Fpt University

FPT POLYTECHNIC

ANDROID NETWORK

BÀI 4: WEB API

www.poly.edu.vn

Mục tiêu

- Tìm hiểu Web API
- Xây dựng Database trên Web Server sử dụng MySQL
- Sử dụng PHP tạo API: đăng ký và đăng nhập

Giới thiệu về API

- Application Programming Interface: Giao diện ứng dụng lập trình
- API là các hàm, phương thức để cho các ứng dụng bên ngoài có thể gọi, tương tác để trao đổi thông tin, tính toán.
- Việc trao đổi này giúp các nhà lập trình tạo ra các service hỗ trợ những lập trình viên khác có thể tương tác với ứng dụng của chính mình.
- Ví dụ: dịch vụ của google, youtube, facebook…
- Mỗi phần mềm, ứng dụng có các cung cấp các API để các ứng dụng khác có thể tương tác với nó. Và việc xây dựng lên các API để cho các ứng dụng bên ngoài cũng cần tuân thủ các chuẩn công nghệ để nhiều nền tảng công nghệ có thể sử dụng được API mà ứng dụng cung cấp.

Giới thiệu web api

- Web API giúp chúng ta xây dựng lên các Service cung cấp dịch vụ cho các ứng dụng web, window…
- Một ứng dụng đơn giản như là: Chúng ta có 1 Web API cung cấp các dịch vụ lưu trữ dữu liệu, cung cấp các chỉ số chứng khoán, kết quả bóng đá, xổ số…
- Các ứng dụng Client như website, ứng dụng winform… có thể kết nối vào Web API để lấy các dữ liệu về xử lý, cũng như cập nhật thông tin lại Web API
- Web API dùng phương thức trao đổi dữ liệu là HTTP, kiểu dữ liệu trao đổi là JSON, một chuẩn dữ liệu hướng đối tượng được dùng khá nhiều trong việc lưu chuyển thông tin trên Internet .

TÌM HIỂU WEB API

- Do dùng jSon là kiểu dữ liệu chuyển đổi nên tốc độ các trang web sử dụng web API tương tác dữ liệu có tốc độ khá cao.
- Web API dùng giao thức HTTP nên hầu như tất cả các ứng dụng trên các công nghệ đều có thể kết nối tới để lấy cũng như tương tác với web API cụ thể như chúng ta có thể dùng các công nghệ web như : Asp.net, PHP, JSP hay các ứng dụng winform đều có thể dễ dàng kết nối tới web API.
- Do đó với Web API chúng ta có thể ứng dụng vào các dự án Web (cũng như window) lớn để phát triển trên nhiều tầng xử lý khác nhau
- Dùng web API chúng ta dễ dàng xây dựng các ứng dụng window kiểu điện toán (dữ liệu ở server) còn client chỉ cài giao diện.

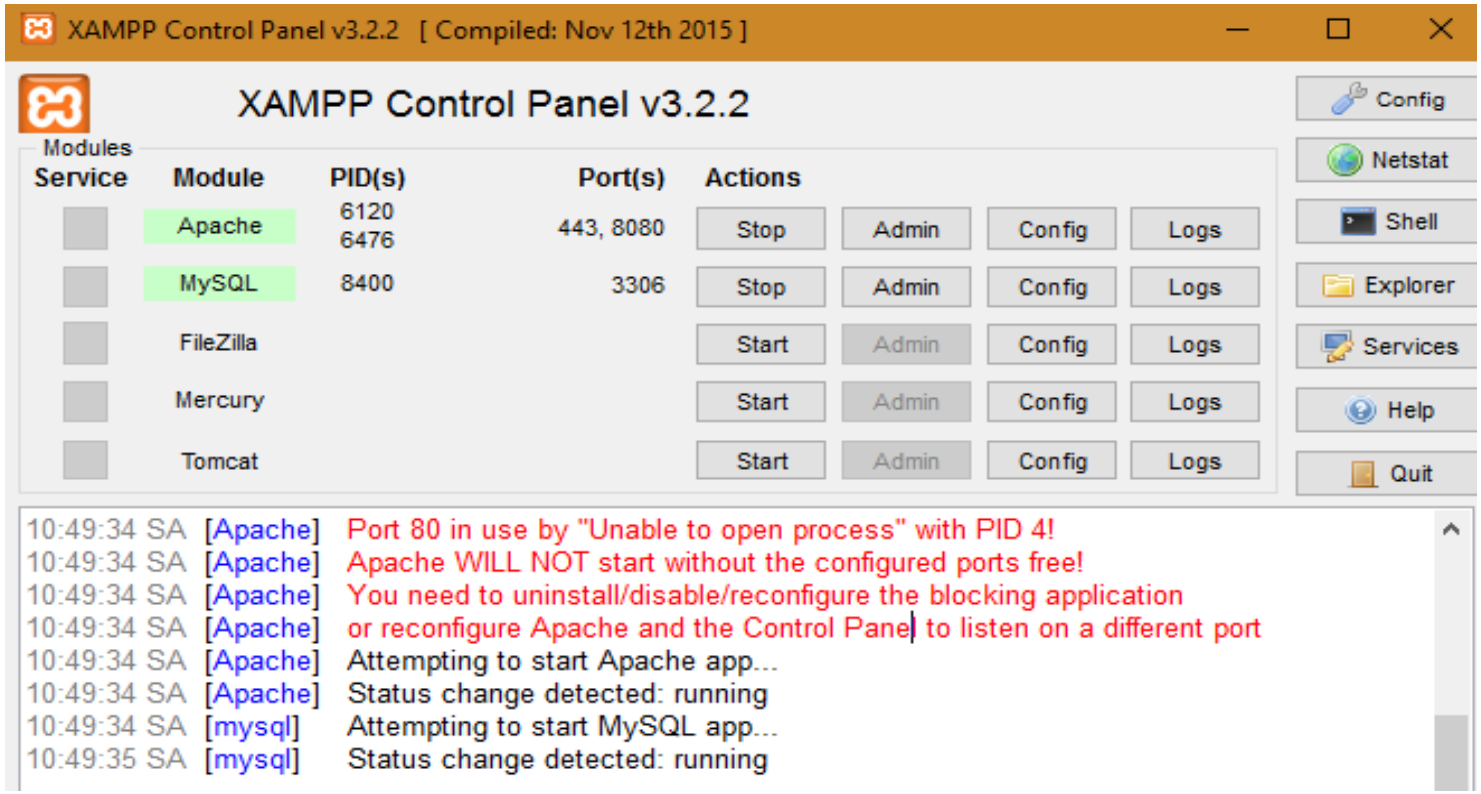
MySQL Database:

MySQL là một RDBMS nhanh và dễ dàng để sử dụng. MySQL đang được sử dụng cho nhiều công việc kinh doanh từ lớn tới nhỏ. Nó khá phổ biến vì nhiều lý do:

- MySQL là mã ngồn mở.
- MySQL là một chương trình rất mạnh mẽ.
- MySQL sử dụng một Form chuẩn của ngôn ngữ dữ liệu nổi tiếng là SQL.
- MySQL làm việc trên nhiều Hệ điều hành và với nhiều ngôn ngữ như PHP, PERL, C, C++, Java, …
- MySQL rất thân thiện với PHP, một ngôn ngữ rất đáng giá để tìm hiểu để phát triển Web.
- MySQL hỗ trợ các cơ sở dữ liệu lớn, lên tới 50 triệu hàng hoặc nhiều hơn nữa trong một bảng.

Xây dựng MySQL Database:

Để cài đặt MySQL trên local, chúng ta cần cài Xampp, link download: https://www.apachefriends.org/download.html



XÂY DỰNG DATABASE TRÊN WEB SERVER

Tạo Database:

create database <tên database>

Ví dụ:
create database learn-login-register

Truy xuất Database:

use <tên database>

Ví dụ:
use learn-login-register

Tạo table:

create table <tên table>(…)

Ví dụ: tạo bảng User

```
CREATE TABLE users (
    sno int(11) NOT NULL AUTO_INCREMENT,
    unique_id varchar(23) NOT NULL,
    name varchar(50) NOT NULL,
    email varchar(50) NOT NULL,
    encrypted_password varchar(256) NOT NULL,
    salt varchar(10) NOT NULL,
    created_at datetime DEFAULT NULL,
    PRIMARY KEY (sno)
)
```

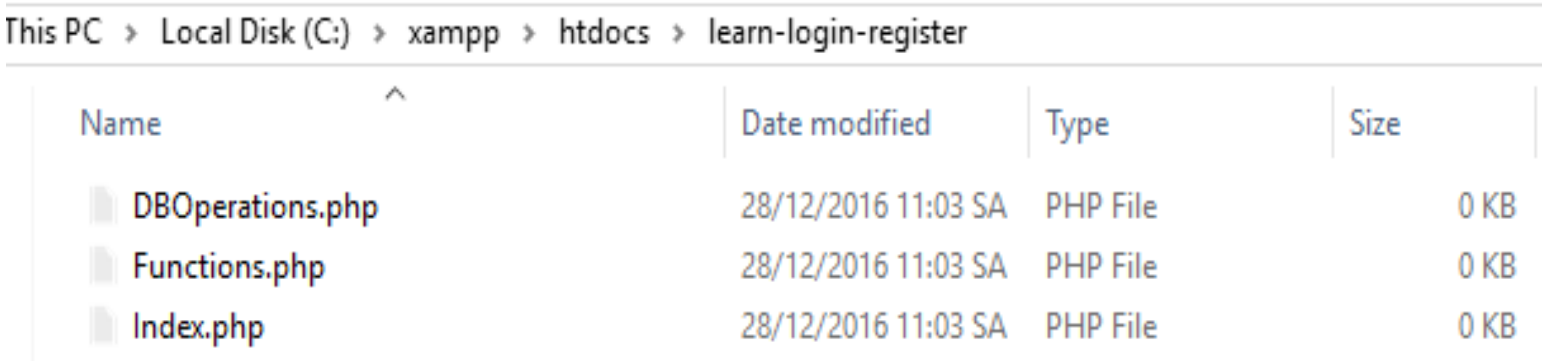
Các lệnh SQL cơ bản: (GV ôn lại bài cho SV)

- ✓ SELECT được dùng khi bạn muốn đọc (hoặc lựa chọn) dữ liệu của bạn.
- ✓ INSERT được dùng khi bạn muốn thêm (hoặc chèn) dữ liệu mới.
- ✓ UPDATE được sử dụng khi bạn muốn thay đổi (hoặc cập nhật) dữ liệu sẵn có.
- ✓ DELETE được sử dụng khi bạn muốn loại bỏ (hoặc xóa) dữ liệu sẵn có.


DEMO
Xây dựng Database trên MySQL

SỬ DỤNG PHP TẠO API: ĐĂNG KÝ VÀ ĐĂNG NHẬP

Mở thư mục htdocs ở thư mục xampp, tạo folder mới đặt tên learn-login-register và tạo 3 file php như sau:



- ✓ File DBOperations.php chứa các hàm kết nối với MySQL database như insertData(), checkLogin(), chekcUserExist()và các câu lệnh thực thi được định nghĩa ở file này.
- ✓ File Functions.php chứa các hàm gọi các phương thức được định nghĩa ở DBOperations.php xử lý việc đăng nhập, đăng ký, thay đổi mật khẩu và cuối cùng trả kết quả JSON success hoặc failure
- ✓ File Index.php là file chính nhận JSON data từ phía android, parse JSON data và gửi data đến file Functions.php xử lý

FILE DBOPERATIONS.PHP

Khai thông số kết nối MySQL

```php
<?php

class DBOperations{

    private $host = '127.0.0.1';
    private $user = 'root';
    private $db = 'learn-login-register';
    private $pass = '';
    private $conn;

public function __construct() {

    $this -> conn = new PDO("mysql:host=".$this -> host.";dbname=".$this -> db, $this -> user, $this -> pass);

}
```

Kết nối MySQL dùng PDO

FILE DBOPERATIONS.PHP

```php
public function insertData($name,$email,$password){

    $unique_id = uniqid('', true);
    $hash = $this->getHash($password);
    $encrypted_password = $hash["encrypted"];
    $salt = $hash["salt"];

    $sql = 'INSERT INTO users SET unique_id =:unique_id,name =:name,
    email =:email,encrypted_password =:encrypted_password,salt =:salt,created_at = NOW()';

    $query = $this ->conn ->prepare($sql);
    $query->execute(array('unique_id' => $unique_id, ':name' => $name, ':email' => $email,
     ':encrypted_password' => $encrypted_password, ':salt' => $salt));

    if ($query) {

        return true;

    } else {

        return false;

    }
}
```

FILE DBOPERATIONS.PHP

```php
public function checkLogin($email, $password) {

    $sql = 'SELECT * FROM users WHERE email = :email';
    $query = $this -> conn -> prepare($sql);
    $query -> execute(array(':email' => $email));
    $data = $query -> fetchObject();
    $salt = $data -> salt;
    $db_encrypted_password = $data -> encrypted_password;

    if ($this -> verifyHash($password.$salt,$db_encrypted_password) ) {


        $user["name"] = $data -> name;
        $user["email"] = $data -> email;
        $user["unique_id"] = $data -> unique_id;
        return $user;

    } else {

        return false;
    }

}
```

FILE DBOPERATIONS.PHP

```php
public function checkUserExist($email){

    $sql = 'SELECT COUNT(*) from users WHERE email =:email';
    $query = $this -> conn -> prepare($sql);
    $query -> execute(array('email' => $email));

    if($query){

        $row_count = $query -> fetchColumn();

        if ($row_count == 0){

            return false;

        } else {

            return true;

        }
    } else {

        return false;
    }
}

public function getHash($password) {

    $salt = sha1(rand());
    $salt = substr($salt, 0, 10);
    $encrypted = password_hash($password.$salt, PASSWORD_DEFAULT);
    $hash = array("salt" => $salt, "encrypted" => $encrypted);

    return $hash;

}
```

FILE FUNCTIONS.PHP

```php
public function registerUser($name, $email, $password) {

    $db = $this -> db;

    if (!empty($name) && !empty($email) && !empty($password)) {

        if ($db -> checkUserExist($email)) {

            $response["result"] = "failure";
            $response["message"] = "User Already Registered !";
            return json_encode($response);

        } else {

            $result = $db -> insertData($name, $email, $password);

            if ($result) {

                $response["result"] = "success";
                $response["message"] = "User Registered Successfully !";
                return json_encode($response);

            } else {

                $response["result"] = "failure";
                $response["message"] = "Registration Failure";
                return json_encode($response);

            }
        }
    } else {

        return $this -> getMsgParamNotEmpty();

    }
}
```

FILE FUNCTIONS.PHP

```php
public function loginUser($email, $password) {
    $db = $this -> db;
    if (!empty($email) && !empty($password)) {

        if ($db -> checkUserExist($email)) {

            $result =  $db -> checkLogin($email, $password);

            if(!$result) {

             $response["result"] = "failure";
             $response["message"] = "Invaild Login Credentials";
             return json_encode($response);

            } else {

             $response["result"] = "success";
             $response["message"] = "Login Successful";
             $response["user"] = $result;
             return json_encode($response);

            }

        } else {

            $response["result"] = "failure";
            $response["message"] = "Invaild Login Credentials";
            return json_encode($response);

        }
    } else {

        return $this -> getMsgParamNotEmpty();
    }

}
```

FILE INDEX.PHP

```php
<?php

require_once 'Functions.php';

$fun = new Functions();


if ($_SERVER['REQUEST_METHOD'] == 'POST')
{
  $data = json_decode(file_get_contents("php://input"));

  if(isset($data -> operation)){

    $operation = $data -> operation;

    if(!empty($operation)){

        if($operation == 'register'){

            if(isset($data -> user ) && !empty($data -> user) && isset($data -> user -> name)
                && isset($data -> user -> email) && isset($data -> user -> password)){

                $user = $data -> user;
                $name = $user -> name;
                $email = $user -> email;
                $password = $user -> password;

          if ($fun -> isEmailValid($email)) {

            echo $fun -> registerUser($name, $email, $password);

          } else {

            echo $fun -> getMsgInvalidEmail();
          }

```

FILE INDEX.PHP

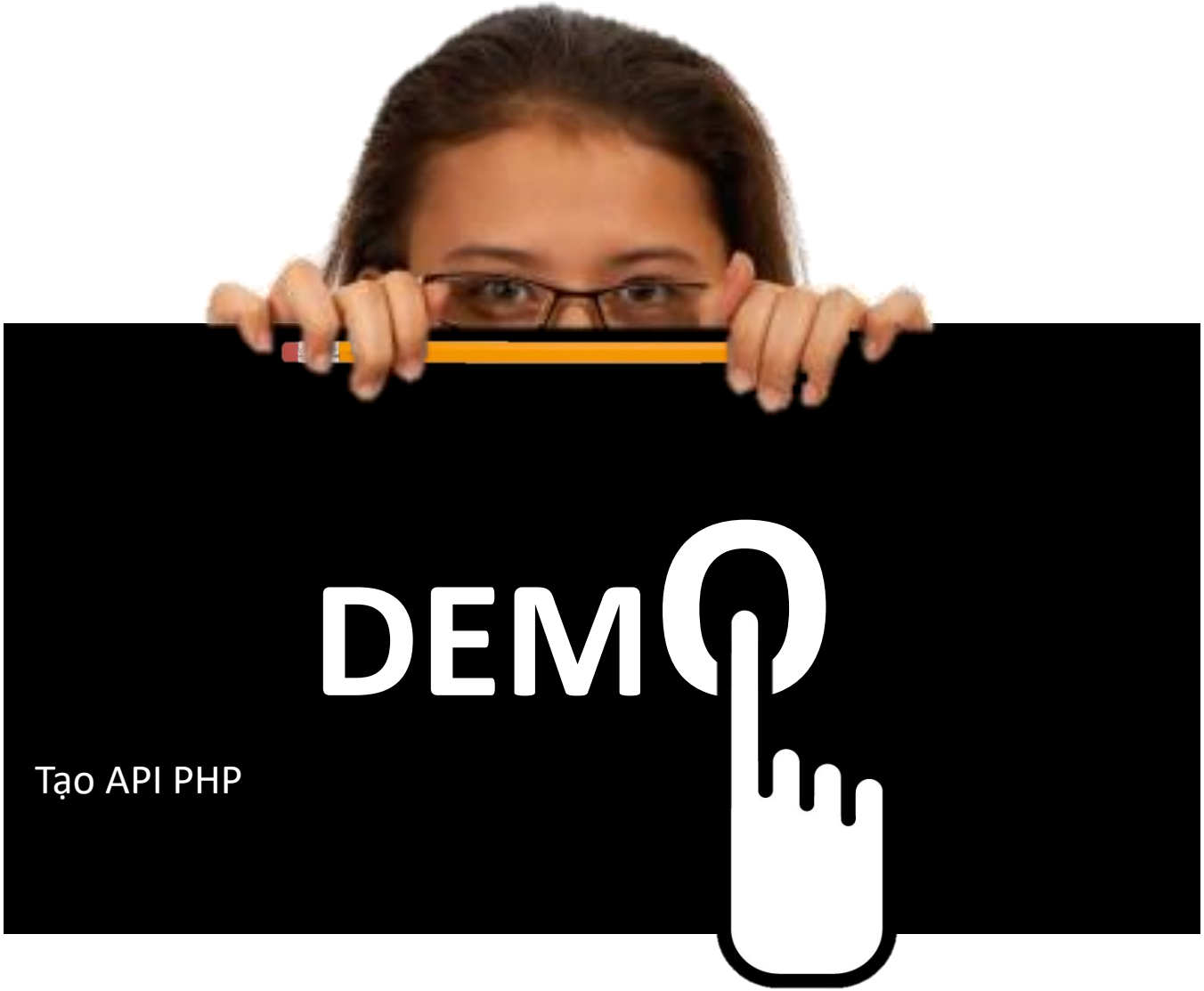
```
37          } else {
38
39              echo $fun -> getMsgInvalidParam();
40
41          }
42
43      }else if ($operation == 'login') {
44
45      if(isset($data -> user ) && !empty($data -> user) && isset($data -> user -> email) && isset($data -> user -> password)){
46
47        $user = $data -> user;
48        $email = $user -> email;
49        $password = $user -> password;
50
51        echo $fun -> loginUser($email, $password);
52
53      } else {
54
55        echo $fun -> getMsgInvalidParam();
56
57      }
58    } else if ($operation == 'chgPass') {
59
60      if(isset($data -> user ) && !empty($data -> user) && isset($data -> user -> email) && isset($data -> user -> old_password)
61        && isset($data -> user -> new_password)){
62
63        $user = $data -> user;
64        $email = $user -> email;
65        $old_password = $user -> old_password;
66        $new_password = $user -> new_password;
67
68        echo $fun -> changePassword($email, $old_password, $new_password);
69
70      } else {
71
```

FILE INDEX.PHP

```
72          echo $fun -> getMsgInvalidParam();
73
74        }
75      }else if ($operation == 'resPassReq') {
76
77        if(isset($data -> user) && !empty($data -> user) &&isset($data -> user -> email)){
78
79          $user = $data -> user;
80          $email = $user -> email;
81
82          echo $fun -> resetPasswordRequest($email);
83
84        } else {
85
86          echo $fun -> getMsgInvalidParam();
87
88        }
89      }else if ($operation == 'resPass') {
90
91        if(isset($data -> user) && !empty($data -> user) && isset($data -> user -> email) && isset($data -> user -> password)
92          && isset($data -> user -> code)){
93
94          $user = $data -> user;
95          $email = $user -> email;
96          $code = $user -> code;
97          $password = $user -> password;
98
99          echo $fun -> resetPassword($email,$code,$password);
100
101       } else {
102
103         echo $fun -> getMsgInvalidParam();
104
105       }
106     }
```

FILE INDEX.PHP

```
107
108         }else{
109
110
111             echo $fun -> getMsgParamNotEmpty();
112
113         }
114     } else {
115
116             echo $fun -> getMsgInvalidParam();
117
118     }
119 } else if ($_SERVER['REQUEST_METHOD'] == 'GET'){
120
121
122     echo "Learn2Crack Login API";
123
124 }
125
```


DEMO
Tạo API PHP

Các bước làm việc trên Android với Web API

1. Xây dựng ứng dụng android
2. Sử dụng các kỹ thuật JSON parser để lấy dữ liệu JSON từ Server và dùng AsyncTask.


DEMO
Ứng dụng Login và Register

HOẠT ĐỘNG MỞ

Sinh viên tìm hiểu và sử dụng API có sẵn của tài nguyên để thực hiện các công việc:

1. Thay đổi mật khẩu.
2. Reset password về mail.
3. Hiển thị thông tin User.

TỔNG KẾT NỘI DUNG BÀI HỌC

- Tìm hiểu Web API
- Xây dựng Database trên Web Server sử dụng MySQL
- Sử dụng PHP tạo API: đăng ký và đăng nhập