

PRUEBA SERVIFORM



Sergi Sanz Carreres

Instalaciones Requeridas

1.- Instalación de PHP 7.4.13: <https://windows.php.net/download#php-7.4>

2.- Instalación de Composer: <https://getcomposer.org/download/>

3.- Instalación de **Symfony**: <https://symfony.com/download>

4.- MySQL: <https://dev.mysql.com/downloads/>

Pasos necesarios para su utilización

1.- **Actualizar librerías utilizando composer:** composer u

2.- En el archivo '**php.ini**' que se encuentra en la carpeta donde esta instalado el php, asegurarnos de tener descomentado la extensión '**extension=pdo_mysql**'

3.- En el documento '**.env**' del proyecto modificar los datos correspondientes a la configuración de MySQL (lo encontramos en una línea tal que esta: **# DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion=5.7"**) por los datos correspondientes a los definidos en la instalación de MySQL:

- **mysql:** Motor de base de datos a utilizar
- **db_user:** Usuario
- **db_password:** Contraseña a utilizar
- **db_name:** Nombre de la base de datos

4.- Como la Base de datos ya se encuentra creada en nuestro proyecto, simplemente habrá que utilizar el comando '**php bin/console doctrine:migrations:migrate**' para que cargue la estructura de la base de datos que se ha desarrollado. Posteriormente, nos conectaremos a MySQL para mediante el comando '**source ./202011281347.sql**' poblar la base de datos con los datos para poder ejemplificar el programa desarrollado. Si hacemos un select de las distintas tablas obtendremos los siguientes resultados cerciorando que se han incluido datos en las tablas creadas.

```
mysql> select * from alumno;
+-----+-----+-----+
| id | nombre | apellidos |
+-----+-----+-----+
| 1 | Sergi | Sanz |
| 2 | Rene | Descartes |
| 3 | Alonso | Ferreras |
| 4 | Aina | Sanz |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from curso;
+-----+-----+
| id | nombre |
+-----+-----+
| 1 | quimica |
| 2 | fisica |
| 3 | matematicas |
+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from alumno_curso;
+-----+-----+
| alumno_id | curso_id |
+-----+-----+
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 2 | 2 |
| 2 | 3 |
| 3 | 1 |
| 3 | 3 |
| 4 | 2 |
+-----+-----+
8 rows in set (0.00 sec)
```

5.- Ahora será necesario inicializar el servidor en symfony utilizando el comando **'symfony server:start'** , obteniendo un resultado como el siguiente:

```
C:\Windows\System32\cmd.exe - symfony server:start

C:\Users\Sergi\OneDrive - UPV (1)\Escritorio\Prueba Servinform>symfony server:start

[WARNING] run "symfony.exe server:ca:install" first if you want to run the web server with TLS support, or use "--no-tls" to avoid this warning

Tailing Web Server log file (C:\Users\Sergi\.symfony\log\062a582fab6eaa42a0ceb4ca69ea6571e717bab0.log)
Tailing PHP-CGI log file (C:\Users\Sergi\.symfony\log\062a582fab6eaa42a0ceb4ca69ea6571e717bab0\79ca75f9e90b4126a5955a33ea6a41ec5e854698.log)
Tailing PHP-CGI log file (C:\Users\Sergi\.symfony\log\062a582fab6eaa42a0ceb4ca69ea6571e717bab0\79ca75f9e90b4126a5955a33ea6a41ec5e854698.log)

[OK] Web server listening

The Web server is using PHP CGI 7.4.13

http://127.0.0.1:8000

[Web Server ] Nov 29 14:51:52 |INFO | PHP listening path="C:\\Program Files\\PHP\\php-cgi.exe" php="7.4.13" port=51161
```

6.- Finalmente, en otra ventana de terminal, utilizamos el comando **'php Test.php'** para ejecutar el script que hace una petición al WebService haciendo uso de CURL. De forma que le indicamos el ID del curso, devuelve un JSON con los nombres y apellidos de todos los alumnos de ese curso. En caso de que no exista ese ID o que le introduzcamos algún carácter no numérico devolverá "Se ha producido un error".

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.18363.1198]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sergi\OneDrive - UPV (1)\Escritorio\Prueba Servinform>php Test.php
Indique el ID del curso: 1
[{"nombre":"Sergi","apellidos":"Sanz"}, {"nombre":"Alonso","apellidos":"Ferrerias"}]
C:\Users\Sergi\OneDrive - UPV (1)\Escritorio\Prueba Servinform>php Test.php
Indique el ID del curso: 2
[{"nombre":"Sergi","apellidos":"Sanz"}, {"nombre":"Rene","apellidos":"Descartes"}, {"nombre":"Aina","apellidos":"Sanz"}]
C:\Users\Sergi\OneDrive - UPV (1)\Escritorio\Prueba Servinform>php Test.php
Indique el ID del curso: 3
[{"nombre":"Sergi","apellidos":"Sanz"}, {"nombre":"Rene","apellidos":"Descartes"}, {"nombre":"Alonso","apellidos":"Ferrerias"}]
C:\Users\Sergi\OneDrive - UPV (1)\Escritorio\Prueba Servinform>php Test.php
Indique el ID del curso: 500000000000000000
Se ha producido un error
```

Explicación SCRIPT

Curso Controller:

El archivo '**Curso Controller.php**' correspondo al Webservice en Php que tiene como objetivo recibir un id de curso por POST y devolver un JSON con los nombres y apellidos de todos los alumnos vinculados al curso.

```
class CursoController extends AbstractController
{
    /**
     * @Route("/curso/{idCurso}", name="curso_alumnos", methods={"POST"})
     * Exceptions: {
     *   500: El curso no existe
     * }
     * Description: Devuelve un listado de alumnos por nombre y apellido de un curso
     * obtenido por una id
     * Return: Json
     */
    public function cursoAlumno(int $idCurso): Response
    {
        $curso = $this->getDoctrine()
            ->getRepository(Curso::class)
            ->find($idCurso);

        if (!isset($curso)) {
            throw new \Exception('El curso no existe');
        }

        $alumnos = $curso->getAlumnos();

        $data = [];
        foreach ($alumnos as $alumno) {
            $data[] = [
                'nombre' => $alumno->getNombre(),
                'apellidos' => $alumno->getApellidos(),
            ];
        }

        return $this->json($data, $status = 200, $headers = [], $context = []);
    }
}
```

Donde en la clase 'CursoController' se ha desarrollado una función 'cursoAlumno' que recibe como parámetro el identificador del Curso. Posteriormente, se utiliza la librería Doctrine de Symfony para interactuar con la base de datos, donde en ella accedemos a la tabla curso y buscamos el identificador de ese curso.

Luego utilizo isset para comprobar si el identificador está en null y en caso de estarlo lanzar la excepción.

A continuación, creo la variable alumnos, donde se almacena los alumnos consultados de curso que corresponden con el identificador requerido que cumple con las condiciones.

Finalmente, recorremos los datos de cada alumno y los almacenamos en un array de nombre 'data' que devolvemos finalmente como un json.

Test:

El archivo '**Test.php**' corresponde al Desarrollo de un script en PHP con una petición al Webservice haciendo uso de CURL.

```
<?php

echo "Indique el ID del curso: ";
$idCurso = rtrim(fgets(STDIN));

$parametros = array(
    CURLOPT_URL => 'http://localhost:8000/curso/' . $idCurso,
    CURLOPT_POST => true,
    CURLOPT_RETURNTRANSFER => true
);

$conexion = curl_init();
curl_setopt_array($conexion, $parametros);
$respuesta = curl_exec($conexion);

if (curl_getinfo($conexion, CURLINFO_HTTP_CODE) === 200) {
    echo $respuesta;
} else {
    echo 'Se ha producido un error';
}

curl_close($conexion);

?>
```

Donde mediante el uso de echo solicitamos al usuario el id del curso y posteriormente lo almacenamos en la variable 'idCurso'. A continuación, en la variable parámetros almacenamos un array con 'CURLOPT_URL,CURLOPT_POST,CURLOPT_RETURNTRANSFER' y iniciamos el 'curl_init()' que se almacena en la variable conexión, donde al ejecutarlo usando 'curl_exec' estamos haciendo una llamada HTTP asociada a la función 'CursoAlumno'. En caso de que esa llamada sea satisfactoria devolveremos la respuesta que en este caso será un json de todos los

alumnos que formen parte de ese curso, mientras que en caso contrario indicará un error. Finalmente, se cerrará la conexión del curl.