# Authentication API Integration

Hanu Deol
Master's Student, Cybersecurity
Vancouver, Canada
hdeol@nyit.edu

Jayesh Zala
Master's Student, Cybersecurity
Vancouver, Canada
jzala03@nyit.edu

Wen Lun Hsu
Master's Student, Cybersecurity
Vancouver, Canada
whsu05@nyit.edu

***Abstract* — Integrating f8th and Duo API functions via RESTful API with PHP language and some test cases. Eventually, we will explain how to integrate Duo Auth API with a customized login page.**

**Keywords — RESTful API, API Integration, Customized Login Page**

## I. Introduction

This industry project is about integrating f8th API and Duo API using RESTful API in PHP. We divided into two groups. One is working on the f8th API, and another group is working on Duo API. Besides this, we had a weekly group meeting in the morning on Wednesday. The primary purpose of the weekly meeting is to know the updated progress of the project.

## II. Potential Issues

Initially, we struggled with providing Duo API Key, Duo Web SDK of its restriction and PHP dependencies issue. When we tried to use the provided Duo API key, the result indicated that the credentials were lack of authentication, authorization and a data header, which was vague in the documentation [1]. However, we also got issues with Duo Web SDK API functionality in Figures 1 and 2. Consequently, we did some research and email Stephen, the Duo Web SDK only can work on the Duo Web SDK platform [2].

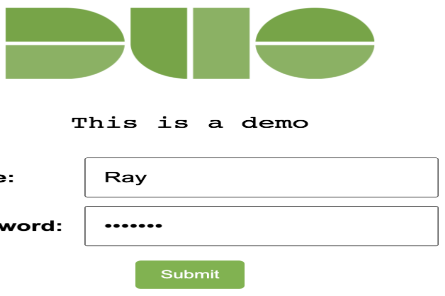
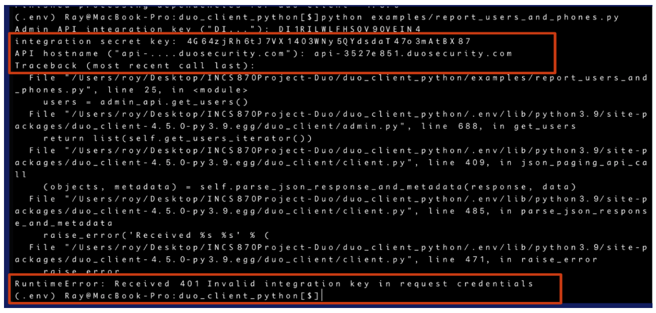


Fig. 1. Duo Web SDK Login Console



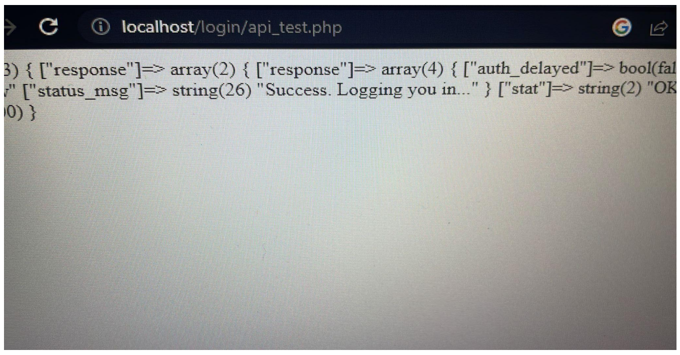
```
(.env) Ray@MacBook-Pro:duo_client_python[$]python examples/report_users_and_phones.py
Admin API integration key ("DI..."): DI1RILWLFHS0V90VEIN4
integration secret key: 4G64zjRh6tJ7VX1403WNy5QYdsdaT47o3mAtBX87
API hostname ("api-....duosecurity.com"): api-3527e851.duosecurity.com
Traceback (most recent call last):
  File "/Users/roy/Desktop/INCS870Project-Duo/duo_client_python/examples/report_users_and
_phones.py", line 25, in <module>
    users = admin_api.get_users()
  File "/Users/roy/Desktop/INCS870Project-Duo/duo_client_python/.env/lib/python3.9/site-p
ackages/duo_client-4.5.0-py3.9.egg/duo_client/admin.py", line 688, in get_users
    return list(self.get_users_iterator())
  File "/Users/roy/Desktop/INCS870Project-Duo/duo_client_python/.env/lib/python3.9/site-p
ackages/duo_client-4.5.0-py3.9.egg/duo_client/client.py", line 409, in json_paging_api_ca
ll
    (objects, metadata) = self.parse_json_response_and_metadata(response, data)
  File "/Users/roy/Desktop/INCS870Project-Duo/duo_client_python/.env/lib/python3.9/site-p
ackages/duo_client-4.5.0-py3.9.egg/duo_client/client.py", line 485, in parse_json_respons
e_and_metadata
    raise_error('Received %s %s' % (
  File "/Users/roy/Desktop/INCS870Project-Duo/duo_client_python/.env/lib/python3.9/site-p
ackages/duo_client-4.5.0-py3.9.egg/duo_client/client.py", line 471, in raise_error
    raise_error
RuntimeError: Received 401 Invalid integration key in request credentials
(.env) Ray@MacBook-Pro:duo_client_python[$]
```

Fig. 2. Duo Web SDK Authentication Issue,

Therefore, since the Duo Web SDK had not worked, we worked on Duo Auth API, which has not been rebuilt from DUO. So, we struggled with integrating Duo Auth API [3] and troubleshooting PHP of its dependencies [4] [5].

## III. Proposed Solutions

The proposed remedies for the potential issues are creating our own integration key and doing some test cases to ensure the integration key is doable.



localhost/login/api_test.php

```
3) { ["response"]=> array(2) { ["response"]=> array(4) { ["auth_delayed"]=> bool(fal
," ["status_msg"]=> string(26) "Success. Logging you in..." } ["stat"]=> string(2) "OK
0) }
```

Fig. 3. Duo Auth API Test Case Result

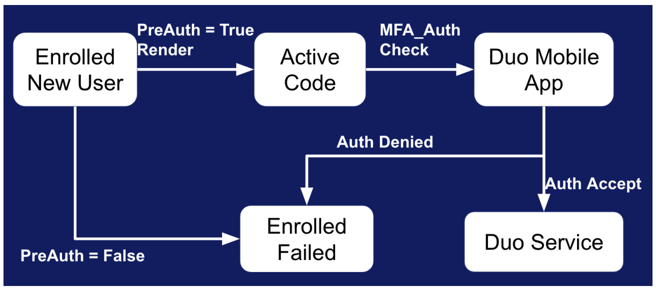


Fig. 4. Duo Auth API Data Flow

As for the test case, we used the Duo Auth API resources [6], which shows the data flow from the application side works fluently in Figure 3.

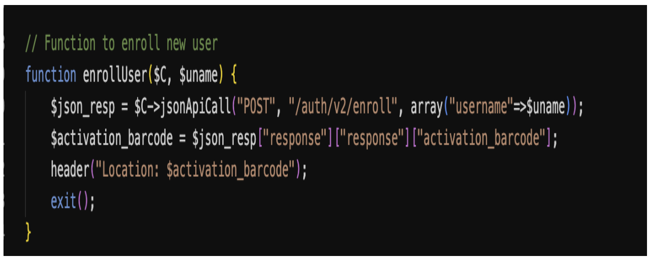
Besides this, we check the integration key, and the data flow works well, but the integration of Duo Auth functionalities still needs to work on because Duo has not been documented. The one possible remedy is to check the /auth/check API endpoint, HTTP basic authentication and data header with some further API testing.

## IV. Methodology of the Project

We utilized RESTful API to integrate each functionality of Duo Auth API [7] [8], and we also followed up on the PHP documentation to achieve the integration [9]. At the same time, some other cases still need to be solved via Postman API testing.

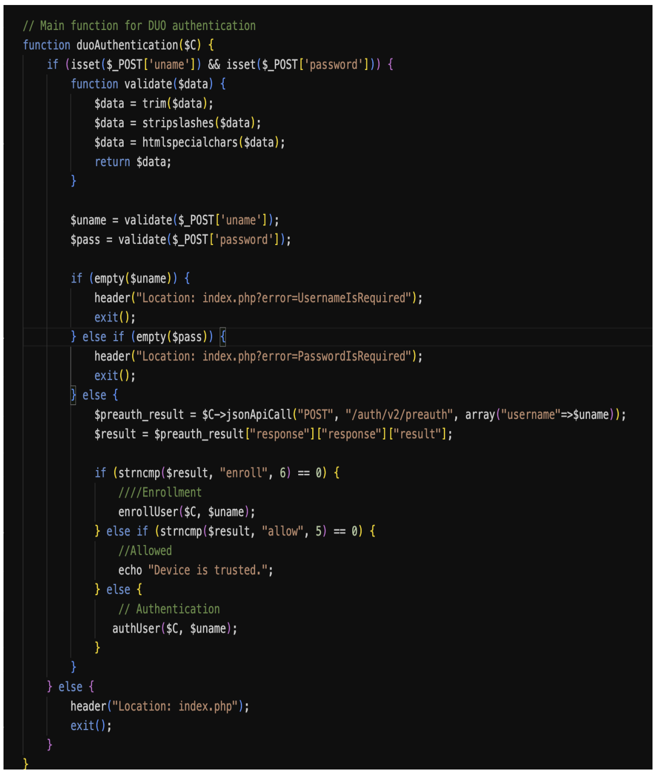
## V. How to Implement Duo Auth API

In Fig. 4, this is the Duo Auth API data flow. The program will check whether the user's username is an existing account. Once we enroll the new user, the system will pop up a QRcode for scanning purposes, then the program will call Duo /auth API function for authentication.



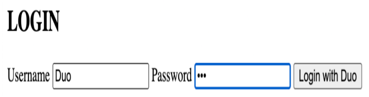
```
// Function to enroll new user
function enrollUser($C, $uname) {
    $json_resp = $C->jsonApiCall("POST", "/auth/v2/enroll", array("username"=>$uname));
    $activation_barcode = $json_resp["response"]["response"]["activation_barcode"];
    header("Location: $activation_barcode");
    exit();
}
```

After registering the new user, the program call /preauth API to check the username in terms of the below code screenshot.



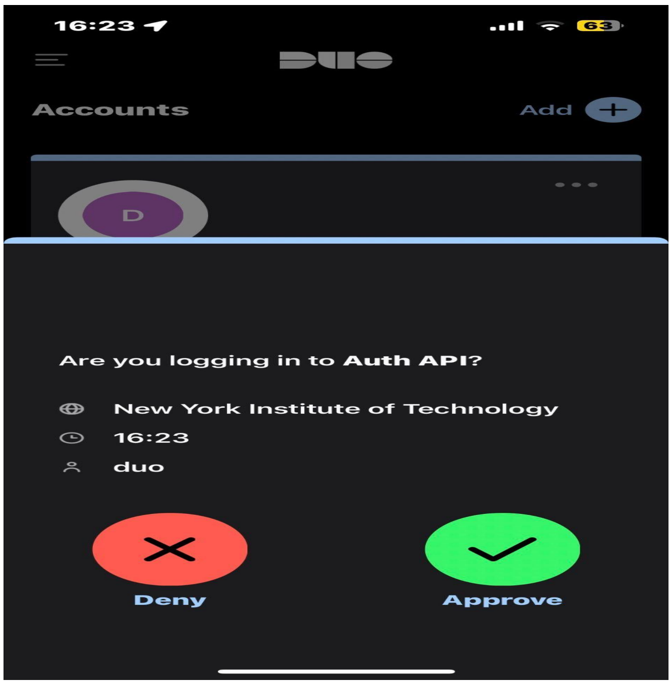
```
// Main function for DUO authentication
function duoAuthentication($C) {
    if (isset($_POST['uname']) && isset($_POST['password'])) {
        function validate($data) {
            $data = trim($data);
            $data = stripslashes($data);
            $data = htmlspecialchars($data);
            return $data;
        }

        $uname = validate($_POST['uname']);
        $pass = validate($_POST['password']);

        if (empty($uname)) {
            header("Location: index.php?error=UsernameIsRequired");
            exit();
        } else if (empty($pass)) {
            header("Location: index.php?error=PasswordIsRequired");
            exit();
        } else {
            $preauth_result = $C->jsonApiCall("POST", "/auth/v2/preauth", array("username"=>$uname));
            $result = $preauth_result["response"]["response"]["result"];

            if (strncmp($result, "enroll", 6) == 0) {
                ////Enrollment
                enrollUser($C, $uname);
            } else if (strncmp($result, "allow", 5) == 0) {
                //Allowed
                echo "Device is trusted.";
            } else {
                // Authentication
               authUser($C, $uname);
            }
        }
    } else {
        header("Location: index.php");
        exit();
    }
}
```

The login page looks like this in the below screenshot, and DUO mobile App pushes notification to divice for verification.



LOGIN

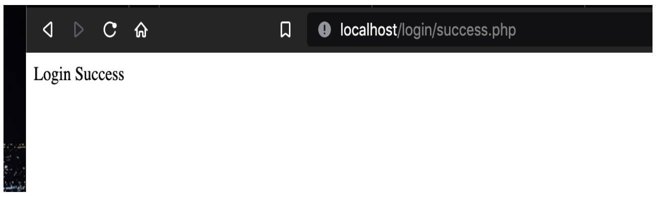
Username Duo Password ••• Login with Duo



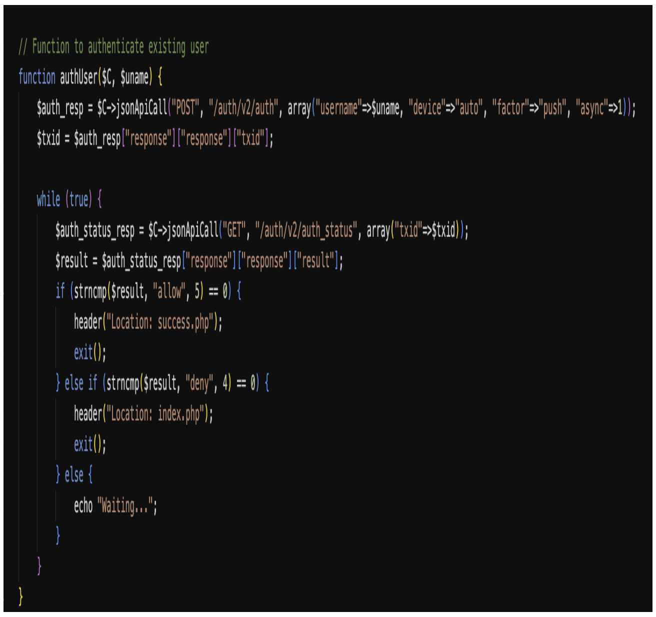
Are you logging in to Auth API?

New York Institute of Technology
16:23
duo

The program shows Login Success after the user approves authentication. Otherwise, the program directs to the Login page to log in again.


localhost/login/success.php

Login Success


```
// Function to authenticate existing user
function authUser($C, $uname) {
    $auth_resp = $C->jsonApiCall("POST", "/auth/v2/auth", array("username"=>$uname, "device"=>"auto", "factor"=>"push", "async"=>1));
    $txid = $auth_resp["response"]["response"]["txid"];

    while (true) {
        $auth_status_resp = $C->jsonApiCall("GET", "/auth/v2/auth_status", array("txid"=>$txid));
        $result = $auth_status_resp["response"]["response"]["result"];
        if (strncmp($result, "allow", 5) == 0) {
            header("Location: success.php");
            exit();
        } else if (strncmp($result, "deny", 4) == 0) {
            header("Location: index.php");
            exit();
        } else {
            echo "Waiting...";
        }
    }
}
```

## VI. Api Testing

### The Scope of API Testing

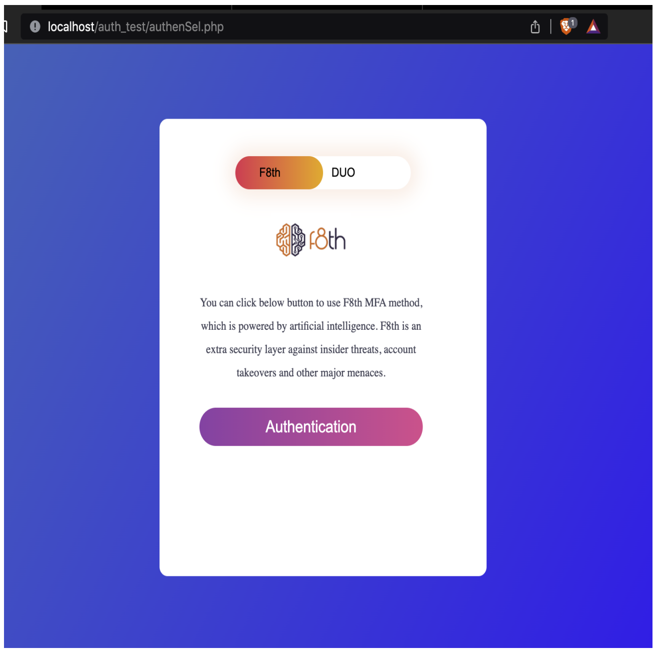
The test we are going to do is only focus on frontend testing instead of backend testing because we do not have any access to the Duo backend service, so it is impossible to parse the Duo Data flow.

The test cases are as below, which are essential to the HTTP Request and Response.

- HTTP Response Status Checking
- HTTP Response Header Checking
- Response Time Checking
- Cookie Testing

### The Testing Methodology

Using Postman for testing F8th SDK and Duo Auth API of HTTP requests, utilizing a GUI, the script uses Chai Assertion Library BDD syntax to perform the test cases.


localhost/auth_test/authenSel.php

You can click below button to use F8th MFA method, which is powered by artificial intelligence. F8th is an extra security layer against insider threats, account takeovers and other major menaces.

### The Objective

***Status Code Testing:***

Sending the HTTP Get Request to the customized Duo login page, and seeing the response status is included with the status set.

***Duo Auth Part:***

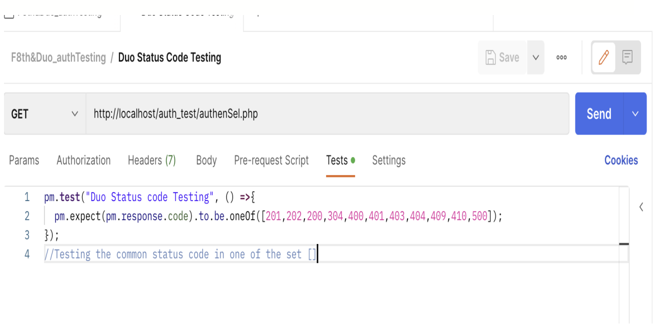
The script is:

pm.test("Duo Status code Testing", () =>{

pm.expect(pm.response.code).to.be.oneOf([201,202,200,304,400,401,403,404,409,410,500]);

});

//Testing the common status code in one of the set []


F8th&Duo_authTesting / Duo Status Code Testing

GET http://localhost/auth_test/authenSel.php

```
pm.test("Duo Status code Testing", () =>{
  pm.expect(pm.response.code).to.be.oneOf([201,202,200,304,400,401,403,404,409,410,500]);
});
//Testing the common status code in one of the set []
```

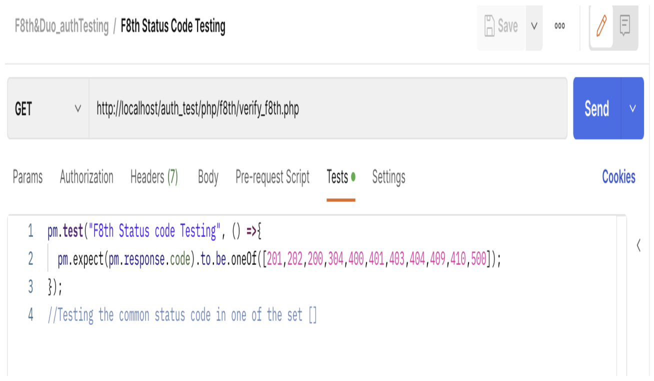
***F8th SDK Part:***

pm.test("F8th Status code Testing", () =>{

pm.expect(pm.response.code).to.be.oneOf([201,202,200,304,400,401,403,404,409,410,500]);

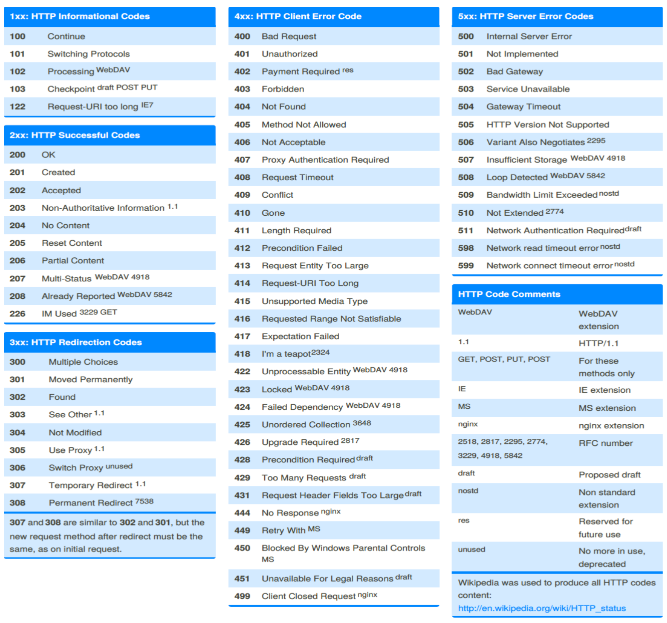
});

//Testing the common status code in one of the set []


F8th&Duo_authTesting / F8th Status Code Testing
Save
GET
http://localhost/auth_test/php/f8th/verify_f8th.php
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies

```
pm.test("F8th Status code Testing", () =>{
  pm.expect(pm.response.code).to.be.oneOf([201,202,200,304,400,401,403,404,409,410,500]);
});
//Testing the common status code in one of the set []
```

The complete picture of the status code can be found in the below cheatsheet.



| 1xx: HTTP Informational Codes | |
|---|---|
| 100 | Continue |
| 101 | Switching Protocols |
| 102 | Processing WebDAV |
| 103 | Checkpoint draft POST PUT |
| 122 | Request-URI too long IE7 |

| 2xx: HTTP Successful Codes | |
|---|---|
| 200 | OK |
| 201 | Created |
| 202 | Accepted |
| 203 | Non-Authoritative Information 1.1 |
| 204 | No Content |
| 205 | Reset Content |
| 206 | Partial Content |
| 207 | Multi-Status WebDAV 4918 |
| 208 | Already Reported WebDAV 5842 |
| 226 | IM Used 3229 GET |

| 3xx: HTTP Redirection Codes | |
|---|---|
| 300 | Multiple Choices |
| 301 | Moved Permanently |
| 302 | Found |
| 303 | See Other 1.1 |
| 304 | Not Modified |
| 305 | Use Proxy 1.1 |
| 306 | Switch Proxy unused |
| 307 | Temporary Redirect 1.1 |
| 308 | Permanent Redirect 7538 |

**307** and **308** are similar to **302** and **301**, but the new request method after redirect must be the same, as on initial request.

| 4xx: HTTP Client Error Code | |
|---|---|
| 400 | Bad Request |
| 401 | Unauthorized |
| 402 | Payment Required res |
| 403 | Forbidden |
| 404 | Not Found |
| 405 | Method Not Allowed |
| 406 | Not Acceptable |
| 407 | Proxy Authentication Required |
| 408 | Request Timeout |
| 409 | Conflict |
| 410 | Gone |
| 411 | Length Required |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 414 | Request-URI Too Long |
| 415 | Unsupported Media Type |
| 416 | Requested Range Not Satisfiable |
| 417 | Expectation Failed |
| 418 | I'm a teapot 2324 |
| 422 | Unprocessable Entity WebDAV 4918 |
| 423 | Locked WebDAV 4918 |
| 424 | Failed Dependency WebDAV 4918 |
| 425 | Unordered Collection 3648 |
| 426 | Upgrade Required 2817 |
| 428 | Precondition Required draft |
| 429 | Too Many Requests draft |
| 431 | Request Header Fields Too Large draft |
| 444 | No Response nginx |
| 449 | Retry With MS |
| 450 | Blocked By Windows Parental Controls MS |
| 451 | Unavailable For Legal Reasons draft |
| 499 | Client Closed Request nginx |

| 5xx: HTTP Server Error Codes | |
|---|---|
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 502 | Bad Gateway |
| 503 | Service Unavailable |
| 504 | Gateway Timeout |
| 505 | HTTP Version Not Supported |
| 506 | Variant Also Negotiates 2295 |
| 507 | Insufficient Storage WebDAV 4918 |
| 508 | Loop Detected WebDAV 5842 |
| 509 | Bandwidth Limit Exceeded nostd |
| 510 | Not Extended 2774 |
| 511 | Network Authentication Required draft |
| 598 | Network read timeout error nostd |
| 599 | Network connect timeout error nostd |

| HTTP Code Comments | |
|---|---|
| WebDAV | WebDAV extension |
| 1.1 | HTTP/1.1 |
| GET, POST, PUT, POST | For these methods only |
| IE | IE extension |
| MS | MS extension |
| nginx | nginx extension |
| 2518, 2817, 2295, 2774, 3229, 4918, 5842 | RFC number |
| draft | Proposed draft |
| nostd | Non standard extension |
| res | Reserved for future use |
| unused | No more in use, deprecated |

Wikipedia was used to produce all HTTP codes content: http://en.wikipedia.org/wiki/HTTP_status

***Header Testing:***

Checking what types of response header the customized login page utilizes. And, add case-insensitive regular expression i flag at the end of the matched string.

The script is:

***Duo Auth Part***

pm.**test**("Duo Verify Content-Type Header", () => {

  pm.response.to.have.header("Content-Type");

  //case-insensitive regular expression with i flag

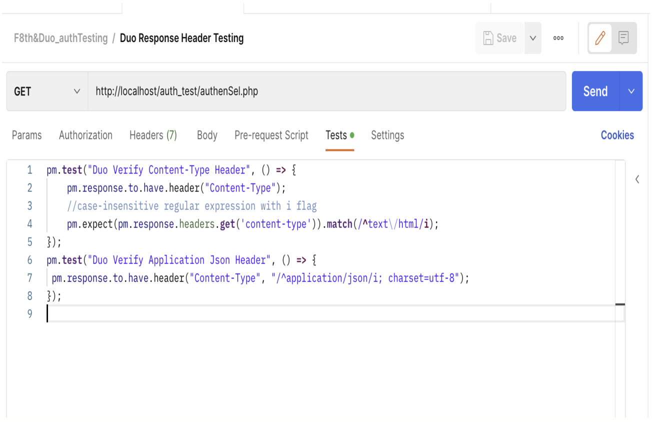
pm.expect(pm.response.headers.get('content-type')).**match**(/^text\/html/**i**);

});

pm.**test**("Duo Verify Application Json Header", () => {

pm.response.to.have.header("Content-Type", "/^application/json/i; charset=utf-8");

});


F8th&Duo_authTesting / Duo Response Header Testing
Save
GET
http://localhost/auth_test/authenSel.php
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies

```
pm.test("Duo Verify Content-Type Header", () => {
    pm.response.to.have.header("Content-Type");
    //case-insensitive regular expression with i flag
    pm.expect(pm.response.headers.get('content-type')).match(/^text\/html/i);
});
pm.test("Duo Verify Application Json Header", () => {
 pm.response.to.have.header("Content-Type", "/^application/json/i; charset=utf-8");
});
```

***F8th SDK Part:***

pm.test("F8th Verify Content-Type Header", () => {

  pm.response.to.have.header("Content-Type");

  //case-insensitive regular expression with i flag

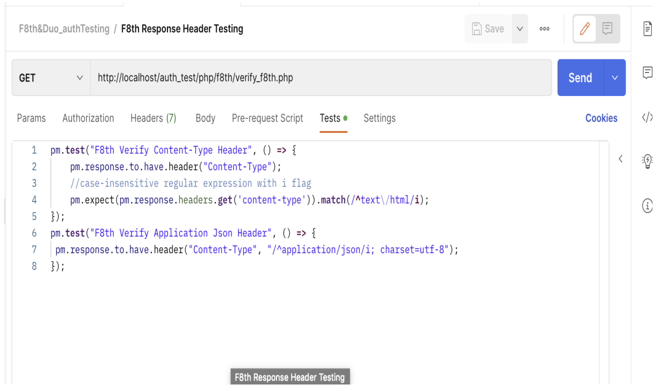
pm.expect(pm.response.headers.get('content-type')).match(/^text\/html/i);

});

pm.test("F8th Verify Application Json Header", () => {

pm.response.to.have.header("Content-Type", "/^application/json/i; charset=utf-8");

});


F8th&Duo_authTesting / F8th Response Header Testing
Save
GET
http://localhost/auth_test/php/f8th/verify_f8th.php
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
F8th Response Header Testing

```
pm.test("F8th Verify Content-Type Header", () => {
    pm.response.to.have.header("Content-Type");
    //case-insensitive regular expression with i flag
    pm.expect(pm.response.headers.get('content-type')).match(/^text\/html/i);
});
pm.test("F8th Verify Application Json Header", () => {
 pm.response.to.have.header("Content-Type", "/^application/json/i; charset=utf-8");
});
```

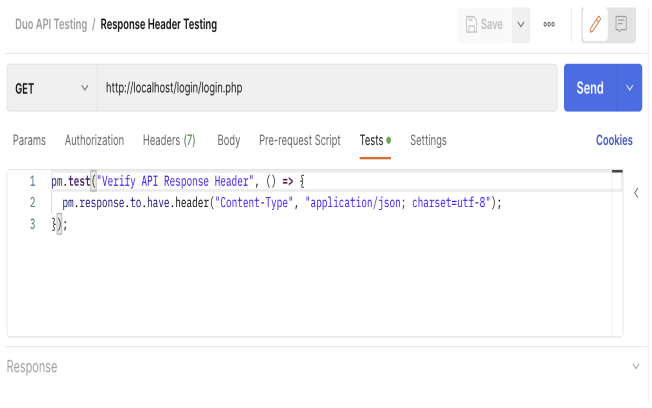
Checking what types of response header the customized login page utilizes.

The script is:

pm.test("Verify API Response Header", () => {

      pm.response.to.have.header("Content-Type", "/^application/json/i; charset=utf-8");

});



Duo API Testing / Response Header Testing

GET http://localhost/login/login.php

```
pm.test("Verify API Response Header", () => {
  pm.response.to.have.header("Content-Type", "application/json; charset=utf-8");
});
```

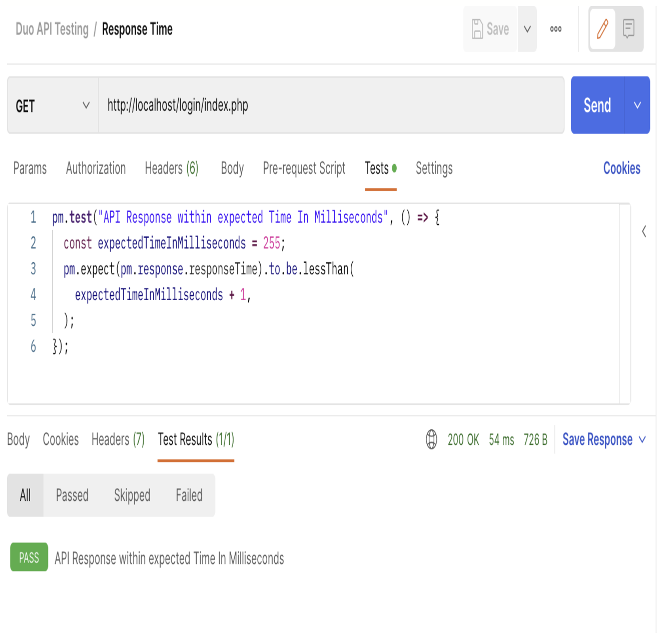
Response

***Response Time Testing:***

The test is to check the HTTP response time that is in the expected time.

The script is:

```
pm.test("API Response within expected Time In Milliseconds", () => {
 const expectedTimeInMilliseconds = 255;
 pm.expect(pm.response.responseTime).to.be.lessThan(
   expectedTimeInMilliseconds + 1,
   );
});
```



Duo API Testing / Response Time

GET http://localhost/login/index.php

```
pm.test("API Response within expected Time In Milliseconds", () => {
  const expectedTimeInMilliseconds = 255;
  pm.expect(pm.response.responseTime).to.be.lessThan(
    expectedTimeInMilliseconds + 1,
  );
});
```

Body Cookies Headers (7) Test Results (1/1) 200 OK 54 ms 726 B Save Response

All Passed Skipped Failed

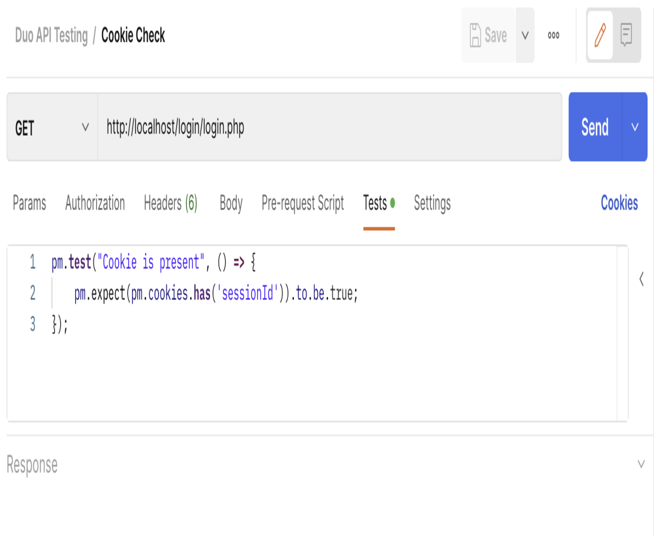
PASS API Response within expected Time In Milliseconds

***Cookie Testing:***

Check the cookie exists if we send the HTTP request to the login page.

The script is:

```
pm.test("Cookie is present", () => {
  pm.expect(pm.cookies.has('sessionId')).to.be.true;
});
```



Duo API Testing / Cookie Check

GET http://localhost/login/login.php

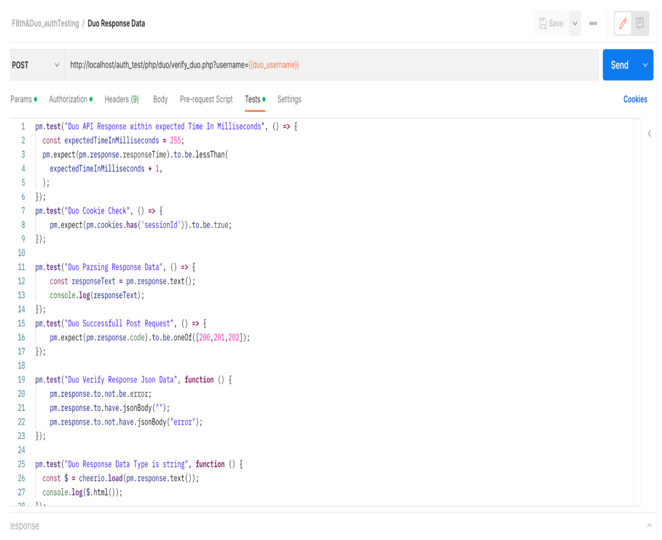
```
pm.test("Cookie is present", () => {
  pm.expect(pm.cookies.has('sessionId')).to.be.true;
});
```

Response

***Response Data Testing:***

***Duo Auth Part:***

```
pm.test("Duo API Response within expected Time In Milliseconds", () => {
 const expectedTimeInMilliseconds = 255;
 pm.expect(pm.response.responseTime).to.be.lessThan(
   expectedTimeInMilliseconds + 1,
 );
});
pm.test("Duo Cookie Check", () => {
  pm.expect(pm.cookies.has('sessionId')).to.be.true;
});

pm.test("Duo Parsing Response Data", () => {
  const responseText = pm.response.text();
  console.log(responseText);
});
pm.test("Duo Successfull Post Request", () => {

pm.expect(pm.response.code).to.be.oneOf([200,201,202]);
});

pm.test("Duo Verify Response Json Data", function () {
  pm.response.to.not.be.error;
  pm.response.to.have.jsonBody("");
  pm.response.to.not.have.jsonBody("error");
});

pm.test("Duo Response Data Type is string", function () {
 const $ = cheerio.load(pm.response.text());
 console.log($.html());
});
```



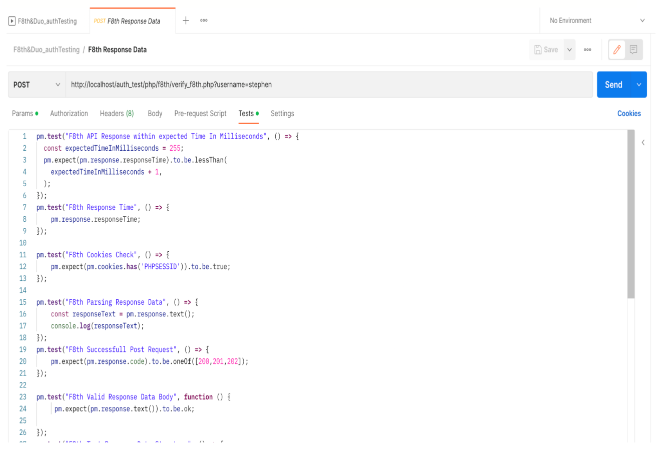
F8th&Duo_authTesting / Duo Response Data

http://localhost/auth_test/php/duo/verify_duo.php?username={{duo_username}}

```
pm.test("Duo API Response within expected Time In Milliseconds", () => {
  const expectedTimeInMilliseconds = 255;
  pm.expect(pm.response.responseTime).to.be.lessThan(
    expectedTimeInMilliseconds + 1,
  );
});
pm.test("Duo Cookie Check", () => {
    pm.expect(pm.cookies.has('sessionId')).to.be.true;
});

pm.test("Duo Parsing Response Data", () => {
    const responseText = pm.response.text();
    console.log(responseText);
});
pm.test("Duo Successfull Post Request", () => {
    pm.expect(pm.response.code).to.be.oneOf([200,201,202]);
});

pm.test("Duo Verify Response Json Data", function () {
    pm.response.to.not.be.error;
    pm.response.to.have.jsonBody("");
    pm.response.to.not.have.jsonBody("error");
});

pm.test("Duo Response Data Type is string", function () {
  const $ = cheerio.load(pm.response.text());
  console.log($.html());
```

***F8th SDK Part:***

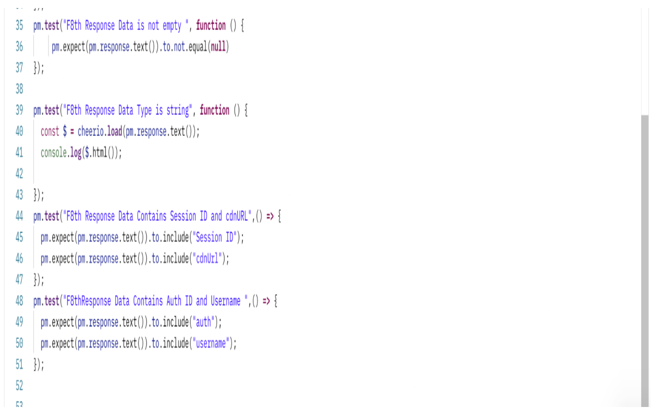
```
pm.test("F8th API Response within expected Time In Milliseconds", () => {
 const expectedTimeInMilliseconds = 255;
 pm.expect(pm.response.responseTime).to.be.lessThan(
   expectedTimeInMilliseconds + 1,
 );
});
pm.test("F8th Response Time", () => {
   pm.response.responseTime;
});

pm.test("F8th Cookies Check", () => {
   pm.expect(pm.cookies.has('PHPSESSID')).to.be.true;
});

pm.test("F8th Parsing Response Data", () => {
   const responseText = pm.response.text();
   console.log(responseText);
});
pm.test("F8th Successfull Post Request", () => {
   pm.expect(pm.response.code).to.be.oneOf([200,201,202]);
});

pm.test("F8th Valid Response Data Body", function () {
    pm.expect(pm.response.text()).to.be.ok;

});
pm.test("F8th Test Response Data Structure", () => {
 const textData = pm.response.text();
 pm.expect(textData).to.be.an("string");
 pm.expect(pm.response.text()).to.include("stephen");
 // check if there is any url is sent
 pm.expect(textData.website).to.be.undefined;
 });
pm.test("F8th Response Data is not empty ", function () {
    pm.expect(pm.response.text()).to.not.equal(null)
});

pm.test("F8th Response Data Type is string", function () {
 const $ = cheerio.load(pm.response.text());
 console.log($.html());
 });
pm.test("F8th Response Data Contains Session ID and cdnURL",() => {
 pm.expect(pm.response.text()).to.include("Session ID");
 pm.expect(pm.response.text()).to.include("cdnUrl");
});
pm.test("F8thResponse Data Contains Auth ID and Username ",() => {
 pm.expect(pm.response.text()).to.include("auth");
 pm.expect(pm.response.text()).to.include("username");
});
```



F8th&Duo_authTesting / F8th Response Data

http://localhost/auth_test/php/f8th/verify_f8th.php?username=stephen

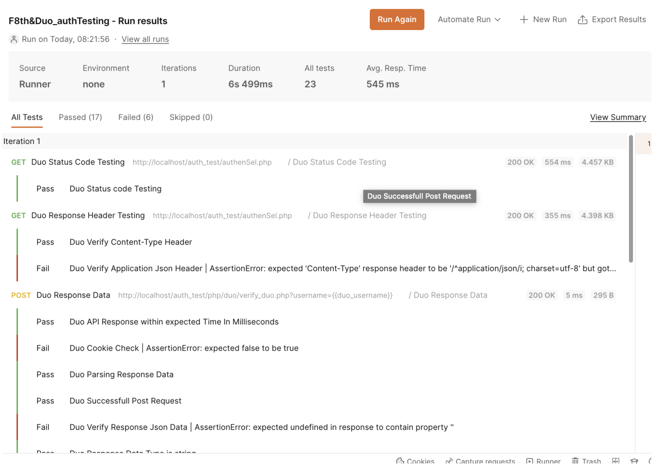
```
pm.test("F8th API Response within expected Time In Milliseconds", () => {
  const expectedTimeInMilliseconds = 255;
  pm.expect(pm.response.responseTime).to.be.lessThan(
    expectedTimeInMilliseconds + 1,
  );
});
pm.test("F8th Response Time", () => {
    pm.response.responseTime;
});

pm.test("F8th Cookies Check", () => {
    pm.expect(pm.cookies.has('PHPSESSID')).to.be.true;
});

pm.test("F8th Parsing Response Data", () => {
    const responseText = pm.response.text();
    console.log(responseText);
});
pm.test("F8th Successfull Post Request", () => {
    pm.expect(pm.response.code).to.be.oneOf([200,201,202]);
});

pm.test("F8th Valid Response Data Body", function () {
     pm.expect(pm.response.text()).to.be.ok;

});
```



```
pm.test("F8th Response Data is not empty ", function () {
     pm.expect(pm.response.text()).to.not.equal(null)
});

pm.test("F8th Response Data Type is string", function () {
  const $ = cheerio.load(pm.response.text());
  console.log($.html());

});
pm.test("F8th Response Data Contains Session ID and cdnURL",() => {
  pm.expect(pm.response.text()).to.include("Session ID");
  pm.expect(pm.response.text()).to.include("cdnUrl");
});
pm.test("F8thResponse Data Contains Auth ID and Username ",() => {
  pm.expect(pm.response.text()).to.include("auth");
  pm.expect(pm.response.text()).to.include("username");
});
```

***The complete picture of Testing results:***

If each test case is put together, run the collection with the Postman plugin. Some test cases failed because of the Response header and cookie existence.

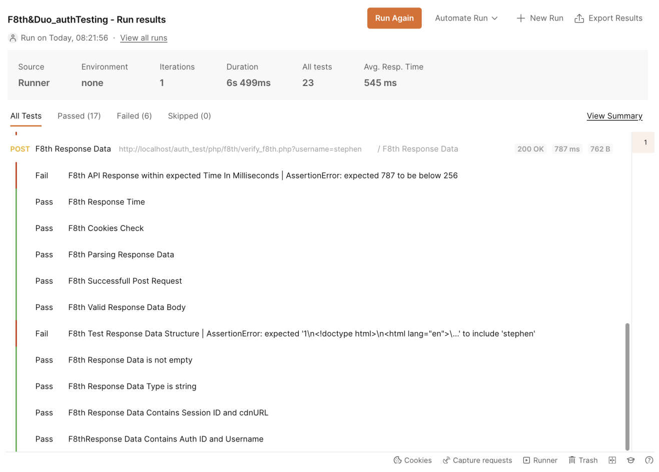


F8th&Duo_authTesting - Run results
Run on Today, 08:21:56 · View all runs

| Source | Environment | Iterations | Duration | All tests | Avg. Resp. Time |
|---|---|---|---|---|---|
| Runner | none | 1 | 6s 499ms | 23 | 545 ms |

All Tests Passed (17) Failed (6) Skipped (0)

```
Iteration 1
GET  Duo Status Code Testing  http://localhost/auth_test/authenSel.php  / Duo Status Code Testing  200 OK  554 ms  4.457 KB
     Pass   Duo Status code Testing
GET  Duo Response Header Testing  http://localhost/auth_test/authenSel.php  / Duo Response Header Testing  200 OK  355 ms  4.398 KB
     Pass   Duo Verify Content-Type Header
     Fail   Duo Verify Application Json Header | AssertionError: expected 'Content-Type' response header to be '/^application/json/i; charset=utf-8' but got...
POST Duo Response Data  http://localhost/auth_test/php/duo/verify_duo.php?username={{duo_username}}  / Duo Response Data  200 OK  5 ms  295 B
     Pass   Duo API Response within expected Time In Milliseconds
     Fail   Duo Cookie Check | AssertionError: expected false to be true
     Pass   Duo Parsing Response Data
     Pass   Duo Successfull Post Request
     Fail   Duo Verify Response Json Data | AssertionError: expected undefined in response to contain property ''
```



F8th&Duo_authTesting - Run results
Run on Today, 08:21:56 · View all runs

| Source | Environment | Iterations | Duration | All tests | Avg. Resp. Time |
|---|---|---|---|---|---|
| Runner | none | 1 | 6s 499ms | 23 | 545 ms |

All Tests Passed (17) Failed (6) Skipped (0)

```
POST F8th Response Data  http://localhost/auth_test/php/f8th/verify_f8th.php?username=stephen  / F8th Response Data  200 OK  787 ms  762 B
     Fail   F8th API Response within expected Time In Milliseconds | AssertionError: expected 787 to be below 256
     Pass   F8th Response Time
     Pass   F8th Cookies Check
     Pass   F8th Parsing Response Data
     Pass   F8th Successfull Post Request
     Pass   F8th Valid Response Data Body
     Fail   F8th Test Response Data Structure | AssertionError: expected '1\n<!doctype html>\n<html lang="en">\...' to include 'stephen'
     Pass   F8th Response Data is not empty
     Pass   F8th Response Data Type is string
     Pass   F8th Response Data Contains Session ID and cdnURL
     Pass   F8thResponse Data Contains Auth ID and Username
```

## VII. Conclusion

To summarize this industry project, it is almost finished, we acknowledge how the RESTful API works and its basic knowledge. The Auth API has various functions to check the users' credentials before trafficking the data to the server for authorization. The API integration is a way to connect multiple applications, allowing data cross-domain.

Most importantly, we have obtained essential knowledge of the Duo authentication data flow with RESTful API using PHP and some critical mindset to solve PHP of its dependencies issues.

## REFERENCES

[1] "Duo web V4 SDK (software development kit)," *Duo Security*. [Online]. Available: https://duo.com/docs/duoweb. [Accessed: 16-Nov-2022].

[2] Duo Security, "Duo_universal_php/example at main · duo security/duo_universal_php," *GitHub*. [Online]. Available: https://github.com/duosecurity/duo_universal_php/tree/main/example. [Accessed: 16-Nov-2022].

[3] "Duo Auth Api," *Duo Security*. [Online]. Available: https://duo.com/docs/authapi. [Accessed: 18-Nov-2022].

[4] "PHP Warning: PHP Startup: Unable to load Dynamic Library," *Stack Overflow*, 01-May-1958. [Online]. Available: https://stackoverflow.com/questions/5282264/php-warning-php-startup-unable-to-load-dynamic-library. [Accessed: 18-Nov-2022].

[5] Phpredis, "Unable to load Dynamic Library '/USR/local/PHP/LIB/PHP/Extensions/no-debug-ZTS-20100525/redis.so' · issue #271 · phpredis/phpredis," *GitHub*. [Online]. Available: https://github.com/phpredis/phpredis/issues/271. [Accessed: 21-Nov-2022].

[6] (PHP extension) duo AUTH API - auth," *PHP Extension Duo Auth API - Auth*. [Online]. Available: https://www.example-code.com/phpext/duo_auth_mfa_auth.asp. [Accessed: 18-Nov-2022].

[7] "Tutorial - integrate basic authentication in a REST API using PHP," *YouTube*, 06-Aug-2019. [Online]. Available: https://www.youtube.com/watch?v=DRFdftHS5wc. [Accessed: 21-Nov-2022].

[8] "PHP login system with source code," *StarTutorial*. [Online]. Available: https://startutorial.com/view/php-login-system-with-source-code. [Accessed: 21-Nov-2022].