

Programación III

API REST - SLIM

Clase 10

Maximiliano Neiner
Villegas Octavio
Rampi Mario

Temas a Tratar

- ✖ Definición de Middleware en PSR7
- ✖ Middleware en Slim

Temas a Tratar

 Definición de Middleware en PSR7

 Middleware en Slim

Middleware - PSR7 (1/3)

- ❖ Middleware es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos.
- ❖ Técnicamente hablando, un middleware es una función que puede retornar un *callable*, que acepta tres argumentos:
 - `\Psr\Http\Message\ServerRequestInterface` - El objeto de solicitud PSR7
 - `\Psr\Http\Message\ResponseInterface` - El objeto de respuesta PSR7
 - *callable* - El siguiente middleware callable

Middleware - PSR7 (2/3)

✖ PHP Standards Recommendations

✖ Documentación oficial

- <http://www.php-fig.org/psr/psr-7/>

✖ Un mensaje HTTP es una petición de un cliente a un servidor o una respuesta de un servidor a un cliente.

✖ Esta especificación define interfaces para los mensajes HTTP:

- Psr\Http\Message\RequestInterface
- Psr\Http\Message\ResponseInterface

Middleware - PSR7 (3/3)

- ✚ El único requisito es que un middleware DEBE devolver una instancia de \Psr\Http\Message\ResponseInterface.
- ✚ Cada middleware DEBERÍA invocar al siguiente middleware y pasarle los objetos Request y Response como argumentos.

Temas a Tratar

 Definición de Middleware en PSR7

 Middleware en Slim

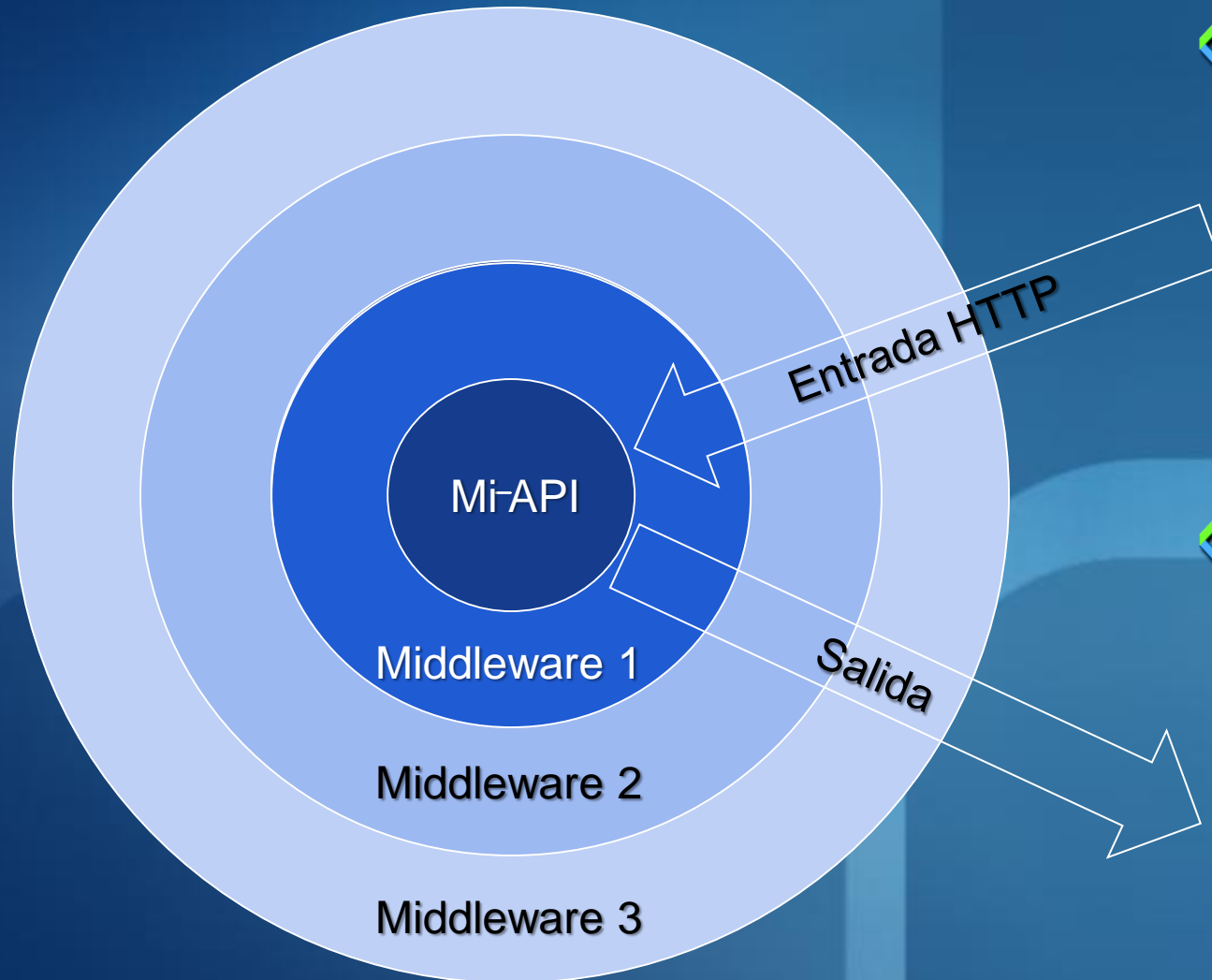
Middleware en Slim (1/3)

- ✚ En Slim, podemos ejecutar código antes y después de una llamada a nuestra apiRest, para poder manipular los objetos Request y Response como mejor nos parezca.
- ✚ Esto se realiza por medio de un middleware.
- ✚ Posibles usos:
 - Proteger la aplicación de la falsificación de solicitudes cruzadas.
 - Autenticar las solicitudes antes de ejecutar su aplicación.
 - Etc.

Middleware en Slim (2/3)

- ✚ Slim añade middleware como capas concéntricas que rodean su aplicación principal.
- ✚ Cada nueva capa de middleware rodea cualquier capa de middleware existente.
- ✚ La estructura concéntrica se expande hacia afuera a medida que se añaden capas de middleware adicionales.
- ✚ La última capa de middleware agregada es la primera en ser ejecutada.

Middleware en Slim (3/3)



Entrada:

- 3
- 2
- 1
- Mi API



Salida

- Mi API
- 1
- 2
- 3

Temas a Tratar

 Definición de Middleware en PSR7

 Middleware en Slim

- Funciones middleware
- Middleware Route
- Middleware Group
- Middleware con P00
- Utilidades

Funciones Middleware

- ✚ Para definir la función se debe respetar la firma de la estandarización PSR7

```
//*****  
//FUNCION MIDDLEWARE  
//*****  
$mwUno = function($request, $response, $next) {  
    //EJECUTO ACCIONES ANTES DE INVOCAR A LA API  
    $response->getBody()->write('Antes de ejecutar MiddleWare UNO...<br>');  
    //INVOCO A LA API  
    $response = $next($request, $response);  
    //EJECUTO ACCIONES DESPUES DE INVOCAR A LA API  
    $response->getBody()->write('<br>...Despues de ejecutar MiddleWare UNO');  
  
    return $response;  
};
```

- ✚ Para agregarla, se hace con el método de instancia add(), de application, route o group.

```
//*****  
//AGREGO MIDDLEWARE PARA TODOS LOS VERBOS  
//*****  
$app->add($mwUno);
```

Temas a Tratar

 Definición de Middleware en PSR7

 Middleware en Slim

- Funciones middleware
- **Middleware Route**
- Middleware Group
- Middleware con P00
- Utilidades

Middleware - Route (1/2)

- ✚ El 'middleware de ruta' solo se invoca si su ruta coincide con el método de solicitud HTTP actual y la URI.
- ✚ Este middleware es ejecutado inmediatamente después de invocar cualquiera de los métodos de enrutamiento de la aplicación Slim (por ejemplo, get o post).
- ✚ El middleware se agrega con el método `add()` de la instancia `Route`.

Middleware - Route (2/2)

```
//*****  
//AGREGO ROUTE MIDDLEWARE, SOLO PARA PUT  
//*****  
$app->put('/param/', function (Request $request, Response $response) {  
    $response->getBody()->write("API => PUT");  
    return $response;  
  
})->add(function ($request, $response, $next) {  
    //EJECUTO ACCIONES ANTES DE INVOCAR A LA API  
    $response->getBody()->write('Antes de ejecutar Middleware DOS...<br>');  
    //INVOCO A LA API  
    $response = $next($request, $response);  
    //EJECUTO ACCIONES DESPUES DE INVOCAR A LA API  
    $response->getBody()->write('<br>...Despues de ejecutar Middleware DOS');  
  
    return $response;  
});
```


Temas a Tratar

 Definición de Middleware en PSR7

 Middleware en Slim

- Funciones middleware
- Middleware Route
- **Middleware Group**
- Middleware con P00
- Utilidades

Middleware - Group (1/2)

- ✖ Los middlewares, además de poder agregarse a la aplicación y a las rutas, pueden también agregarse a los métodos de grupos, como a las rutas individuales internas.
- ✖ El 'middleware de grupo' solo se invoca si su ruta coincide con uno de los métodos de solicitud HTTP definidos y los URI del grupo.
- ✖ Para agregar el middleware dentro de un *callback* individual o para todo el grupo, se establecerá mediante el método de instancia `add()`.

Middleware - Group (2/2)

```
//*****  
//AGREGO GROUP MIDDLEWARE  
//*****  
$app->group('/grupo', function () {  
  
    $this->get('/hoy', function (Request $request, Response $response) {  
        return $response->getBody()->write(date('Y-m-d'));  
    });  
  
    $this->get('/hora', function (Request $request, Response $response) {  
        return $response->getBody()->write(date('H:i:s'));  
    }->add(function($request, $response, $next) {  
        //EJECUTO ACCIONES ANTES DE INVOCAR A LA API  
        $response->getBody()->write('Antes de ejecutar Middleware CUATRO...<br>');  
        //INVOCO A LA API  
        $response = $next($request, $response);  
        //EJECUTO ACCIONES DESPUES DE INVOCAR A LA API  
        $response->getBody()->write('<br>...Despues de ejecutar Middleware CUATRO');  
  
        return $response;  
    }));  
  
    }->add(function ($request, $response, $next) {  
        //EJECUTO ACCIONES ANTES DE INVOCAR A LA API  
        $response->getBody()->write('Antes de ejecutar Middleware TRES...<br>');  
        //INVOCO A LA API  
        $response = $next($request, $response);  
        //EJECUTO ACCIONES DESPUES DE INVOCAR A LA API  
        $response->getBody()->write('<br>...Despues de ejecutar Middleware TRES');  
  
        return $response;  
    }));  
});
```

EJERCICIO

Temas a Tratar

 Definición de Middleware en PSR7

 Middleware en Slim

- Funciones middleware
- Middleware Route
- Middleware Group
- **Middleware con POO**
- Utilidades

Middleware con POO (1/2)



Se pueden crear clases con los métodos que puedan ser usados como middleware

```
class MiClase
{
    public function MostrarInstancia($request, $response, $next){
        $response->getBody()->write('<br>Desde método de instancia<br>');
        return $next($request, $response);
    }

    public static function MostrarEstatico($request, $response, $next){
        $response->getBody()->write('<br>Desde método estático<br>');
        return $next($request, $response);
    }
}
```

Middleware con P00 (2/2)



Lo agrego a la api, rutas o grupos.

```
$app->group('/grupo/P00', function () {  
  
    //EN LA RAIZ DEL GRUPO  
    $this->get('/', function (Request $request, Response $response) {  
        return $response->getBody()->write("-GET- En el raíz del grupo...");  
    });  
  
    $this->post('/', function (Request $request, Response $response) {  
        return $response->getBody()->write("-POST- En el raíz del grupo...");  
    });  
    //-----  
  
    $this->get('/hoy', function (Request $request, Response $response) {  
        return $response->getBody()->write(date('Y-m-d'));  
    });  
  
    $this->get('/hora', function (Request $request, Response $response) {  
  
        date_default_timezone_set('America/Argentina/Buenos_Aires'); //Australia/Melbourne  
  
        return $response->getBody()->write(date('H:i:s'));  
  
    })->add(\MiClase::class . ":MostrarInstancia");  
  
    })->add(\MiClase::class . "::MostrarEstatico");  
});
```


Temas a Tratar



Definición de Middleware en PSR7



Middleware en Slim

- Funciones middleware
- Middleware Route
- Middleware Group
- Middleware con P00
- Utilidades

Funciones Middleware



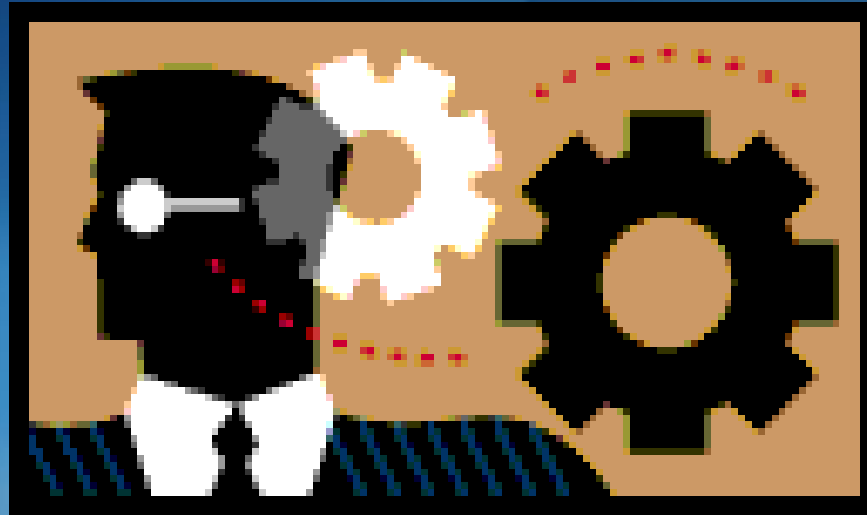
Posibles utilidades:

- Habilitar CORS (Cross Origin Resource Sharing)

```
/**
 * Habilitar CORS para todos los verbos
 */
app->add(function ($request, $response, $next) {
    $response = $next($request, $response);
    return $response
        ->withHeader('Access-Control-Allow-Origin', 'http://localhost')
        ->withHeader('Access-Control-Allow-Headers', 'X-Requested-With, Content-Type, Accept, Origin, Authorization')
        ->withHeader('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE, OPTIONS');
});
```

- Manipular archivos, modificar tamaño de imágenes.
- Encriptar, firewall, validaciones.
- Verificador de credenciales (JWT)
- Información del cliente: Geolocalización, dispositivos, IPs.
- Querés MAS??

<https://github.com/oscarotero/psr7-middlewares>



Ejercitación

Ejercicios

- ✖ Hacer un middleware de aplicación que tome usuario y contraseña y verifique en BD.
- ✖ Hacer middleware de grupo, solo para post, que permita agregar un nuevo usuario, si el perfil es 'admin'.
- ✖ Hacer middleware de grupo, solo para delete, que permita borrar un usuario, si el perfil es 'super_admin'.
- ✖ Hacer middleware de ruta, solo para put y get, que tome el tiempo de demora entre que entra y sale la petición.