


Gravador, Keith Yrrel A.
BSIT - 3K


Activity 3

Making HTTP GET Request in PHP



PHP Online Compiler

Premium Coding Courses by Programiz

[Programiz PRO >](#)

main.php

Run

Share

Clear

```
1 <?php
2 $response = '{ "id":1,"title":"Sample Data","body":"This is a mock API
   response."}';
3
4 // Check if response exists
5 if ($response === FALSE) {
6     // Handle error
7     echo "Failed to retrieve data.";
8 } else {
9     // Decode JSON
10    $data = json_decode($response, true);
11
12    // Print formatted data
13    echo "ID: " . $data["id"] . "\n";
14    echo "Title: " . $data["title"] . "\n";
15    echo "Body: " . $data["body"] . "\n";
16 }
17 ?>
18
```


Output

Clear

```
ID: 1
Title: Sample Data
Body: This is a mock API response.


=== Code Execution Successful ===
```

Making HTTP POST Requests in PHP



PHP Online Compiler

Premium Coding Courses by Programiz

[Programiz PRO >](#)

main.php

Run

Share

Clear

```
1 <?php
2 $data = array(
3     'name' => 'john doe',
4     'email' => 'john.doe@example.com'
5 );
6
7 // Normally we'd send this to an API, but Programiz can't access the
   internet.
8 // So let's simulate a response as if the API sent it back.
9 $response = array(
10     "status" => "success",
11     "message" => "Data received successfully",
12     "received_data" => $data
13 );
14
15 // Convert to JSON (like a real API would respond)
16 $result = json_encode($response, JSON_PRETTY_PRINT);
17
18 // Output
19 echo $result;
20 ?>
21
```


Output

Clear

```
{
  "status": "success",
  "message": "Data received successfully",
  "received_data": {
    "name": "john doe",
    "email": "john.doe@example.com"
  }
}


=== Code Execution Successful ===
```

Using cURL for Robust HTTP Requests



PHP Online Compiler

Premium Coding Courses by Programiz



Programiz PRO >

main.php

Run

Share

Clear


```
1 <?php
2 $response = json_encode(array(
3     "id" => 123,
4     "name" => "Sample Resource",
5     "status" => "active"
6 ), JSON_PRETTY_PRINT);
7
8
9 echo $response;
10 ?>
11
```

Output

```
{
  "id": 123,
  "name": "Sample Resource",
  "status": "active"
}


=== Code Execution Successful ===
```

Working with HTTP Headers in PHP



PHP Online Compiler

Premium Coding Courses by Programiz



Programiz PRO >

main.php

Run

Share

Clear


```
1 <?php
2 $ch = curl_init("https://api.example.com/protected-resource");
3
4 $headers = array(
5     'User-Agent: MyCustomApp/1.0',
6     'Authorization: Bearer YOUR_API_TOKEN'
7 );
8
9 curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
10
11 $response = json_encode(array(
12     "status" => "success",
13     "message" => "Headers applied correctly",
14     "headers_sent" => $headers
15 ), JSON_PRETTY_PRINT);
16
17 // Output fake response
18 echo $response;
19 ?>
20
```

Output

```
{
  "status": "success",
  "message": "Headers applied correctly",
  "headers_sent": [
    "User-Agent: MyCustomApp/1.0",
    "Authorization: Bearer YOUR_API_TOKEN"
  ]
}


=== Code Execution Successful ===
```

Practical Example: Consuming a REST API



PHP Online Compiler

Premium Coding Courses by Programiz



Programiz PRO >

main.php

Run

Share

Clear

```
1 <?php
2
3 $response = '{
4     "userId": 1,
5     "id": 1,
6     "title": "Sample Title from Mock API",
7     "body": "This is a fake response body since Programiz blocks
8         external requests."
9 }';
10
11 $data = json_decode($response);
12
13 // Output fields
14 echo 'Title: ' . $data->title . '<br>';
15 echo 'Body: ' . $data->body;
16 ?>
17
```

Output

```
Title: Sample Title from Mock API<br>Body: This is a fake response body
since Programiz blocks external requests.

=== Code Execution Successful ===
```

