



G's ACADEMY  
TOKYO

# PHP中級(1) (2015.10.11)

azm



# これからのカリキュラム

基礎は前回までで終了です。

これからは、卒業制作に向けて、「自分でサービスを」「実践的・効率的に開発する」スキルについて取り上げていきます。

常に「どういったサービスを作るか」を心がけてください。

# PHP中級 ゴール

## 1 WebAPI・ライブラリの利用

(WebAPIやライブラリを利用したマッシュアップサービスの作り方の導入)

(1.5)メンター紹介

## 2 PHP中級

(自作関数や自作クラスの作成やその設計)

## 3 CMS / フレームワーク

(CMSを利用したAPIの作成や、PHP フレームワークなど開発の効率化ツールの導入)

# 発表会

# PHP中級(1) 課題

ぐるなびAPIを利用したサイト

# PHP中級(1) 概要

1. json とは
2. json を触ってみる(ファイル)
3. json を触ってみる(ぐるなびAPI )
4. ライブラリとは
5. dBugライブラリを利用したデバッグ
6. HybridAuth ライブラリを利用して twitter ログイン
7. おまけ

# 1. json とは

JSON (JavaScript Object Notation)は、  
軽量のデータ交換フォーマット。

“{” で始まり、“キー”：“値” のセットがあり、“}” で終わる。  
セットは“,(コンマ)”で区切る。

```
{  
  "name": "あずま",  
  "gender": "男"  
}
```

# 1. json とは

配列は、“[ ]角かっこ”でくくり、“,(コンマ)”で区切る。

シングルクォート(')は囲むのに使えない。

```
{  
  "address" : ["渋谷", "新宿", "池袋"]  
}
```

# 1. json とは

似たフォーマットにXML があります。

フォーマットの変遷については、下記サイトにうまくまとまっています。

[第1回 進化するフォーマット | CSV、XML、JSON...データ  
フォーマットの変遷について考える](#)



## 2. json を触ってみる(ファイル)

単純なjsonファイル読み込み1(json\_decode1.php)

```
<?php
```

```
$url = "data/sample1.json";
```

```
$json_data = file_get_contents($url, true);
```

```
$data = json_decode($json_data);
```

```
foreach ($data as $key => $value){
```

```
    echo $key . " = " . $value . "<br>";
```

```
}
```

```
?>
```

## 2. json を触ってみる(ファイル)

単純なjsonファイル読み込み2(json\_decode2.php)

```
<?php
$url = "data/sample2.json";
$json_data = file_get_contents($url, true);
$data = json_decode($json_data);
foreach ($data as $key => $value){
    if (!is_object($value)) {
        echo $key . " = " . $value . "<br>";
    } else {
        echo $key . "<br>";
        foreach ($value as $key => $value2) {
            echo " " . $key . " = " . $value2 . "<br>";
        }
    }
}
?>
```

## 2. json を触ってみる(ファイル)

単純なjsonファイル読み込み2(値だけ表示)(json\_decode22.php)

```
<?php
```

```
$url = "data/sample2.json";
```

```
$json_data = file_get_contents($url, true);
```

```
$data = json_decode($json_data);
```

```
echo $data->name . "<br>";
```

```
echo $data->gender . "<br>";
```

```
echo $data->info->living . "<br>";
```

```
echo $data->info->birthday . "<br>";
```

```
?>
```

## 2. json を触ってみる(ファイル)

### ワーク

sample3.json の値を表示する  
json\_decode33.php を実装する

余力があれば、キーと値を表示する  
json\_decode3.php を実装する

### 3. json を触ってみる(ぐるなびAPI)

ぐるなびWebサービス(APIテストツールが便利)

API(WebAPI)を使うときは、まずはドキュメントを読むこと！

- 利用規約(API利用ルール)
- API仕様

※利用回数制限など！

### 3. json を触ってみる(ぐるなびAPI)

APIの仕様は大きく2つ定義されている

- リクエスト(リクエストパラメータ)

リクエストは、こういったデータが欲しいという問い合わせ内容

- レスポンス

レスポンスは、リクエストした結果の返ってくる形

### 3. json を触ってみる(ぐるなびAPI)

ぐるなびAPIでは、APIテストツールが提供されています。

これを利用して、色々試してみましょう。

#### ※JSON整形サービス

(整形と文字コード(Unicode)変換してくれる)

### 3. json を触ってみる(ぐるなびAPI)

ぐるなび都道府県マスタ取得API(json\_decode4.php)

```
<?php
```

```
require("config.php");
```

```
$url = "http://api.gnavi.co.jp/master/PrefSearchAPI/20150630/?
```

```
keyid=" . $key_id . "&format=json";
```

```
$json_data = file_get_contents($url, true);
```

```
$data = json_decode($json_data);
```

```
$pref_array = $data->pref;
```

```
foreach ($pref_array as $key => $pref_value){
```

```
    echo $pref_value->pref_code . ":";
```

```
    echo $pref_value->pref_name . "<br>";
```

```
}
```

```
?>
```



### 3. json を触ってみる(ぐるなびAPI)

ぐるなびレストラン検索API(json\_decode5.php)

```
<?php
```

```
require("config.php");
```

```
$url = "http://api.gnavi.co.jp/RestSearchAPI/20150630/?keyid=" .
```

```
$key_id . "&format=json";
```

```
$json_data = file_get_contents($url, true);
```

```
$data = json_decode($json_data);
```

```
var_dump($data->rest[0]);
```

```
echo "<hr>";
```

```
echo $data->rest[0]->name . "<br>";
```

```
?>
```

### 3. json を触ってみる(ぐるなびAPI)

ぐるなびレストラン検索API(json\_decode5.php)

```
<?php
```

```
require("config.php");
```

```
$url = "http://api.gnavi.co.jp/RestSearchAPI/20150630/?keyid=" .
```

```
$key_id . "&format=json";
```

```
$json_data = file_get_contents($url, true);
```

```
$data = json_decode($json_data);
```

```
var_dump($data->rest[0]);
```

```
echo "<hr>";
```

```
echo $data->rest[0]->name . "<br>";
```

```
?>
```

## 4. ライブラリとは

ライブラリとは、便利な機能をPHP に追加する  
「phpのファイル」「php ディレクトリ」

ライブラリのインストール方法

- ・ファイルをダウンロードして、サーバーにアップロード
- ・PEAR を使って、サーバーにインストール
- ・git (gitコマンド)を使って、サーバーにインストール
- ・Composer を使って、サーバーにインストール

## 5. dBugライブラリを利用したデバッグ

```
dbugライブラリ(debug.php)
<?php
require("lib/dBug/dBug.php");
$url = "data/sample1.json";
$json_data = file_get_contents($url, true);
$data = json_decode($json_data);
var_dump($data);
echo"<hr>";
new dBug($data);
?>
```

## 6. HybridAuth ライブラリを利用して twitter ログイン

twitter アプリケーションの設定

Callback URL

`http://127.0.0.1/hybridauth/?hauth.done=Twitter`

**azgz test** Test OAuth

Details Settings Keys and Access Tokens Permissions

### Application Details

**Name \***

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

**Description \***

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

**Website \***

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your a source attribution for tweets created by your application and will be shown in user-facing authorization screens.  
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

**Callback URL**

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their OAuth callback URL on the

**azgz test** Test OAuth

Details Settings Keys and Access Tokens Permissions

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	KQ9AnRw9ar0hN2vOQA4s4DfYV
Consumer Secret (API Secret)	64xzwMJJa1DkdsRytuENQTrJbuDvLoMt4CQqYRxcXQqEDxf6XDb
Access Level	Read and write ( <a href="#">modify app permissions</a> )
Owner	azazgz1
Owner ID	3800910434

## 6. HybridAuth ライブラリを利用して twitter ログイン

config.php

<?php

return

```
array(  
    // "base_url" => "http://localhost/hybridauth-git/hybridauth/",  
    "base_url" => "http://127.0.0.1/php1011/hybridauth/",  
    "providers" => array(  
        "Twitter" => array(  
            "enabled" => true,  
            // "keys" => array("key" => "", "secret" => ""),  
            "keys" => array("key" => "KQ9AnRw9ar0hN2vOQA4s4DfYV", "secret" =>  
                "64xzwMJJa1DkdsRytuENQTrJbuDvLoMt4CQqyRxcXQqEDxf6XDb"),  
            "includeEmail" => false  
        ),  
    ),  
    // If you want to enable logging, set 'debug_mode' to true.  
    // You can also set it to  
    // - "error" To log only error messages. Useful in production  
    // - "info" To log info and error messages (ignore debug messages)  
    "debug_mode" => false,  
    // "debug_mode" => true,  
    // Path to file writable by the web server. Required if 'debug_mode' is not false  
    "debug_file" => "",  
);
```

## 6. HybridAuth ライブラリを利用して twitter ログイン

login.php(新規作成)

```
<?php
$config = "config.php";
require_once("Hybrid/Auth.php");
try {
    $hybridauth = new Hybrid_Auth($config);
    $twitter = $hybridauth->authenticate( "Twitter" );
    $user_profile = $twitter->getUserProfile();
    var_dump($user_profile);
    echo "<hr>";
    echo "Hi there! " . $user_profile->displayName;
} catch ( Exception $e ) {
    echo "Ooops, we got an error: " . $e->getMessage();
}
?>
```

# PHP中級(1) 課題

## ぐるなびAPIを利用したサイト

- ・フォームからエリアを絞り込みさせたり、
- ・お店の位置情報と地図系APIを使ってみたり、
- ・データベースと連携させてお店のクリップ機能をつけたり



## 7. おまけ

git を使ったインストール

```
$ mkdir /Applications/XAMPP/htdocs/test
```

```
$ cd /Applications/XAMPP/htdocs/test
```

```
$ git clone git://github.com/hybridauth/hybridauth.git
```

```
Cloning into 'hybridauth'...
```

```
remote: Counting objects: 6472, done.
```

```
remote: Total 6472 (delta 0), reused 0 (delta 0), pack-reused 6472
```

```
Receiving objects: 100% (6472/6472), 5.47 MiB | 453.00 KiB/s,  
done.
```

```
Resolving deltas: 100% (3325/3325), done.
```

```
Checking connectivity... done.
```

# 7. おまけ

## Composer のインストール

Mac

```
$ curl -sS https://getcomposer.org/installer | php
```

```
#!/usr/bin/env php
```

All settings correct for using Composer

Downloading...

Composer successfully installed to:

/Applications/XAMPP/xamppfiles/htdocs/test/composer.phar

Use it: php composer.phar

```
$ mv composer.phar /usr/local/bin/composer
```

# 7. おまけ

Composer の利用

Mac

```
$ vi composer.json
```

```
{  
    "require": {  
        "monolog/monolog": "1.0.*"  
    }  
}
```

```
$ composer install
```

Loading composer repositories with package information

Installing dependencies (including require-dev)

- Installing monolog/monolog (1.0.2)

Downloading: 100%

Writing lock file

Generating autoload files

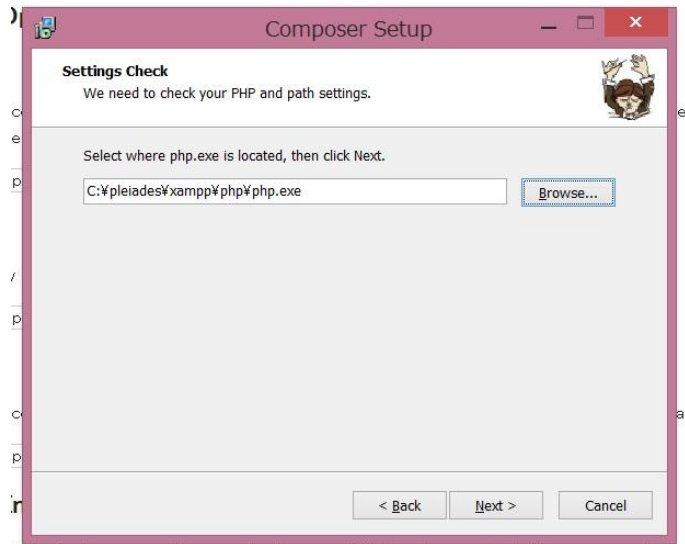
# 7. おまけ

## Composer のインストールの注意点

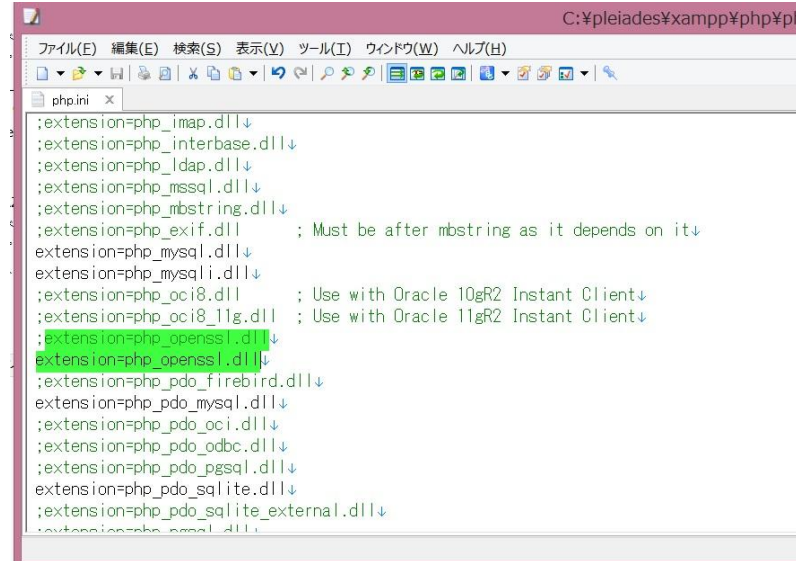
### Win

PHPのパスはXAMPP以下を指定

php.ini の;extension=php\_openssl.dll を有効化



If you download composer for you and set up your PATH environment variable so you can simply



### 参考サイト

<http://goo.gl/Ex7xWS>

<http://goo.gl/WYKvdz>