

# Phát Triển Phần Mềm Mã Nguồn Mở

## ADVANCED TOPICS

NGUYỄN HẢI TRIỀU

Bộ môn Kỹ thuật phần mềm  
Khoa Công nghệ thông tin  
ĐH Nha Trang

# Mục lục

- 1 Password Hashing Functions
- 2 JSON
- 3 RESTful API

# 1 Password Hashing Functions

## 2 JSON

## 3 RESTful API

## Sign up

Your Name


Your Email

Password

Repeat your password

☐ I agree all statements in [Terms of service](#)

Have already an account? [Login here](#)



**Hình 1:** Thiết lập form đăng kí như hình trên, thông tin đăng kí được lưu lại trong database.

Sign in with



Or

Enter a valid email address

Email address

Enter password

Password

☐ Remember me

[Forgot password?](#)

Login

Don't have an account? [Register](#)

**Hình 2:** Nếu đăng kí thành công, người dùng có thể đăng nhập vào form tài khoản vừa đăng kí.

- As more secure web services **avoid storing passwords in plain text format**
- Create password hashes using `password_hash()` function.

```
1 $options = [  
2   'cost' => 12,  
3 ];  
4 $hashedPassword = password_hash($plaintextPassword,  
5 PASSWORD_DEFAULT, $options);
```

- `PASSWORD_DEFAULT` contains the value as `PASSWORD_BCRYPT` using bcrypt standard.
- In PHP Version  $\geq 7.0$ , random salt will be generated by `password_hash()` for each password hashed

# password\_verify()

is the built-in function provided (as of PHP  $\geq 5.5$ ) to verify the validity of a password against a known hash.

```
1  <?php
2  if (password_verify($plaintextPassword, $hashedPassword)) {
3  echo 'Valid Password';
4  }
5  else {
6  echo 'Invalid Password.';
7  }
8  ?>
```

- 1 Password Hashing Functions
- 2 **JSON**
- 3 RESTful API



The `json_decode()` function takes a JSON-encoded string as its first parameter and parses it into a PHP variable. It will return an object of `stdClass` if the top level item in the JSON object is a dictionary or an indexed array if the JSON object is an array.

```
1 // Returns an object (The top level item in the JSON string is a
   JSON dictionary)
2 $json_string = '{"name": "Jeff", "age": 20, "active": true, "
   colors": ["red", "blue"]}';
3 $object = json_decode($json_string);
4 printf('Hello %s, You are %s years old.', $object->name, $object
   ->age);
5 #> Hello Jeff, You are 20 years old.
6 // Returns an array (The top level item in the JSON string is a
   JSON array)
7 $json_string = '["Jeff", 20, true, ["red", "blue"]]';
8 $array = json_decode($json_string);
9 printf('Hello %s, You are %s years old.', $array[0], $array[1]);
```

Use `var_dump()` to view the types and values of each property on the object we decoded above.

```
1  var_dump($object);
2  //class stdClass#2 (4) {
3  //  ["name"] => string(4) "Jeff"
4  //  ["age"] => int(20)
5  //  ["active"] => bool(true)
6  //  ["colors"] =>
7  //    array(2) {
8  //      [0] => string(3) "red"
9  //      [1] => string(4) "blue"
10 //    }
```

To return an associative array for JSON objects instead of returning an object, pass **true** as the second parameter to `json_decode()`.

```
1  $array = json_decode($json_string, true); // Note the second
    parameter
2  var_dump($array);
3  //array(4) {
4  //      ["name"] => string(4) "Jeff"
5  //      ["age"]  => int(20)
6  //      ["active"] => bool(true)
7  //      ["colors"] =>
8  //array(2) {
9  //          [0] => string(3) "red"
10 //          [1] => string(4) "blue"
11 //}
12 //}
```

Note that `json_decode()` will return `NULL` if the string cannot be converted.

```
1 <?php
2 $json = "{ 'name': 'Jeff', 'age': 20 }" ; // invalid json
3 $person = json_decode($json);
4 var_dump($person->name); // Notice: Trying to get property of non
    -object: returns null
5 echo json_last_error(); # 4 (JSON_ERROR_SYNTAX)
6 echo json_last_error_msg(); # unexpected character
7 ?>
```

The `json_encode` function will **convert a PHP array to a JSON-encoded string**. It returns a JSON-encoded string on success or **FALSE** on failure.

```
1 $array = [  
2     'name' => 'Jeff',  
3     'age' => 20,  
4     'active' => true,  
5     'colors' => ['red', 'blue'],  
6     'values' => [0=>'foo', 3=>'bar'],  
7 ];  
8 echo json_encode($array);  
9 //{"name":"Jeff","age":20,"active":true,"colors":["red","blue"],"values":{"0":"foo","3":"bar"}}
```

Forces the creation of an object instead of an array

```
1 $array = ['Joel', 23, true, ['red', 'blue']];  
2 echo json_encode($array, JSON_FORCE_OBJECT);  
3 //{"0":"Joel","1":23,"2":true,"3":{"0":"red","1":"blue"}}
```

## Makes the JSON *easily readable*

```
1 $array = ['a' => 1, 'b' => 2, 'c' => 3, 'd' => 4];
2 echo json_encode($array);
3 echo json_encode($array, JSON_PRETTY_PRINT);
4 //{"a":1,"b":2,"c":3,"d":4}
5 //{
6     // "a": 1,
7     // "b": 2,
8     // "c": 3,
9     // "d": 4
10 }
```

- 1 Password Hashing Functions
- 2 JSON
- 3 RESTful API**

- ➊ **A**pplication **P**rogramming **I**nterface is a system that contains a set of rules or protocols or tools which help in providing interaction between two applications or software by standard data access.
- ➋ APIs get requests and return the result in the programmer's software system.
- ➌ Prerequisites: PHP and PHP [cURL library](#)



# cURL

**cURL (GET Requests)** is a tool for **transferring data** with URL syntax. It support HTTP, FTP, SCP and many others(curl >= 7.19.4). Remember, you need to install and enable the cURL extension to use it.

```
1 // a little script check is the cURL extension loaded or not
2 if(!extension_loaded("curl")) {
3     die("cURL extension not loaded! Quit Now.");
4 }
5 // Actual script start
6 // create a new cURL resource
7 // $curl is the handle of the resource
8 $curl = curl_init();
9 // set the URL and other options
10 curl_setopt($curl, CURLOPT_URL, "http://www.example.com");
11 // execute and pass the result to browser
12 curl_exec($curl);
13 // close the cURL resource
14 curl_close($curl);
```

# cURL

If you want to mimic **HTML form POST action**, you can use cURL.

```
1 // a little script check is the cURL extension loaded or not
2 if(!extension_loaded("curl")) {
3     die("cURL extension not loaded! Quit Now.");
4 }
5 // Actual script start
6 // create a new cURL resource
7 // $curl is the handle of the resource
8 $curl = curl_init();
9 // set the URL and other options
10 curl_setopt($curl, CURLOPT_URL, "http://www.example.com");
11 // execute and pass the result to browser
12 curl_exec($curl);
13 // close the cURL resource
14 curl_close($curl);
```

**Types of Web APIs** (Web APIs are those which are accessible over the internet):

- ❶ Open APIs: These APIs are publicly available as there is no restrictions.
- ❷ Partner APIs: The user needs license and special rights to access this type of APIs.
- ❸ Private APIs: Owned by companies for internal systems.
- ❹ Composite APIs: Its a combination of data and service APIs for speeding up the execution process.

**Types of Web Service APIs** (Web service APIs are platform-independent methods that are accessed by a network connection):

- ① **SOAP**: Simple Object Access Protocol which uses web service definition language or XML for data transfer. It is very robust. These are used in integrating APIs.
- ② **JSON-RPC**: For data transfer it uses JSON.
- ③ **REST**: The set of rules includes some standard architectural principles for data exchange. For making a request, it uses HTTP methods of getting, PUT, POST, PATCH, DELETE for all CRUD operations.

## REST output is in the form of JSON:

- ➊ GET: Read or retrieves information.
- ➋ POST: Creates new record.
- ➌ PUT: Update a record.
- ➍ DELETE: Deletes a record

**GET:** Fetch information or data collection. For example product details from a table.

```
1 $returnData = callAPI('GET', 'https://api.geeksforgeeks.com/  
  url_for_get/' . $user['user']['buyer_id'], false); $response =  
  json_decode($returnData, true); $errors = $response['response  
  ']['errors'];  
2 $data = $response['response']['data'][0];
```

**POST:** Adds or creates new information such as feedback or reviews of some restaurant.

```
1 $arrayOfData = array( "buyer" => $user['user']['buyer_id'], "
    payment" => array( "accountNumber" => $this->request->data['
    accountNumber'], "routing" => $this->request->data['routing'
    ], "method" => $this->request->data['method'] ), );
2 $apiCall = APIcall('POST', 'https://api.geeksforgeeks.com/
    url_for_post/', json_encode($data_array));
3 $response = json_decode($apiCall, true); $errors = $response['
    response']['errors'];
4 $data = $response['response']['data'][0];
```

## Bài tập

Sử dụng API: google-translate1 từ trang

`https://rapidapi.com/googlecloud/api/google-translate1`,  
Method: POST. Hãy tạo trang chứa form dịch ngôn ngữ tự động có  
chức năng cơ bản như google translate như hình 3. Biết rằng

```

1 <?php
2 //https://rapidapi.com/googlecloud/api/google-translate1
3 $curl = curl_init();
4 curl_setopt_array($curl, [
5     CURLOPT_URL => "https://google-translate1.p.rapidapi.com/
        language/translate/v2",
6     CURLOPT_RETURNTRANSFER => true,
7     CURLOPT_FOLLOWLOCATION => true,
8     CURLOPT_ENCODING => "",
9     CURLOPT_MAXREDIRS => 10,
10    CURLOPT_TIMEOUT => 30,
11    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
12    CURLOPT_CUSTOMREQUEST => "POST",
13    CURLOPT_POSTFIELDS => "q=Hello%2C%20guy!&target=vi&source=en"
14    ,
15    CURLOPT_HTTPHEADER => [

```



## Ví dụ đọc API bằng PHP

Sử dụng API: google-translate1 từ trang `\url{https://rapidapi.com/googlecloud/api/google-translate1}`,  
Method: GET để tạo ra ứng dụng dịch ngôn ngữ

English ▼

Welcome to Nha Trang University

Translate

Russian ▼

You have exceeded the MONTHLY quota for Characters on your current plan, BASIC. Upgrade your plan at <https://rapidapi.com/googlecloud/api/google-translate1>

**Hình 3:** Kết quả thiết kế form sử dụng GET google-translate1 api

## Reference

- [1] [PHP Notes for Professionals](https://goalkicker.com/PHPBook/),  
<https://goalkicker.com/PHPBook/>
- [2] [How to add API function to a simple PHP Page ?](https://www.geeksforgeeks.org/how-to-add-api-function-to-a-simple-php-page/),  
[https://www.geeksforgeeks.org/  
how-to-add-api-function-to-a-simple-php-page/](https://www.geeksforgeeks.org/how-to-add-api-function-to-a-simple-php-page/)