



Clase 08

API REST y SLIM

[Volver](#)

Temas:

- Introducción a REST
- Composer
- Slim Framework

REST - Representational State Transfer

Es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar **HTTP**.

REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda **HTTP**, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como **SOAP** y **XML-RPC**.

Por lo tanto **REST** es el tipo de arquitectura más natural y estándar para crear **APIs** para servicios orientados a Internet.

REST - Representational State Transfer

REST es un conjunto de convenciones para aplicaciones web y servicios web, que se centra principalmente en la manipulación de recursos a través de especificaciones **HTTP**.

Podemos decir que **REST** es una interfaz web estándar y simple que nos permite interactuar con servicios web de una manera muy cómoda.

Gracias a **REST** la web ha disfrutado de escalabilidad como resultado de una serie de diseños fundamentales clave:

Un protocolo cliente/servidor **sin estado**:

Cada mensaje **HTTP** contiene toda la información necesaria para comprender la petición.

Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes.

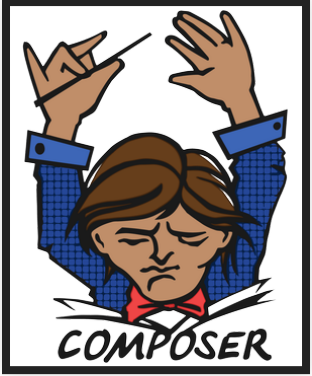
Un conjunto de operaciones **bien definidas**:

Que se aplican a todos los recursos de información: **HTTP** en sí define un conjunto pequeño de operaciones, las más importantes son **POST, GET, PUT y DELETE**.

Temas:

- Introducción a REST
- **Composer**
- Slim Framework

Composer



Composer es una herramienta para la gestión de dependencias en **PHP**.

Permite declarar las bibliotecas de las que depende tu proyecto y las administrará (instalará/actualizará) para vos.

<https://getcomposer.org/download/>

Temas:

- Introducción a REST
- Composer
- Slim Framework
 - Generalidades
 - Configurar index.php & .htaccess
 - Ruteo

Slim Framework



Slim es un framework micro de PHP que ayuda a escribir rápidamente aplicaciones Web y APIs sencillas pero poderosas.

En esencia, Slim es un despachador que recibe una solicitud HTTP, invoca una rutina de devolución de llamada apropiada y devuelve una respuesta HTTP.

<https://www.slimframework.com/docs/start/installation.html>

Temas:

- Introducción a REST
- Composer
- Slim Framework
 - Generalidades
 - Configurar index.php & .htaccess
 - Ruteo

Configurar **index.php**

En este archivo se pondrán los verbos HTTP que se recibirán y las rutas por las cuales vamos a poder acceder a ellos.

```
use Psr\Http\Message\ResponseInterface as Response;  
use Psr\Http\Message\ServerRequestInterface as Request;  
use Psr\Http\Server\RequestHandlerInterface;  
use Slim\Factory\AppFactory;  
use Slim\Routing\RouteCollectorProxy;  
use Slim\Routing\RouteContext;  
  
// Instantiate App  
$app = AppFactory::create();
```

Configurar **index.php**

Definir el conjunto de operaciones son: POST, GET, PUT y DELETE.

```
// Instantiate App
$app = AppFactory::create();
// Routes
$app->group('/usuarios', function (RouteCollectorProxy $group) {
    $group->get('[/]', \UsuarioController::class . ':TraerTodos');
    $group->get('/{usuario}', \UsuarioController::class . ':TraerUno');
    $group->post('[/]', \UsuarioController::class . ':CargarUno');
});
$app->get('[/]', function (Request $request, Response $response) {
    $response->getBody()->write("Slim Framework 4 PHP");
    return $response;
});
$app->run();
```

Configurar **index.php**

Todas la peticiones de HTTP son atendidas por este archivo y no tendría que ponerse explícitamente en la ruta de acceso

Esto esta mal :

<http://localhost:8080/miApiRest/index.php/saludo>

Esto esta bien :

<http://localhost:8080/miApiRest/saludo>

Para lograr esto se debe crear o modificar el archivo “.**HTACCESS**”

Configurar .htaccess

Los ficheros .htaccess o "ficheros de configuración distribuida", facilitan la forma de realizar cambios en la configuración de contexto de un directorio.

Un fichero, que contiene una o más directivas, se coloca en un documento específico de un directorio, y estas directivas aplican a ese directorio y todos sus subdirectorios.

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

Temas:

- Introducción a REST
- Composer
- Slim Framework
 - Generalidades
 - Configurar index.php & .htaccess
 - Ruteo

Ruteo

```
$app->get('/libros', function ($request, $response, array $args) {  
    // Obtener libros  
});  
$app->get('/libros/{id}', function ($request, $response, array $args)  
    // Obtener libros por ID  
});  
$app->post('/libros', function ($request, $response, array $args) {  
    // Crear un libro  
});  
$app->put('/libros/{id}', function ($request, $response, array $args)  
    // Actualizar libro por ID  
});  
$app->delete('/libros/{id}', function ($request, $response, array $a  
    // Borrar libros por ID
```


Ruteo

Más ejemplos de ruteo:

<https://www.slimframework.com/docs/v4/objects/routing.html>

Dudas ó consultas