WEB PROGRAMMING AND APPLICATIONS (503073)

WEEK 9

Prepared by Mai Van Manh

Exercise: In this lesson, you are required to create a restful web service to manage product information. Information that needs to be saved for a product includes: id, product name, price and description. The data needs to be stored in the MySQL database system. You don't need to create a database and table from scratch, use the attached sql file to initialize the database and table. Please import it into MySQL via phpMyAdmin.

Specifically, your web service needs to provide the following functionality:

1. Add a new product (add_product.php)
2. Read product list (get_products.php)
3. Read detailed information of a product by its id (get_product.php)
4. Update product information (update_product.php)
5. Delete a product based on id (delete_product.php)

Below is a detailed description of each function.

File Name	Support Method	Params	Sample Body
add_product.php	POST		{"name":"iPhone","price":1099,"desc":"Like new"}
get_products.php	GET		
get_product.php	GET	id (int)	
update_product.php	PUT	id (int)	{"name":"iPhone","price":1099,"desc":"Like new"}
delete_product.php	DELETE	id (int)	

For examples:

- To add a new product: you need to send an http POST request to the url `add_product.php` with the request body of type json and the value similar to: `{"name":"iPhone","price":1099,"desc" :"Like new"}`.
- To delete a product with id 123: you need to send an http request to `add_product.php?id=123` using the http DELETE method. In this case, you don't need to send any content in the request body.

Input validation:

- Input information needs to be carefully checked. If the input is invalid, the api will return the appropriate error code and message.
- The http method is also considered as input and also needs to be checked carefully. For example, when deleting a product, we only accept http delete request, all other http methods are not accepted.
- For example, when the client sends a request with insufficient information or contains invalid information, we will return the following content: `{"code":1,"message":"Invalid parameters"}`
- For example, when the client sends a request with an invalid method, we will return the following content: `{"code":2,"message":"GET method is not supported"}`

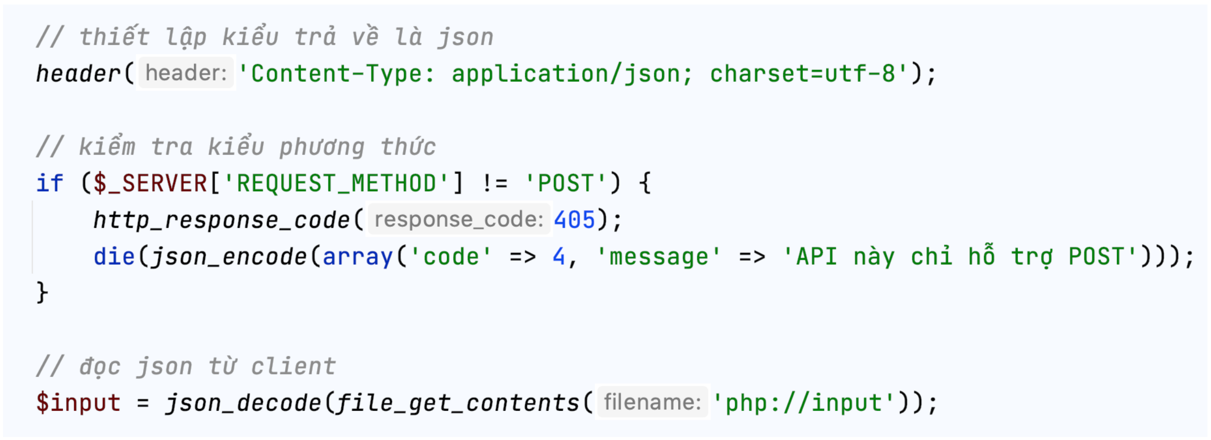
To test the api you can use http client tools such as Postman.

Hint:

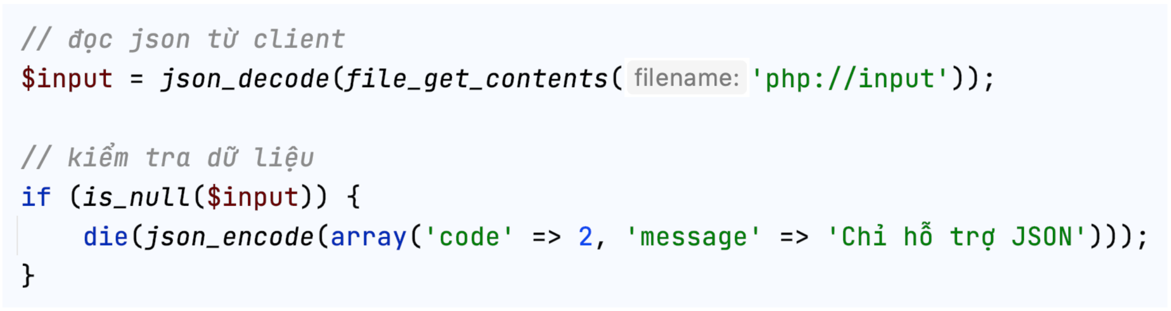
- For restful api, each request is usually tied to a specific http method, for example http put method is often used for updating information. To check with which http method the request was sent, use the `$_SERVER['REQUEST_METHOD']` variable.
- As requested, the information sent from the client is of json type (*application/json* to be exact) so on the php side we cannot read them using the `$_GET` or `$_POST`

superglobals. Instead we have to read raw data from request body using `file_get_contents('php://input')` function and convert it to json object using `json_decode()` function.

- The output from `json_decode()` is an object, it can be null so you need to use functions like `is_null()` to validate before using it. When the object is not null, the properties inside it can still be null, then you can use the `property_exists()` and `empty()`, `isset()` functions to validate the properties.
- The opposite of `json_decode()` is `json_encode()`. You use this function to convert the array containing the returned data into json to respond to the client.
- When working with the restful api, the Content-type and response code of the http response also need to be set up properly using the `header()` and `http_response_code()` functions.



```php
// thiết lập kiểu trả về là json
header( header: 'Content-Type: application/json; charset=utf-8');

// kiểm tra kiểu phương thức
if ($_SERVER['REQUEST_METHOD'] != 'POST') {
    http_response_code( response_code: 405);
    die(json_encode(array('code' => 4, 'message' => 'API này chỉ hỗ trợ POST')));
}

// đọc json từ client
$input = json_decode(file_get_contents( filename: 'php://input'));
```



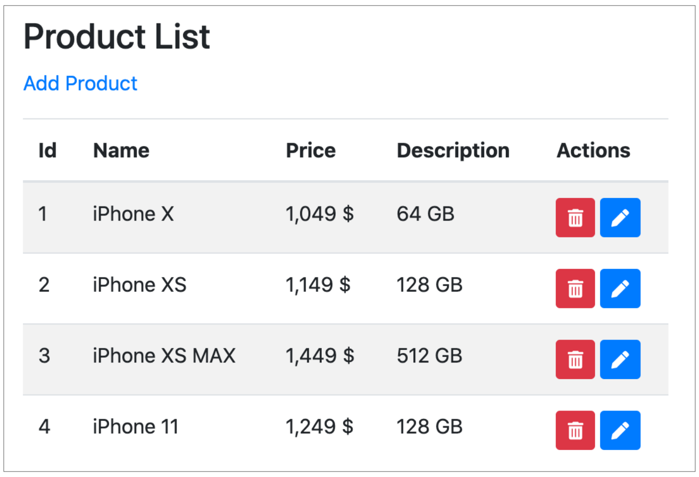
```php
// đọc json từ client
$input = json_decode(file_get_contents( filename: 'php://input'));

// kiểm tra dữ liệu
if (is_null($input)) {
    die(json_encode(array('code' => 2, 'message' => 'Chỉ hỗ trợ JSON')));
}
```

```
// kiểm tra dữ liệu
if (!property_exists($input, property: 'name') ||
    !property_exists($input, property: 'price') ||
    !property_exists($input, property: 'description')) {

    http_response_code( response_code: 400);
    die(json_encode(array('code' => 1, 'message' => 'Thiếu thông tin đầu vào')));
}

// kiểm tra dữ liệu
if (empty($input->name) ||
    empty($input->price) ||
    empty($input->description)) {
    die(json_encode(array('code' => 1, 'message' => 'Thông tin không hợp lệ')));
}
```

```
// TODO: truy vấn database để chèn dữ liệu vào bảng

die(json_encode(array('code' => 0, 'message' => 'Chèn thành công')));
```

Homework:

- Create a client side website using html and javascript language to consume restful web service. This website makes ajax calls to the api endpoints on the server to perform the following functions: display product list, add new products, update products, and delete products.

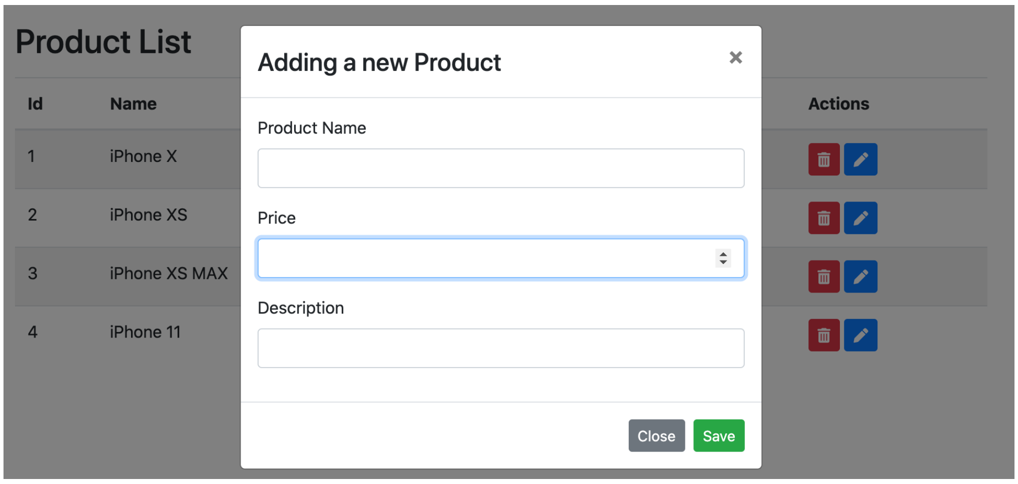


Product List

Add Product

Id	Name	Price	Description	Actions
1	iPhone X	1,049 $	64 GB	🗑 ✎
2	iPhone XS	1,149 $	128 GB	🗑 ✎
3	iPhone XS MAX	1,449 $	512 GB	🗑 ✎
4	iPhone 11	1,249 $	128 GB	🗑 ✎

Sample image of home page showing list of products read from restful web service using ajax



Product List

Adding a new Product

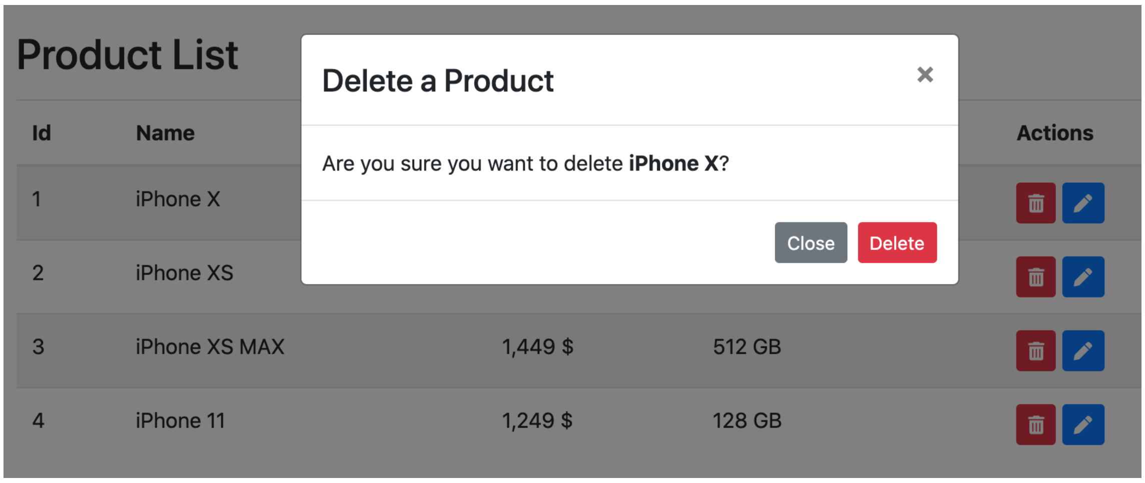
Product Name

Price

Description

Id	Name	Actions
1	iPhone X	
2	iPhone XS	
3	iPhone XS MAX	
4	iPhone 11	

Sample image of a dialog when adding a new product



Product List

Delete a Product

Are you sure you want to delete **iPhone X**?

Id	Name			Actions
1	iPhone X			
2	iPhone XS			
3	iPhone XS MAX	1,449 $	512 GB	
4	iPhone 11	1,249 $	128 GB	

Sample image of a confirmation dialog when deleting a product