

# Web Service

- A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. (W3C, Web Services Glossary[1])
- Free OS dependency
- Free programming language dependency
- Designed specific
- As simple as possible
- Keep it secure
- Lightweight
- Versioning

# Format & HTTP status

- JSON

```
{  
  message: "successfully created",  
  object_id: 10200  
  status: 201  
}
```

- XML

```
<message>succesfully created</message>  
<object_id>10200</object_id>  
<status>201</status>
```

- <http://www.restapitutorial.com/httpstatuscodes.html>

# Consumed by

- Desktop calls
- Mobile app
- Web / other W/S
- etc

# Laravel requirements

- <https://laravel.com/docs/5.3/installation>
  - <https://nodejs.org/dist/v4.6.1/node-v4.6.1-linux-x64.tar.xz>
  - PHP  $\geq$  5.6.4
  - OpenSSL PHP Extension
  - PDO PHP Extension
  - Mbstring PHP Extension
  - Tokenizer PHP Extension
  - XML PHP Extension
- Mysql

# Installations (php)

```
sudo apt-get update
```

```
sudo apt-get install python-software-properties
```

```
sudo apt-get update
```

```
sudo LC_ALL=C.UTF-8 add-apt-repository ppa:ondrej/php
```

```
sudo apt-get update
```

```
sudo apt-get install php7.0
```

```
php -v
```

*PHP 7.0.12-1+deb.sury.org~trusty+1 (cli) ( NTS )*

Install other required extension: php7.0-mbstring, phpunit, php7.0-zip

# Installations (composer)

- <https://getcomposer.org/installer>  
./installer
- <https://getcomposer.org/download/>  
<https://getcomposer.org/download/1.2.1/composer.phar>

# Installations (mysql)

```
sudo apt-get install mysql-server libapache2-mod-auth-mysql php7.0-mysql
```

Try it:

```
mysql -uroot -p
```

```
mysql>
```

# Installations (phpmyadmin)\*

Sudo apt-get install phpmyadmi

Make phpmyadmin conf if not enabled by default install:

Include /etc/phpmyadmin/apache.conf

Enable it: a2ensite phpmyadmin.conf



# Create web service app (1)

Introduce: **apiary.io**

Let's design first !

**<http://docs.allergic.apiary.io>**

**<http://private-5cad7-allergic.apiary-mock.com/allergics/2>**

# Create web service app (2)

Laravel new Allergic

Allergic\$ composer install

Allergic\$ mv .env.example .env

Allergic\$ nano config/app.php

*Fill chipper key with running: php artisan key:generate*

Scaffold for authentication & registration user:

Allergic\$ php artisan make:auth

Allergic\$ php artisan migrate

Visit your localhost

**Welcome to laravel !**

## Create web input UI app (3)\*

Lets CRUD: <https://github.com/laralib/l5scaffold>

```
$composer require 'laralib/l5scaffold' --dev
```

```
$php artisan make:scaffold Allergic \
```

```
--schema="name:string, avoid:string, take_care:string"
```

```
$php artisan:migrate
```

*Note: If you receive a "class not found" error when running migrations, try running the composer dump-autoload command.*

# Create api controller app (4)

```
$nano app/Providers/RouteServiceProvider.php  
    'middleware' => ['api', 'auth:api'], //http basic
```

```
$ routes/api.php
```

```
Route::group(['namespace' => 'api'], function()  
{  
    Route::group(['prefix' => 'v1', 'namespace' => 'v1'], function()  
    {  
        Route::get('allergics', 'AllergicsController@index');  
    });  
});  
  
Handle Token mismatch: app/Http/Middleware/VerifyCsrfToken.php  
Skip 'api/*'
```

# How to try consuming API ?

- Use curl see [apiary.io](http://apiary.io)
- Use http requester (addons firefox)

# Deploy web service app to heroku

- My source:
- <https://github.com/arissetyawan/allergic-api.git>
- Next time :)