



EBook Gratis

APRENDIZAJE

slim

Free unaffiliated eBook created from
Stack Overflow contributors.

#slim

Tabla de contenido

Acerca de	1
Capítulo 1: Empezando con Slim	2
Observaciones.....	2
Versiones.....	2
Examples.....	2
Instalación o configuración.....	2
Primeros pasos con Slim Framework.....	4
Obtener datos de Json de la tabla de base de datos (API REST).....	4
Hola mundo ejemplo.....	5
Proyecto de esqueleto de andamio con `composer` y `@akrabats` Slim-Skeleton`.....	5
Capítulo 2: Devolver una respuesta JSON con Slim	6
Examples.....	6
Devolver una respuesta JSON con Slim.....	6
Solicitud POST con Slim 3 (API REST).....	6
Creditos	8

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [slim](#)

It is an unofficial and free slim ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official slim.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con Slim

Observaciones

Esta sección proporciona una descripción general de qué es Slim y por qué un desarrollador puede querer usarlo.

También debe mencionar los temas grandes dentro de slim y vincular a los temas relacionados. Dado que la Documentación para slim es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Versiones

Versión	Descripción	Fecha de lanzamiento
3.8.1	Este contenido relacionado con la versión delgada 3	2017-03-19

Examples

Instalación o configuración

La forma recomendada de instalar Slim framework es mediante el uso de composer.

1. Crear un directorio vacío

Este directorio contendrá todos los archivos necesarios para que se ejecute nuestra aplicación Slim. Llamamos a este directorio directorio **raíz** para que podamos abordar todos los demás archivos de aplicaciones y directorios relacionados con el directorio raíz.

```
mkdir slim-app
```

2. Instala el framework Slim y sus requerimientos usando el compositor

```
cd slim-app
composer require slim/slim "^3.0"
```

De ahora en adelante, asumimos que este es nuestro directorio de trabajo.

Después de que el compositor termine de descargar los archivos requeridos, debe haber dos archivos **composer.json** y **composer.lock** y un directorio llamado **proveedor** que incluye los archivos descargados por el compositor. Ahora estamos listos para crear nuestra aplicación. Para organizar nuestra aplicación, creamos otro directorio:

```
mkdir public
```

Llamamos a esto directorio **público** y le diremos a nuestro servidor web que sirva nuestra aplicación desde este directorio.

3. Ejemplo de aplicación "Hello World"

Para usar Slim, cree un `index.php` en el directorio público con el siguiente código:

public / index.php

```
<?php
include "../vendor/autoload.php";

$app = new \Slim\App();

$app->get('/', function ($request, $response, $args) {
    $response->getBody()->write("Hello world!");
});

$app->run();
```

4. Iniciar PHP incorporado en el servidor

Ahora podemos usar el servidor incorporado de PHP para servir nuestra aplicación:

```
php -S localhost:8080 -t public
```

y ejecute el proyecto abriendo esta dirección en un navegador web:

<http://localhost:8080>

Salida

¡Hola Mundo!

Ahora configure el servidor web para que todas las solicitudes se pasen a este archivo:

Configuración de Apache para URLs limpias (Opcional)

Esto no es obligatorio, pero se recomienda para proyectos delgados para eliminar `index.php` en la URL de la API.

Cree `.htaccess` en la misma carpeta donde se encuentra su `index.php`. El archivo debe contener el siguiente código:

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^ index.php [QSA,L]
```

Asegúrese de que su host virtual de Apache esté configurado con la opción `AllowOverride` para que las reglas de reescritura declaradas `.htaccess` puedan ser utilizadas:

Configuración de Ngnix

TBA

Primeros pasos con Slim Framework

Instalación o configuración del framework Slim

- Instalar composer
- Abrir cmd
- Vaya al directorio raíz de su carpeta de proyectos y ejecute el comando siguiente.

```
composer require delgado / delgado "^ 3.0"
```

Ahora tendrás directorio de proveedores en tu proyecto.

A continuación, cree Index.php en la carpeta raíz y agregue el siguiente código

```
<?php
use \Psr\Http\Message\ServerRequestInterface as Request;
use \Psr\Http\Message\ResponseInterface as Response;

require 'vendor/autoload.php';

$app = new \Slim\App;
$app->get('/hello/{name}', function (Request $request, Response $response) {
    $name = $request->getAttribute('name');
    $response->getBody()->write("Hello, $name");

    return $response;
});
$app->run();
```

Luego ejecute el proyecto en Localhost e intente con el siguiente comando

[http: //localhost/project-root/index.php/hello/any-thing](http://localhost/project-root/index.php/hello/any-thing)

Salida

Hola cualquier cosa

Obtener datos de Json de la tabla de base de datos (API REST)

```
use \Psr\Http\Message\ServerRequestInterface as Request;
use \Psr\Http\Message\ResponseInterface as Response;

require 'vendor/autoload.php';

$app = new \Slim\App;

$app->get('/employee/view', function ($req, $res) {
```

```

$con = new mysqli('localhost', 'USERNAME', 'PASSWORD', 'DATABASE');

$query = $con->query("SELECT * FROM employee");
while ($row = $query->fetch_assoc()) {
    $data[] = $row;
}
return $res->withJson($data);
});

$app->run();

```

Hola mundo ejemplo

```

<?php

include "vendor/autoload.php";

$app = new \Slim\App();

$app->get('/hello', function () {
    echo "Hello, world";
});

$app->run();

```

Proyecto de esqueleto de andamio con `composer` y @akrabats` Slim-Skeleton`

1. Dependiendo de si ha instalado el [compositor](#) globalmente o localmente.
 - **Localmente:** `php composer.phar create-project slim/slim-skeleton {my-app-name}`
 - **A nivel mundial:** `composer create-project slim/slim-skeleton {my-app-name}`
2. Si está ejecutando un servidor web (por ejemplo, Apache o nginx), apunte la raíz de su documento de host virtual a la nueva carpeta pública de aplicaciones, ej. `{my-app-name}/public`.
3. Asegúrese de que los `logs/` se puedan escribir en la configuración del servidor web.

También puedes ejecutar todo el shebang en modo dev con:

- **Localmente:** `php composer.phar start`
- **A nivel mundial:** `composer start`

Crédito: Rob Allen [@akrabat](#) y el equipo de SlimPHP.

PD: no tomo absolutamente ningún crédito por esto. Casi textualmente de [slimphp / Slim-Skeleton](#)

Lea Empezando con Slim en línea: <https://riptutorial.com/es/slim/topic/6878/empezando-con-slim>

Capítulo 2: Devolver una respuesta JSON con Slim

Examples

Devolver una respuesta JSON con Slim

Slim puede usar `withJson ()` que puede devolver respuestas JSON con la ruta y el método deseados

```
$app = new \Slim\App();
$app->get('/api/person', function ($request, $response, $args) {

    $payload=[];
    array_push($payload, array("name"=>"Bob" , "birth-year"=>1993));
    array_push($payload, array("name"=>"Alice", "birth-year"=>1995));

    return $response->withJson($payload,200);

});

$app->run();
```

Pruébalo con rizo

```
`curl -X GET http://127.0.0.1/api/person`
```

puede devolver una matriz en json

```
[{"name":"Bob","birth-year":1993}, {"name":"Alice","birth-year":1995}]
```

Solicitud POST con Slim 3 (API REST)

```
use \Psr\Http\Message\ServerRequestInterface as Request;
use \Psr\Http\Message\ResponseInterface as Response;

$app = new \Slim\App;

$app->post('/post/data', function (Request $request, Response $response, $arg){

    $_input = $request->getParsedBody();

    $_data_1 = $_input['name'];
    $_data_2 = $_input['email'];
    $rsp = array();

    if(!empty($_data_1) && !empty($_data_2)){

        $rsp["error"] = false;
        $rsp['message'] = "hello my name is ".$_data_1." and my email is ".$_data_2;
    }else{
```



```
    $rsp["error"] = false;
    $rsp['message'] = "you have not posted any data" ;
}

return $response
    ->withStatus(201)
    ->withJson($rsp);
});

$app->run();
```

Lea Devolver una respuesta JSON con Slim en línea:

<https://riptutorial.com/es/slim/topic/9332/devolver-una-respuesta-json-con-slim>

Creditos

S. No	Capítulos	Contributors
1	Empezando con Slim	Community , Dhruvang Gajjar , geggleto , Georgy Ivanov , martisj , Mika Tuupola , Nima
2	Devolver una respuesta JSON con Slim	Inovationist Lloyd Idris , martisj , Ramy hakam , YChi Lu