



Restful trong PHP

Giảng viên: **Bùi Quang Đăng**



Giới thiệu về JSON

WebService, Restful, API là gì ?

Làm việc với Restful trong PHP

Practices



Làm việc với Restful

www.stanford.com.vn

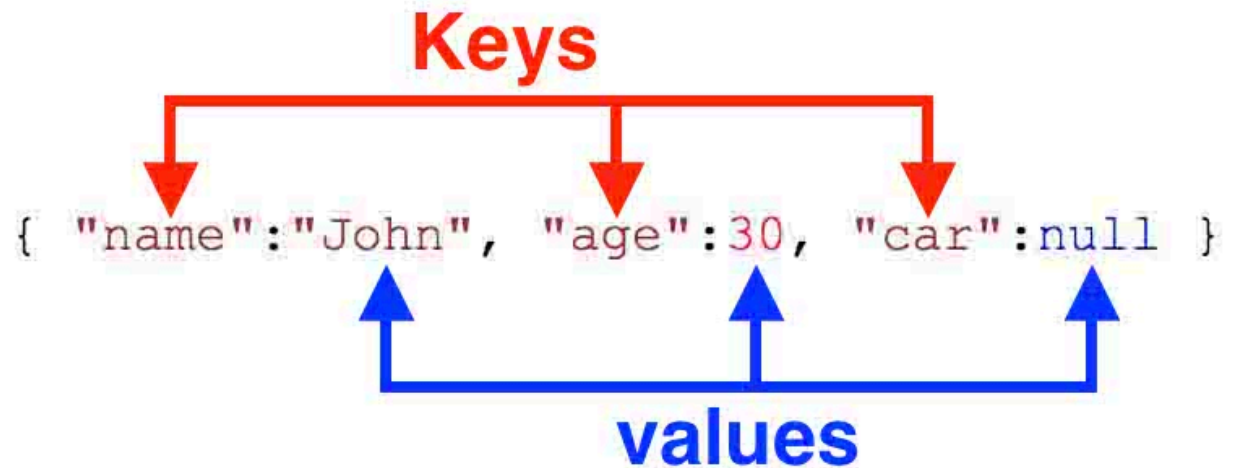
LÀM VIỆC VỚI JSON

❖ JSON là gì ?

- **JSON** là viết tắt của **JavaScript Object Notation**, là một kiểu định dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được.
- JSON là một tiêu chuẩn mở để trao đổi dữ liệu trên web.

❖ JSON là gì ?

{JSON}
JavaScript Object Notation



❖ JSON là gì ?

XML

```
<empinfo>
  <employees>
    <employee>
      <name>James Kirk</name>
      <age>40</age>
    </employee>
    <employee>
      <name>Jean-Luc Picard</name>
      <age>45</age>
    </employee>
    <employee>
      <name>Wesley Crusher</name>
      <age>27</age>
    </employee>
  </employees>
</empinfo>
```

JSON

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "James Kirk",
        "age" : 40,
      },
      {
        "name" : "Jean-Luc Picard",
        "age" : 45,
      },
      {
        "name" : "Wesley Crusher",
        "age" : 27,
      }
    ]
  }
}
```

❖ Định nghĩa JSON

- Định dạng JSON sử dụng các cặp key – value để dữ liệu sử dụng.
- Nó hỗ trợ các cấu trúc dữ liệu như đối tượng và mảng.

❖ Định nghĩa JSON

```
{  
  "name" : "Stanford – Dạy lập trình",  
  "title" : "Chuyên gia đào tạo lập trình",  
  "description" : "Thực chiến qua dự án cùng chuyên gia giỏi,  
  giàu kinh nghiệm"  
}
```


❖ Cấu trúc chuỗi JSON

■ Object là gì?

- Object trong Json được thể hiện bằng dấu ngoặc nhọn {}. Khái niệm Object trong Json cũng khá tương đồng với Object trong Javascript.

❖ Cấu trúc chuỗi JSON

■ Object là gì?

- **Key:** phải luôn nằm trong dấu ngoặc kép, không được phép là biến số.
- **Value:** Chỉ cho phép các kiểu dữ liệu cơ bản: numbers, String, Booleans, arrays, objects, null. Không cho phép function, date, undefined.
- Không cho phép dãy phẩy cuối cùng như Object trong Javascript.

❖ Cấu trúc JSON

■ Object

```
var objSV = {  
  "name" : "Donal Trump",  
  "age" : "74",  
  "gender" : "male"  
}
```

❖ Cấu trúc JSON

■ Object

```
var lstNhanVien = [{  
    "name" : "Donal Trump",  
    "age" : "74",  
    "gender" : "male"  
},  
{  
    "name" : "Obama",  
    "age" : "59",  
    "gender" : "male"  
}  
]
```



PHP for Developer

www.stanford.com.vn

WebService, Restful, API

@stanford.com.vn

❖ Webservice là gì ?

- Web Service là 1 ứng dụng client server giao tiếp với nhau qua giao tiếp HTTP/HTTPS.
- Web Service cung cấp 1 chuẩn giao tiếp chung giữa các nền tảng, framework, ứng dụng...
- Khi sử dụng web service ta chỉ quan tâm đầu vào là gì và đầu ra là gì mà không cần quan tâm tới hệ điều hành, ngôn ngữ lập trình... của ứng dụng gửi yêu cầu là gì.

❖ Webservice là gì ?

- **Ví dụ:** Ứng dụng A cung cấp chức năng lấy thông tin của 1 user qua url là “stanford.com.vn/user/id”:
 - Đầu vào là id của user, đầu ra là thông tin của user đó.
 - Bây giờ ta có một ứng dụng trên mobile, trên web, trên desktop đều muốn lấy thông tin user thì chỉ cần gửi request tới ứng dụng A kèm theo id của user muốn lấy thông tin.
 - Ứng dụng A sẽ nhận id gửi tới và trả về kết quả mà không phân biệt request được gửi từ đâu, viết bằng ngôn ngữ gì.

❖ Webservice là gì ?

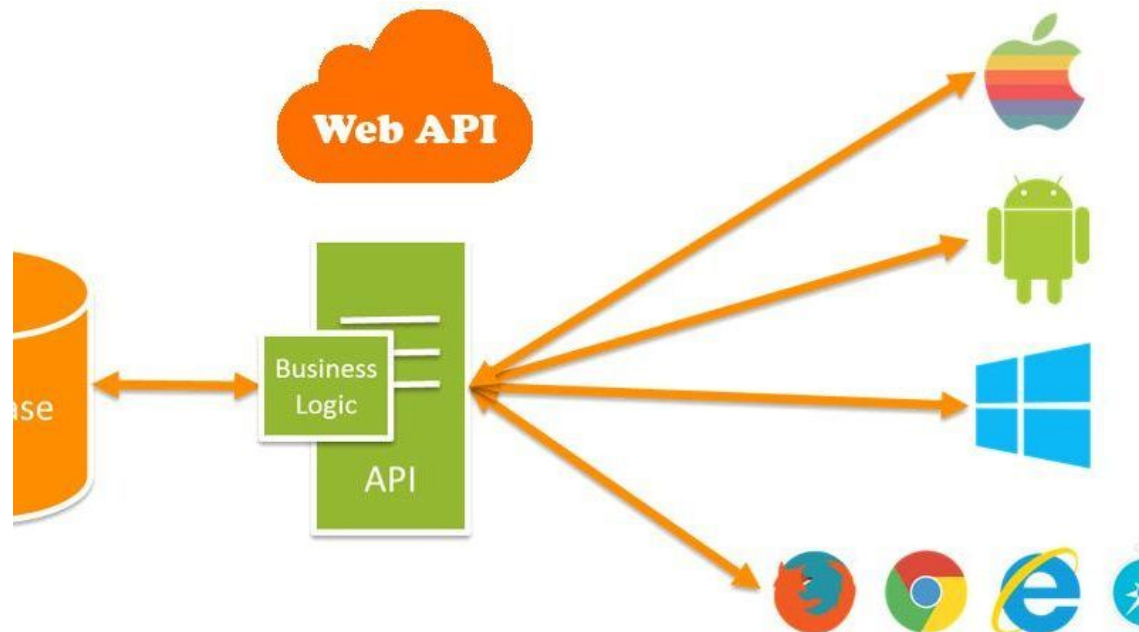


❖ API là gì?

- **API** là viết tắt của **A**pplication **P**rogramming **I**nterface (giao diện lập trình ứng dụng) phương thức kết nối với các thư viện và ứng dụng khác.
- Sử dụng API để viết các hàm, dịch vụ cho nhiều nền tảng khác nhau có thể giao tiếp được với nhau mà không bị rào cản của mỗi ngôn ngữ, đặc thù của từng nền tảng.

❖ API là gì?

- **API** là viết tắt của **A**pplication **P**rogramming **I**nterface (giao diện lập trình ứng dụng) phương thức kết nối với các thư viện và ứng dụng khác.

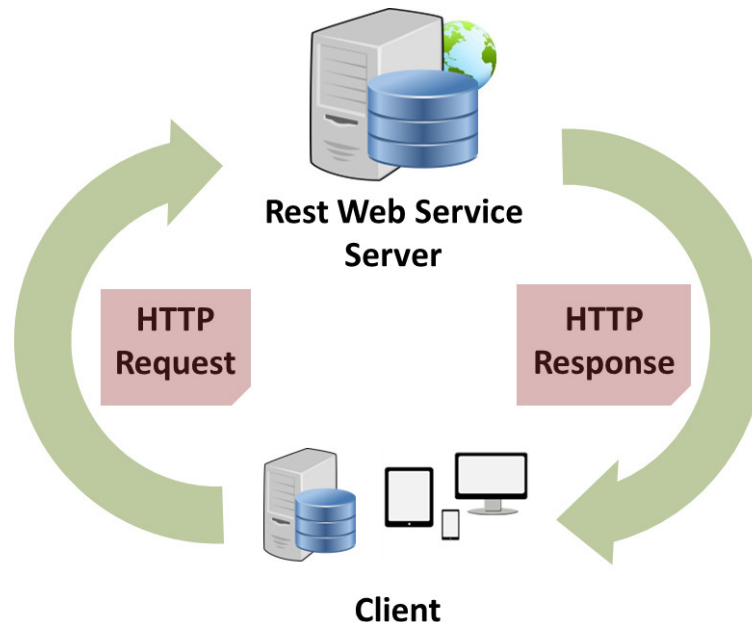


❖ Restful là gì ?

- RESTful (**RE**presentational **State** Transfer) là một kiểu Web Service được viết dựa trên kiến trúc REST. Các đặc điểm của RESTful web service:
 - Sử dụng các phương thức HTTP/HTTPs
 - Phi trạng thái (stateless) (không lưu trữ thông tin của client)
 - Hiển thị cấu trúc thư mục như các URIs
 - Truyền tải dữ liệu với định dạng phong phú: html, json, text, xml... (thường sử dụng nhiều với json)

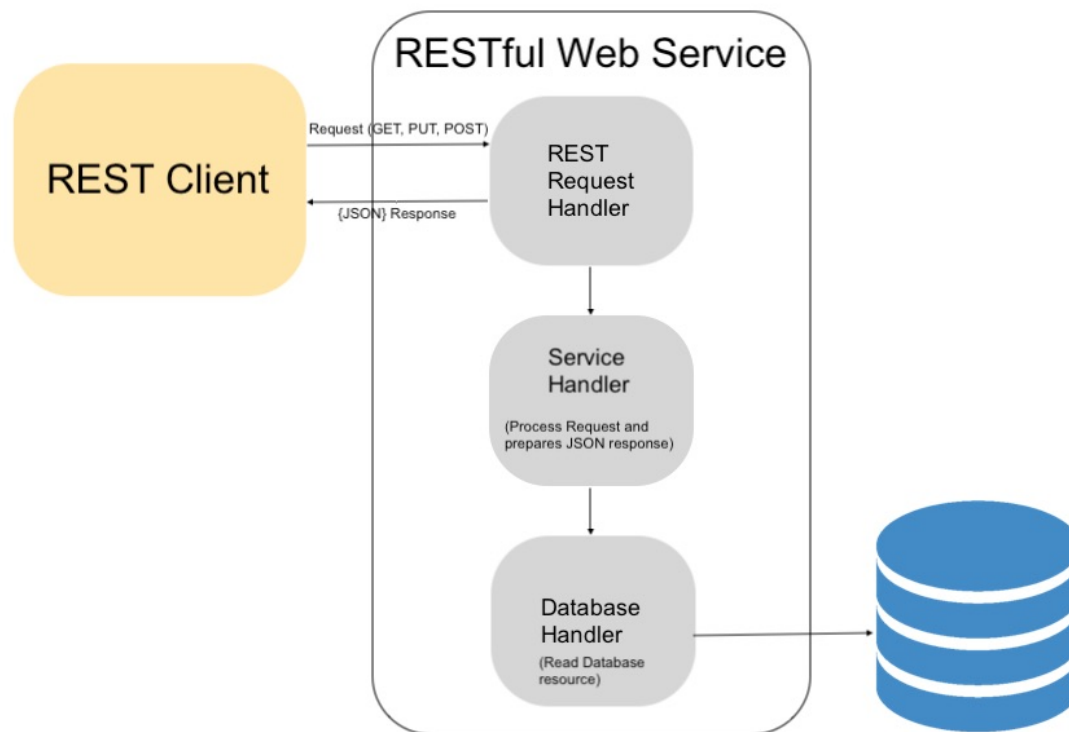
❖ Restful là gì ?

- **Dữ liệu trả về:** RESTful Web Service hỗ trợ trả về dữ liệu với nhiều định dạng khác nhau: XML, JSON, HTML... tuy nhiên chúng ta chỉ nên dùng 1 loại để tránh sự phức tạp, rắc rối khi bảo trì hay giao tiếp với hệ thống khác



❖ Restful là gì ?

- RESTful là một kiểu Web Service được viết dựa trên kiến trúc REST.



❖ Restful là gì ?

- **Các HTTP Method mà RESTful hỗ trợ:** Mỗi hành động trong web service tương ứng với 1 request method của http:
 - Method GET: được sử dụng để truy vấn, lấy dữ liệu từ server. Request sử dụng method GET chỉ nên sử dụng để lấy dữ liệu, không nên chỉnh sửa/tác động tới dữ liệu
 - Method HEAD: giống như method GET nhưng chỉ nhận về status và header, không nhận về body
 - Method POST: được sử dụng để gửi dữ liệu tới server, tạo mới tài nguyên, ví dụ upload file, đăng nhập...
 - Method PUT: được dùng để cập nhật cho tài nguyên (ví dụ update cả đối tượng user)

❖ Restful là gì ?

- **Các HTTP Method mà RESTful hỗ trợ:** Mỗi hành động trong web service tương ứng với 1 request method của http:
 - Method PATCH: giống như method PUT nhưng chỉ cập nhật cho 1 phần dữ liệu (ví dụ chỉ update thông tin 1 thuộc tính của user)
 - Method DELETE: được dùng cho hành động xóa đối tượng trên server.
 - Một số method khác: CONNECT, OPTIONS, TRACE
 - Với lập trình web, khi submit form ta hay sử dụng 2 method là GET, POST.
 - Với web service ta sẽ hay sử dụng 5 method là GET, POST, PUT, PATCH, DELETE.

❖ Restful là gì ?

- **Cấu trúc của URI:** Việc đặt URI cho các api web service thường đặt theo kiểu cấu trúc thư mục, nhiều hành động chung 1 URI nhưng khác method:
- Ví dụ:
 - GET /user: lấy thông tin tất cả user
 - GET /user/1: lấy thông tin của user có id = 1
 - POST /user: tạo user mới
 - PUT /user/1: cập nhật đối tượng user có id = 1
 - PATCH /user/1: cập nhật 1 thuộc tính cho đối tượng user có id = 1
 - DELETE /user/1: xóa đối tượng user có id = 1

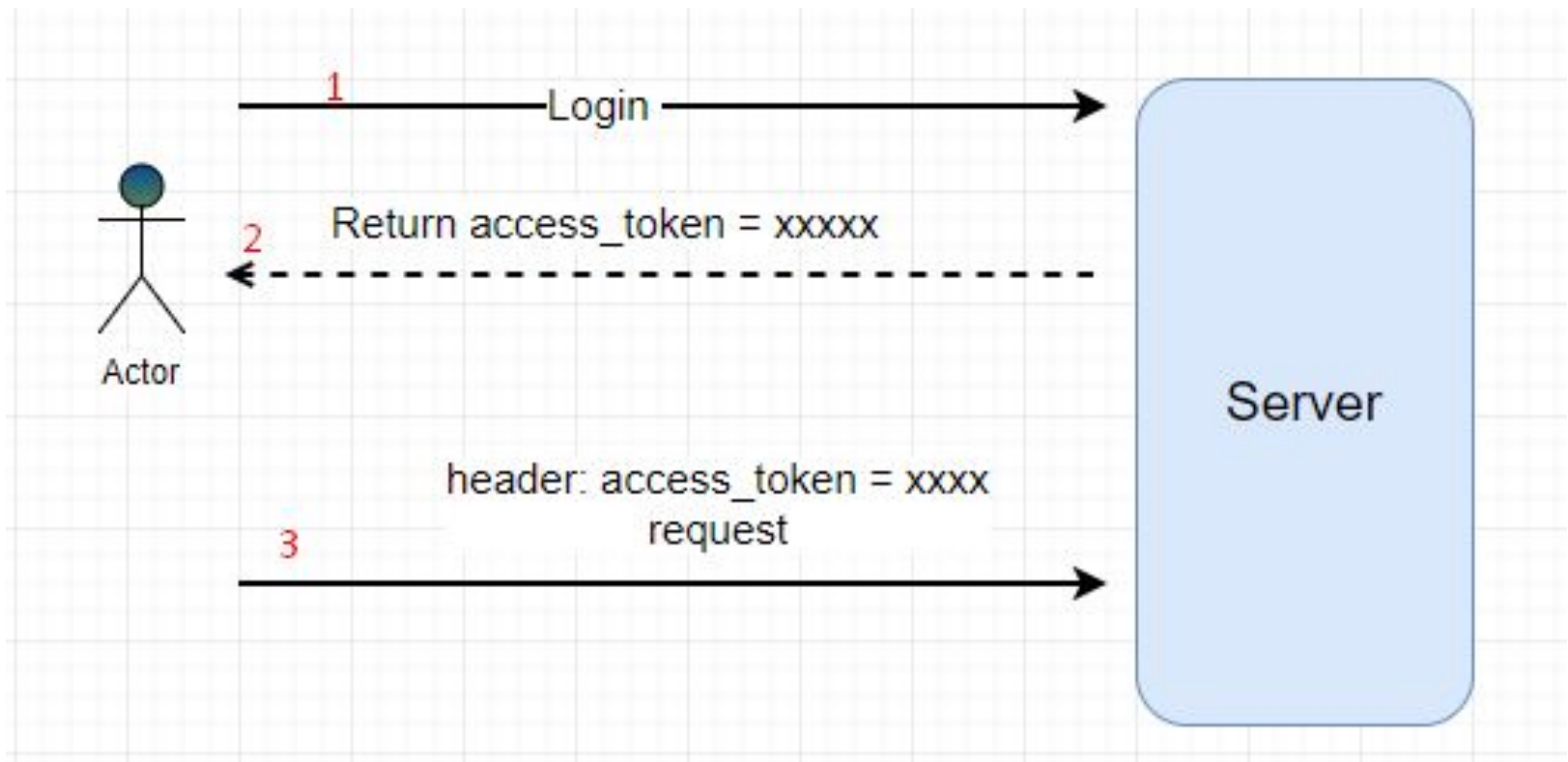
❖ Restful là gì ?

■ Bảo mật, xác thực trong restful web service

- Request trong web service là stateless – phi trạng thái, nó cũng không sử dụng cookies nên ta không thể lưu thông tin đăng nhập vào session.
- Để xác thực trong RESTful web service, sau khi đăng nhập client sẽ nhận được 1 token, mỗi lần gửi request sẽ gửi kèm token đó vào header của request. Server sẽ dựa vào token để xác thực client.

❖ Restful là gì ?

- Bảo mật, xác thực trong restful web service





❖ So sánh Webservice và Restful

- SOAP (stands for Simple Object Access Protocol) và RESTful (REpresentational State Transfer) là 2 kiểu thực hiện web service khác nhau. Sự khác nhau cơ bản giữa SOAP và RESTful là:
 - SOAP chỉ trả về dữ liệu dạng XML còn RESTful trả về dữ liệu ở nhiều định dạng khác nhau: Plain Text, HTML, XML and JSON...
 - Tốc độ của RESTful nhanh hơn SOAP
 - SOAP có tính bảo mật và toàn vẹn hơn RESTful
 - SOAP là một kiểu giao thức (protocol) khá giống HTTP, nó cũng có phần header, body... để định nghĩa dữ liệu gửi về còn RESTful giống như 1 kiểu kiến trúc gửi/ nhận (client gửi request/dữ liệu tới, server trả dữ liệu tương ứng về)



PHP for Developer

www.stanford.com.vn

Làm việc với Restful trong PHP

@stanford.com.vn

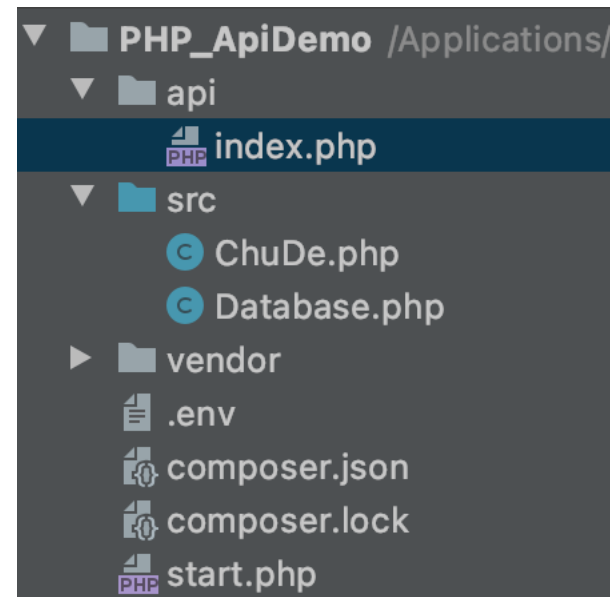
Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- Bước 1: Thực hiện tạo một folder dự án mới chứa các nội dung:

- Tạo thư mục src
- Tạo file cấu hình .env
- Tạo file composer.json



Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- Bước 2: Khai báo nội dung sau trong composer:

```
{
  "require": {
    "vlucas/phpdotenv": "^5.3"
  },
  "autoload": {
    "psr-4": {
      "Src\\": "src/"
    }
  }
}
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- Bước 3: Di chuyển đến thư mục gốc và sử dụng lệnh dưới để cài đặt gói sử dụng:

```
composer install
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 4:** Thực hiện tạo file start.php trong thư mục dự án, file này sẽ được gọi đầu tiên khi chạy dự án:

```
<?php
require 'vendor/autoload.php';
use Dotenv\Dotenv;
use Src\Database;
//Xử lý load thư viện và kết nối đến db
$dotenv = Dotenv::createImmutable(paths: __DIR__);
$dotenv->load();

//In thông tin host
//echo $_ENV['DB_HOST'];
$dbConnection = (new Database())->connect();
```


Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 5:** Khai báo và trả về kết nối với database trong file Database.php trong thư mục như sau:

```
class Database
{
    private $dbConnection = null;
    public function __construct()
    {
        $host = $_ENV['DB_HOST'];
        $port = $_ENV['DB_PORT'];
        $db = $_ENV['DB_DATABASE'];
        $user = $_ENV['DB_USERNAME'];
        $pass = $_ENV['DB_PASSWORD'];
        try {
            $this->dbConnection = new PDO(
                dsn: "mysql:$host;port=$port;dbname=$db;
                charset=utf8mb4", $user, $pass);
        } catch (\PDOException $e) {
            exit($e->getMessage());
        }
    }
}
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 5:** Khai báo và trả về kết nối với database trong file Database.php trong thư mục như sau:

```
public function connet()  
{  
    return $this->dbConnection;  
}
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 6:** Xây dựng lớp xử lý Restful như sau:

```
class ChuDe
{
    private $db;
    private $requestMethod;
    private $ma;
    public function __construct($db, $requestMethod, $ma)
    {
        $this->db = $db;
        $this->requestMethod = $requestMethod;
        $this->ma = $ma;
    }
}
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 6:** Xây dựng lớp xử lý Restful như sau:

```
public function processRequest()
{
    switch ($this->requestMethod) {
        case 'GET':
            if ($this->ma) {
                $response = $this->getChuDe($this->ma);
            } else {
                $response = $this->getAllChuDe();
            };
            break;
        case 'POST':
            $response = $this->createChuDe();
            break;
    }
}
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 6:** Xây dựng lớp xử lý Restful như sau:

```
case 'PUT':
    $response = $this->updateChuDe($this->ma);
    break;
case 'DELETE':
    $response = $this->deleteChuDe($this->ma);
    break;
default:
    $response = $this->notFoundResponse();
    break;
}
header($response['status_code_header']);
if ($response['body']) {
    echo $response['body'];
}
}
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 6:** Xây dựng lớp xử lý Restful như sau:

```
private function getAllChuDe()  
{  
    $query = "SELECT MaChuDe, TenChuDe FROM ChuDe";  
    try {  
        $statement = $this->db->query($query);  
        $result = $statement->fetchAll(PDO::FETCH_ASSOC);  
    } catch (PDOException $e) {  
        exit($e->getMessage());  
    }  
    $response['status_code_header'] = 'HTTP/1.1 200 OK';  
    $response['body'] = json_encode($result);  
    return $response;  
}
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 6:** Xây dựng lớp xử lý Restful như sau:

```
private function getChuDe($id)
{
    $result = $this->find($id);
    if (! $result) {
        return $this->notFoundResponse();
    }
    $response['status_code_header'] = 'HTTP/1.1 200 OK';
    $response['body'] = json_encode($result);
    return $response;
}
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 6:** Xây dựng lớp xử lý Restful như sau:

```
public function find($id)
{
    $query = "SELECT MaChuDe, TenChuDe, MoTa FROM ChuDe WHERE MaChuDe = :id";

    try {
        $statement = $this->db->prepare($query);
        $statement->execute(array('id' => $id));
        $result = $statement->fetch(PDO::FETCH_ASSOC);
        return $result;
    } catch (\PDOException $e) {
        exit($e->getMessage());
    }
}
```


Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 6:** Xây dựng lớp xử lý Restful như sau:

```
private function createChuDe()
{
    $input = (array) json_decode(file_get_contents( filename: 'php://input'),
    assoc: TRUE);
    if (! $this->validateChuDe($input)) {
        return $this->unprocessableEntityResponse();
    }
    $query = "INSERT INTO ChuDe(MaChuDe, TenChuDe) VALUES(:ma, :tenCD)";
    try {
        $statement = $this->db->prepare($query);
        $statement->execute(array(
            'ma' => $input['maChuDe'], 'tenCD' => $input['tenChuDe']));
        $statement->rowCount();
    } catch (\PDOException $e) {
        exit($e->getMessage());
    }
    $response['status_code_header'] = 'HTTP/1.1 201 Created';
    $response['body'] = json_encode(array('message' => 'Post Created'));
    return $response;
}
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 7:** Tạo thư mục api và file index.php với nội dung sau:

```
<?php
require "../start.php";
use Src\ChuDe;
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Methods:
OPTIONS,GET,POST,PUT,DELETE");
header("Access-Control-Max-Age: 3600");
header("Access-Control-Allow-Headers: Content-Type, Access-
Control-Allow-Headers, Authorization, X-Requested-With");
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 7:** Tạo thư mục api và file index.php với nội dung sau:

```
$uri = parse_url($_SERVER['REQUEST_URI'],  
    PHP_URL_PATH);  
$uri = explode('/', $uri);  
//print_r($uri);  
if ($uri[1] !== 'chude') {  
    if ($uri[1] !== 'chudes') {  
        header("HTTP/1.1 404 Not Found"); exit();  
    }  
}  
if ($uri[1] == 'chudes' and isset($uri[2])) {  
    header("HTTP/1.1 404 Not Found"); exit();  
}
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 7:** Tạo thư mục api và file index.php với nội dung sau:

```
$ma = null;
if (isset($uri[2])) {
    $ma = $uri[2];
}

$requestMethod = $_SERVER["REQUEST_METHOD"];
//Truyền phương thức và tham số của chủ đề
$controller = new ChuDe($dbConnection, $requestMethod,
$ma);
$controller->processRequest();
```

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

- **Bước 8:** Thực hiện chạy dự án với lệnh sau:

```
php -S localhost:8008 -t api
```



Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

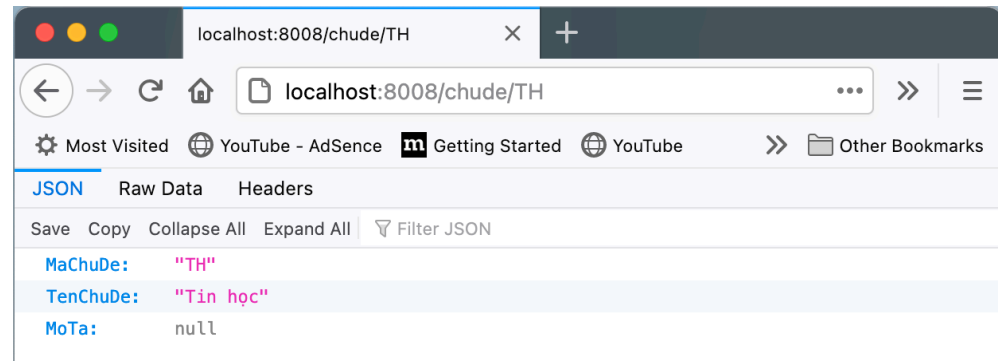
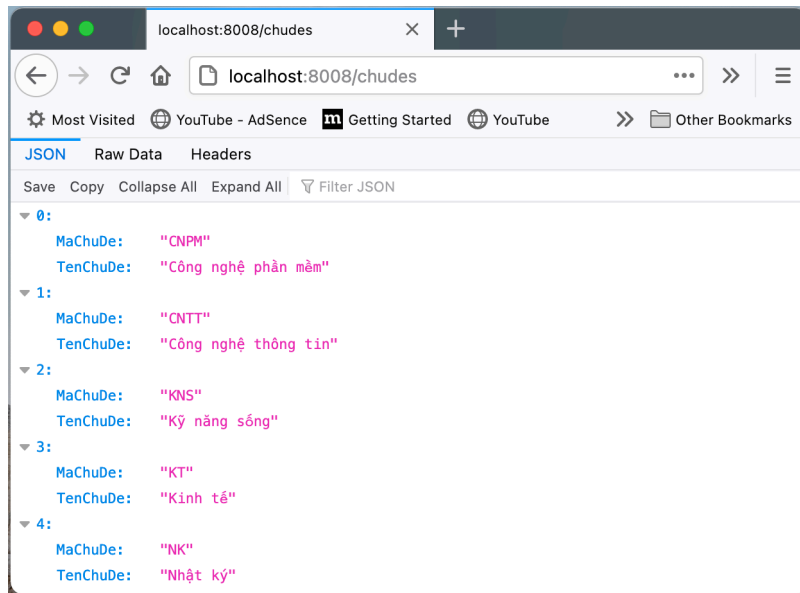
- Kết quả chạy chương trình:
 - Cài đặt công cụ [postman](#) để test các hàm API
 - Sử dụng jQuery Ajax để xử lý các hàm với API

Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

■ Kết quả chạy chương trình:

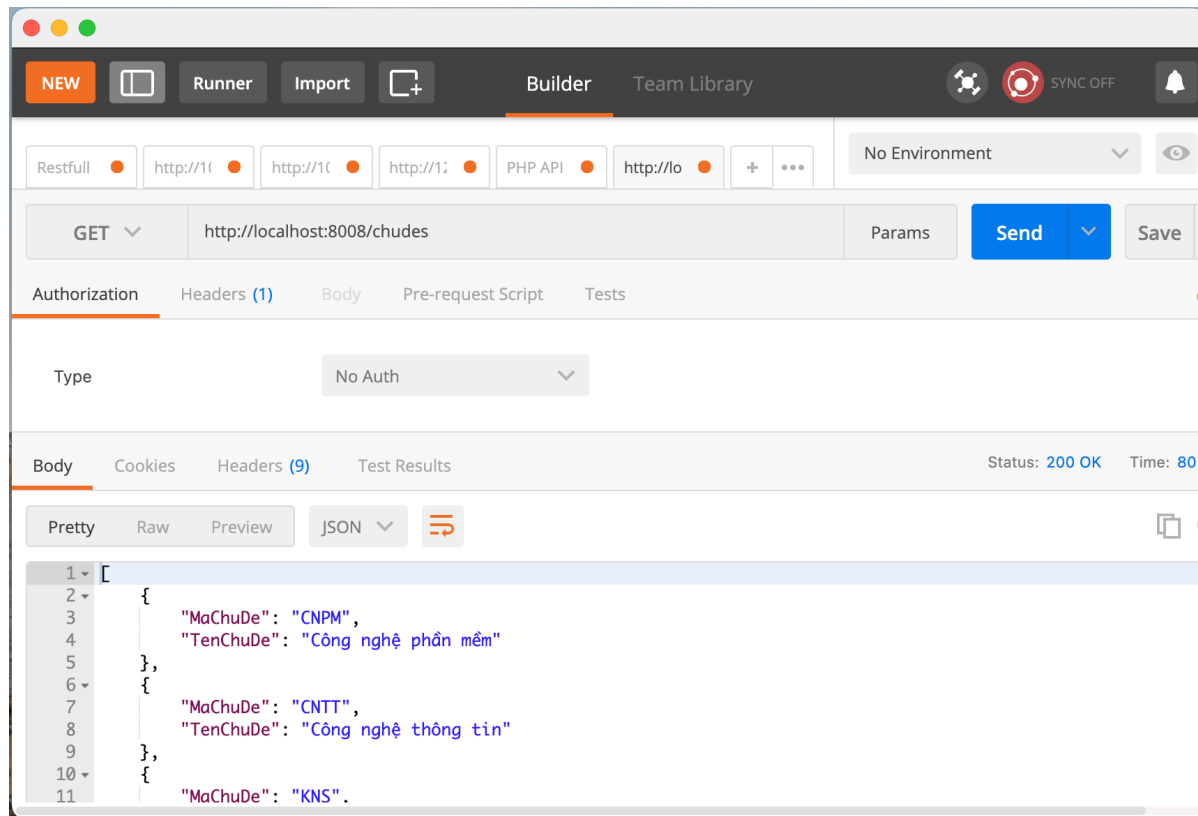


Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

■ Kết quả chạy chương trình:

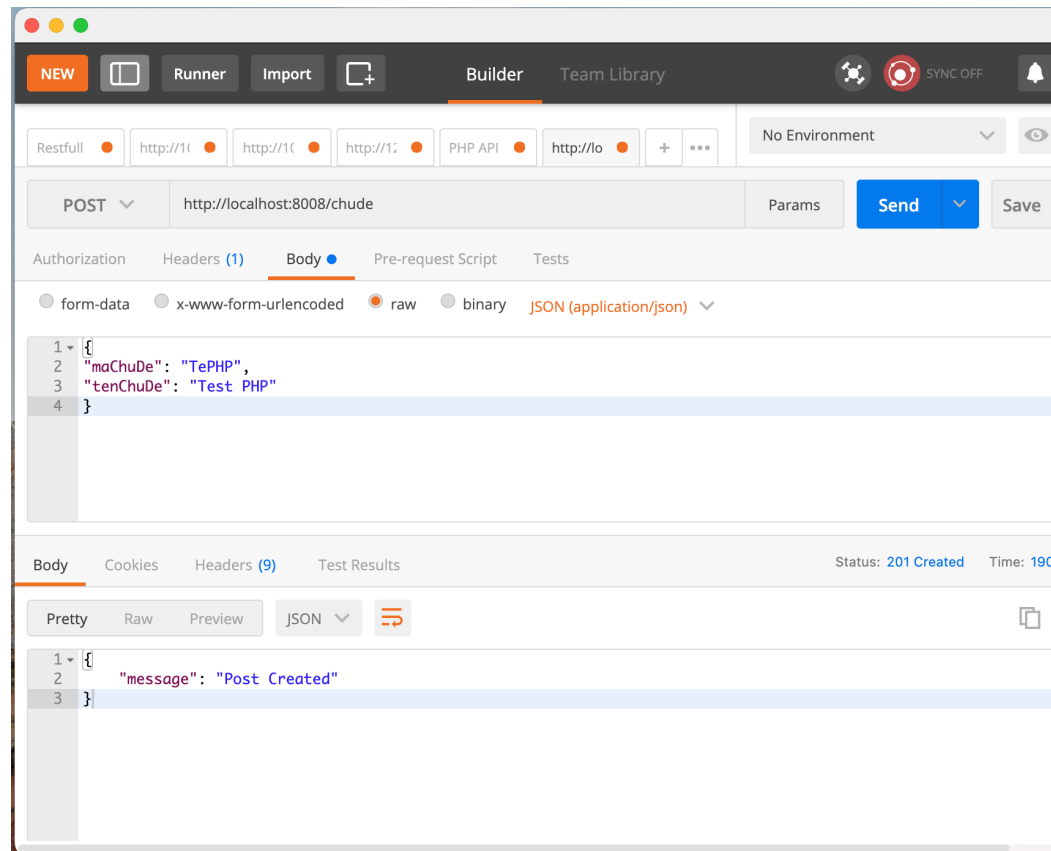


Làm việc với Restful trong PHP

www.stanford.com.vn

❖ Xây dựng Restful trong PHP

■ Kết quả chạy chương trình:





Thank You !

www.stanford.com.vn