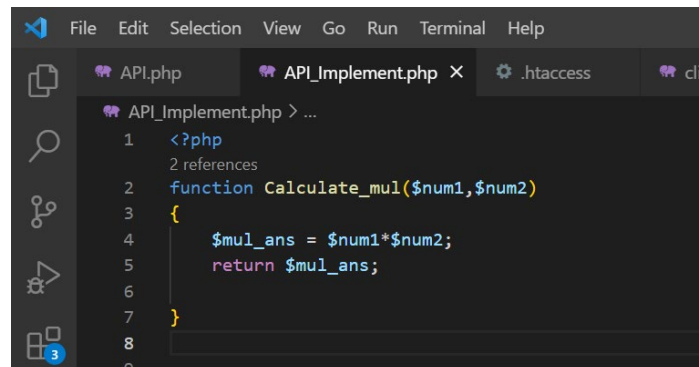
برای تغییر آدرس (redirect) به فرم GET استفاده می شود.

برای مثال آدرس localhost/EAI/API/5,8 را به آدرس localhost/EAI/API.php?num1=5&num2=8 تغییر مسیر می دهد.

عبارت باقاعده نوشته شده هر نوع ورودی را گرفته و بررسی اینکه نوع ورودی عددی است در بخش لاجیک انجام میشود.

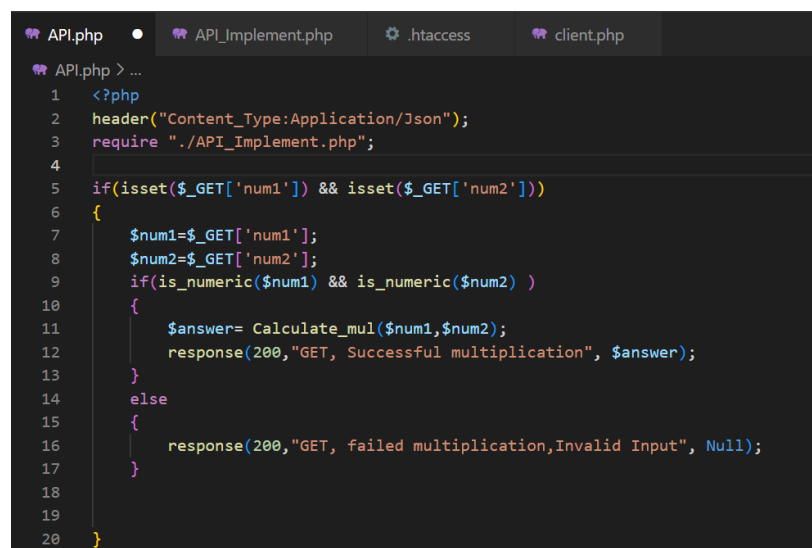
API_Implement.php

شامل یک تابع ساده است که دو عدد را به عنوان ورودی گرفته و حاصل ضرب آن دو را بر می گرداند.



```
File Edit Selection View Go Run Terminal Help
API.php API_Implement.php X .htaccess client.php
API_Implement.php > ...
1 <?php
2 2 references
3 function Calculate_mul($num1,$num2)
4 {
5     $mul_ans = $num1*$num2;
6     return $mul_ans;
7 }
8
9
```

API.php



```
API.php API_Implement.php .htaccess client.php
API.php > ...
1 <?php
2 header("Content_Type:Application/Json");
3 require "../API_Implement.php";
4
5 if(isset($_GET['num1']) && isset($_GET['num2']))
6 {
7     $num1=$_GET['num1'];
8     $num2=$_GET['num2'];
9     if(is_numeric($num1) && is_numeric($num2) )
10    {
11        $answer= Calculate_mul($num1,$num2);
12        response(200,"GET, Successful multiplication", $answer);
13    }
14    else
15    {
16        response(200,"GET, failed multiplication,Invalid Input", Null);
17    }
18
19
20 }
```

در قطعه کد اول برنامه به متد GET سرویس میدهد به گونه ای که دو عدد ورودی را به کمک متغیر گلوبال \$_GET از URL استخراج میکنند و سپس در صورت عددی بودن ورودی ها، حاصل ضرب آن دو را به وسیله تابع موجود در API_Implement.php محاسبه می کند. پاسخ وب سرویس بسته به شرایط به کمک تابع response فراهم می شود.

```

22 else if(isset($_POST['num1']) && isset($_POST['num2']))
23 {
24     $num1=$_POST['num1'];
25     $num2=$_POST['num2'];
26     if(is_numeric($num1) && is_numeric($num2) )
27     {
28         $answer= Calculate_mul($num1,$num2);
29         response(200,"POST, Successful multiplication", $answer);
30     }
31     else
32     {
33         response(200,"POST, failed multiplication,Invalid Input", Null);
34     }
35
36
37 }
38 else
39 {
40     response(200,"failed multiplication", Null);
41 }

```

در قطعه کد بعدی برنامه به متد POST سرویس میدهد به گونه مشابه با متد قبلی، دو عدد ورودی را به کمک متغیر گلوبال \$_POST از بدنه درخواست (request) استخراج میکند و سپس در صورت عددی بودن ورودی ها، حاصل ضرب آن دو را به وسیله تابع موجود در API_Implement.php محاسبه می کند. پاسخ وب سرویس بسته به شرایط به کمک تابع response فراهم می شود.

```

5 references
42 function response($status,$message,$Output)
43 {
44     header("HTTP/1.1 ".$status);
45
46     $reply['status']=$status;
47     $reply['message']=$message;
48     $reply['Output']=$Output;
49
50     $json_reply = json_encode($reply);
51     echo $json_reply;
52 }
53
54 ?>
55

```

در ادامه نیز تابع response پیاده سازی شده است که محتوای پیغام پاسخ را از ورودی گرفته و آن را در قالب جیسون (json) ذخیره و نمایش می دهد.

client.php

```
API.php  API_Implement.php  .htaccess  client.php

client.php > html
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <title>User Interface</title>
6      <meta charset="utf-8">
7      <link type="text/css" rel="stylesheet" href="Style/css/client.css" />
8
9  </head>
10
11
12 <body>
13 <div class="Content">
14 <form class="Form" action="" method="post">
15
16     <label for="">عدد اول را وارد کنید</label><br>
17     <input type="text" name="num1" required> <br><br>
18
19     <label for="">عدد دوم را وارد کنید</label><br>
20     <input type="text" name="num2" required> <br><br>
21
22     <button class="clickBtn" type="submit" name="submit">محاسبه</button>
23
24 </form>
25 </div>
26 </body>
27 </html>
28 <?php
```

این فایل مربوط به لایه رابط کاربری است که به صورت گرافیکی ورودی ها را دریافت میکند و با فراخوانی API خروجی را بدست آورده و نمایش میدهد.

```

24 <?php
25 if (isset($_POST['submit']))
26 {
27     $num1=$_POST['num1'];
28     $num2=$_POST['num2'];
29     $url="http://localhost/EAI/API/".$num1.'.'.$num2;
30     $client = curl_init($url);
31     curl_setopt($client,CURLOPT_RETURNTRANSFER,TRUE);
32
33     $reply = curl_exec($client);
34     $result = json_decode($reply);
35
36     // echo $reply;
37
38     $Output=$result->Output;
39
40     echo '<label for="">حاصل ضرب</label><br>';
41     echo '<input type="" id="output" value="'. $Output .' " readonly> <br>';
42
43     echo '<label for="">پاسخ وب سرویس</label><br>';
44     echo '<p class="WebService" >'.$reply.'</p><br>';
45
46 }
47
48 ?>
49
50 </form>
51 </div>
52

```

در این قطعه کد ورودی هایی که در فرم با متد post ارسال شده اند را بررسی و آنها را درون url قرار می دهد. سپس به کمک API داده ها را مبادله کرده و خروجی به صورت JSON درون متغیر reply ذخیره می شود. سپس ساختار json دیکود شده و مقدار ضرب را استخراج میکند. در ادامه حاصل ضرب و پیغام وب سرویس درون رابط کاربری نمایش داده می شود.

بررسی متد GET با ابزار POSTMAN

localhost/EAI/API/0,F

The screenshot shows the Postman interface with a GET request to `localhost/EAI/API/0,F`. The response is a JSON object with status 200, message "GET, failed multiplication, Invalid Input", and output null.

```
1 { "status": 200, "message": "GET, failed multiplication, Invalid Input", "Output": null }
```

localhost/EAI/API/24,3

The screenshot shows the Postman interface with a GET request to `localhost/EAI/API/24,3`. The response is a JSON object with status 200, message "GET, Successful multiplication", and output 72.

```
1 { "status": 200, "message": "GET, Successful multiplication", "Output": 72 }
```

بررسی متد POST با ابزار POSTMAN

localhost/EAI/API.php

The screenshot shows the Postman application interface. On the left, the 'Scratch Pad' sidebar lists several API endpoints. The main workspace displays a POST request to 'localhost/EAI/API.php'. The 'Body' tab is selected, showing 'x-www-form-urlencoded' data with two parameters: 'num1' with value '10' and 'num2' with value '20'. The 'Send' button is visible. Below the request, the response is shown in 'Pretty' format, indicating a 200 OK status with a message: 'POST, Successful multiplication' and an output of '200'.

Key	Value	Description
num1	10	
num2	20	

```
1 {"status":200,"message":"POST, Successful multiplication","Output":200}
```

localhost/EAI/API.php

The screenshot shows the Postman application interface with a failed POST request to 'localhost/EAI/API.php'. The 'Body' tab is selected, showing 'x-www-form-urlencoded' data with two parameters: 'num1' with value 'g' and 'num2' with value 'v'. The 'Send' button is visible. Below the request, the response is shown in 'Pretty' format, indicating a 200 OK status with a message: 'POST, failed multiplication, Invalid Input' and an output of 'null'.

Key	Value	Description
num1	g	
num2	v	

```
1 {"status":200,"message":"POST, failed multiplication, Invalid Input","Output":null}
```


فراخوانی API با کمک رابط کاربری

قبل از محاسبه: (نمونه اول)

The screenshot shows a web browser window with the title "User Interface" and the address bar displaying "localhost/EAI/client.php". The interface has a dark background with a grid of small white dots on the left and right sides. In the center, there are two input fields. The first input field is labeled "عدد اول را وارد کنید:" (Enter the first number:) and contains the value "42". The second input field is labeled "عدد دوم را وارد کنید:" (Enter the second number:) and contains the value "2". Below the input fields is a yellow button labeled "محاسبه" (Calculate). The interface is decorated with diagonal lines and a small orange circle at the bottom center.

پس از محاسبه: (نمونه اول)

The screenshot shows the same web browser window as before, but the interface has updated. The input fields are now empty. Below the input fields, the text "محاسبه" (Calculate) is still present. Below that, the text "حاصل ضرب" (Product) is displayed. The output field shows the value "84". At the bottom, there is a white box containing the JSON response:

```
{\"status\":200,\"message\":\"GET, Successful multiplication\",\"Output\":\"۸۴\"}
```

. The interface is decorated with diagonal lines and a small orange circle at the bottom center.

قبل از محاسبه: (نمونه دوم)

The screenshot shows a web browser window with the title "User Interface" and the address bar displaying "localhost/EAI/client.php". The interface has a dark background with a grid of small white dots on the left and right sides. In the center, there is a rounded rectangle containing two input fields and a button. The first input field is labeled "عدد اول را وارد کنید:" and contains the letter "h". The second input field is labeled "عدد دوم را وارد کنید:" and contains the letter "R". Below these fields is a button labeled "محاسبه".

پس از محاسبه: (نمونه دوم)

The screenshot shows the same web browser window as before, but the interface has changed. The central rounded rectangle now contains two empty input fields, a button labeled "محاسبه", a label "حاصل ضرب", and a text box displaying the JSON response: {"status": 200, "message": "GET, failed multiplication, Invalid Input", "Output": null}. The background and side elements remain the same.