

<b>NOMBRE Y APELLIDOS:</b>			<b>FECHA: 02/10/2023</b>		
<b>DOCENTE: MANUEL MACÍAS PÉREZ</b>			<b>NOTA:</b>		
<b>(IFCD0210) DESARROLLO DE APLICACIONES CON TECNOLOGÍAS WEB</b>			<b>N.º CURSO: 22-35/008902</b>		
MF:	0492	UNIDADES DE APRENDIZAJE A LAS QUE RESPONDE:	UA1	Duración:	2 hrs.
UF:	1846				
PRÁCTICA N.º:	Actividad 3				
<b>DENOMINACIÓN: API-REST</b>					
<p><b><u>DESCRIPCIÓN</u></b></p> <p>1.- En un supuesto practico, hay que integrar una nueva tabla a una base de datos en sql.</p> <p>Una empresa necesita integrar una nueva categoría de servicios a su base de datos. El nombre de la tabla será "AdminServices" y los campos serán los siguientes: Id, nombre y descripción.</p> <p>El alumno tendrá que realizar dicha integración en la base de datos y la configuración de la API(servidor) para poder probar su funcionamiento desde la aplicación Postman. GET, POST, PUT, DELETE</p> <p>La práctica se realizará de manera individual.</p> <p>Adjuntar pantallazo de las pruebas realizadas y adjuntar los código en el servidor.</p>					

POST

localhost:3000/adminservices/

Send

Status: 200 OK

Size: 44 Bytes

Time: 2.15 s

Query

Headers<sup>2</sup>

Auth

Body<sup>1</sup>

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1 {
2   "name": "hola"
3 }

```

Response

Headers<sup>2</sup>

Cookies

Results

Docs

```

1 {
2   "msg": "¡El admin fue agregado con éxito!"
3 }

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

PORTS

OUTPUT

TERMINAL

```

> nodemon ./src/app.ts

[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: ts,json
[nodemon] starting `ts-node ./src/app.ts`
La conexión con la Base de Datos es correcta
Aquí conectada por el puerto 3000
Executing (default): SELECT 1+1 AS result
Executing (default): INSERT INTO `Adminservices` (`id`,`name`) VALUES (DEFAULT,?);

```

GET

localhost:3000/adminservices/

Send

Status: 200 OK

Size: 48 Bytes

Time: 46 ms

Query

Headers<sup>2</sup>

Auth

Body<sup>3</sup>

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1 {
2   "name": "hola2"
3 }

```

Response

Headers<sup>2</sup>

Cookies

Results

Docs

```

1 [
2   {
3     "id": 1,
4     "name": "hola"
5   },
6   {
7     "id": 2,
8     "name": "hola2"
9   }
10 ]

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

PORTS

OUTPUT

TERMINAL

```

[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: ts,json
[nodemon] starting `ts-node ./src/app.ts`
La conexión con la Base de Datos es correcta
Aquí conectada por el puerto 3000
Executing (default): SELECT 1+1 AS result
Executing (default): INSERT INTO `Adminservices` (`id`,`name`) VALUES (DEFAULT,?);
Executing (default): SELECT `id`,`name` FROM `Adminservices` AS `Adminservices`;
Executing (default): INSERT INTO `Adminservices` (`id`,`name`) VALUES (DEFAULT,?);
Executing (default): INSERT INTO `Adminservices` (`id`,`name`) VALUES (DEFAULT,?);
Executing (default): SELECT `id`,`name` FROM `Adminservices` AS `Adminservices`;

```

PUT localhost:3000/adminservices/1 Send

Status: 200 OK Size: 44 Bytes Time: 121 s

Query Headers Auth Body Tests Pre Run

JSON XML Text Form Form-encoded GraphQL Binary

JSON Content

```
1 {
2   "name": "holas"
3 }
```

Response

```
1 {
2   "msg": "El admin fue actualizado con éxito"
3 }
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS

OUTPUT

```
[nodemon] watching extensions: ts,json
[nodemon] starting 'ts-node ./src/app.ts'
La conexión con la Base de Datos es correcta
Api conectada por el puerto 3000
Executing (default): SELECT 1+1 AS result
Executing (default): INSERT INTO `Adminservices` (`id`,`name`) VALUES (DEFAULT,?);
Executing (default): SELECT `id`,`name` FROM `Adminservices` AS `Adminservices`;
Executing (default): INSERT INTO `Adminservices` (`id`,`name`) VALUES (DEFAULT,?);
Executing (default): SELECT `id`,`name` FROM `Adminservices` AS `Adminservices`;
Executing (default): SELECT `id`,`name` FROM `Adminservices` AS `Adminservices` WHERE `Adminservices`.`id` = '1';
Executing (default): UPDATE `Adminservices` SET `name`=? WHERE `id` = ?
```

DELETE localhost:3000/adminservices/2 Send

Status: 200 OK Size: 42 Bytes Time: 365 ms

Query Headers Auth Body Tests Pre Run

JSON XML Text Form Form-encoded GraphQL Binary

JSON Content

```
1 {
2   "name": "holas"
3 }
```

Response

```
1 {
2   "msg": "El admin fue eliminado con éxito"
3 }
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS

OUTPUT

```
La conexión con la Base de Datos es correcta
Api conectada por el puerto 3000
Executing (default): SELECT 1+1 AS result
Executing (default): INSERT INTO `Adminservices` (`id`,`name`) VALUES (DEFAULT,?);
Executing (default): SELECT `id`,`name` FROM `Adminservices` AS `Adminservices`;
Executing (default): INSERT INTO `Adminservices` (`id`,`name`) VALUES (DEFAULT,?);
Executing (default): SELECT `id`,`name` FROM `Adminservices` AS `Adminservices`;
Executing (default): SELECT `id`,`name` FROM `Adminservices` AS `Adminservices` WHERE `Adminservices`.`id` = '1';
Executing (default): UPDATE `Adminservices` SET `name`=? WHERE `id` = ?
Executing (default): SELECT `id`,`name` FROM `Adminservices` AS `Adminservices` WHERE `Adminservices`.`id` = '2';
Executing (default): DELETE FROM `Adminservices` WHERE `id` = 2
```

### MEDIOS PARA SU REALIZACIÓN

- Equipo informático.
- Aplicación Visual Studio instalada en el equipo.
- Navegadores actualizados