

NODE JS

Clase 29 – Proyecto NodeJS - MySQL



adn
POTENCIÁ TU FUTURO



Proyecto NodeJS- MySQL

01

... Crear Base de Datos



Crear base de datos bancoargentina.

01

... Crear Taba



Crear tabla Usuarios con los siguientes campos.

01

... Campos Usuarios

- id
- user
- password



01

... Campos Usuarios

<input type="checkbox"/>	1	id	 int(11)	No	Ninguna	AUTO_INCREMENT	 Cambiar	 Eliminar	 Más
<input type="checkbox"/>	2	user	varchar(20) utf8mb4_general_ci	No	Ninguna		 Cambiar	 Eliminar	 Más
<input type="checkbox"/>	3	password	varchar(20) utf8mb4_general_ci	No	Ninguna		 Cambiar	 Eliminar	 Más

01

... Crear Taba



Crear tabla cuentabancaria con los siguientes
campos.

01

... Campos Cuentabancaria

- . id
- . cuenta
- . cbu
- . dni
- . nombre
- . apellido
- . pesos
- . dolares

01

... Campos Cuentabancaria

<input type="checkbox"/>	1	id	 int(11)	No	Ninguna	AUTO_INCREMENT	 Cambiar	 Eliminar	 Más
<input type="checkbox"/>	2	cuenta	varchar(50) utf8mb4_general_ci	No	Ninguna		 Cambiar	 Eliminar	 Más
<input type="checkbox"/>	3	cbu	varchar(50) utf8mb4_general_ci	No	Ninguna		 Cambiar	 Eliminar	 Más
<input type="checkbox"/>	4	dni	varchar(20) utf8mb4_general_ci	No	Ninguna		 Cambiar	 Eliminar	 Más
<input type="checkbox"/>	5	nombre	varchar(50) utf8mb4_general_ci	No	Ninguna		 Cambiar	 Eliminar	 Más
<input type="checkbox"/>	6	apellido	varchar(50) utf8mb4_general_ci	No	Ninguna		 Cambiar	 Eliminar	 Más
<input type="checkbox"/>	7	pesos	int(11)	No	Ninguna		 Cambiar	 Eliminar	 Más
<input type="checkbox"/>	8	dolares	int(11)	No	Ninguna		 Cambiar	 Eliminar	 Más

Requerimientos de API



02

... Crear API

Crear API, que pueda hacer consultas a usuarios y a cuentas bancarias



... **Crear API**

- * Requerimientos
 - Obtener Usuarios
 - Crear Usuarios
 - Eliminar Usuarios
 - Obtener Cuentas Bancarias
 - Crear Cuentas Bancarias
 - Eliminar Cuentas Bancarias



... Crear API

Requerimientos Opcionales

- Editar Usuarios
- Editar Cuentas Bancarias



Desarrollo de API

03

... Construcción de API



```
C:\>mkdir proyectoNodeJS
```

03

... Construcción de API

```
› PS C:\proyectoNodeJS> npm i express mysql2
```

03

... Construcción de API

```
› PS C:\proyectoNodeJS> npm i express mysql2
```

03

... Server

```
const mysql = require('mysql2/promise');
const express = require('express');
const app = express();
app.use(express.json());
app.listen(4000, () => {
  console.log('Server on port 4000');
})
```

03

... Conexión con DB

```
const db = mysql.createPool({  
  host: 'localhost',  
  user: 'root',  
  password: '',  
  database: 'bancoargentina'  
});
```

A black and white photograph showing four people in an office environment. Two men and two women are seated at a long desk, each working on a laptop. Behind them are tall, filled bookshelves. The scene is lit from above, creating a professional atmosphere.

Ruta y Control de Peticiones

04

... Get Users

```
// Get Users
app.get('/users', async (req, res) => {
  try {
    const [rows] = await db.query('SELECT * FROM users');
    res.json(rows);
  } catch (error) {
    console.error('Error en la consulta de usuarios:', error.message);
    res.status(500).json({ error: 'Error interno del servidor' });
  }
});
```

04

... Delete Users

```
// Delete Cuentas Bancarias
app.delete('/cuentabancaria/:id', async ( req ,res ) => {
  try {
    const { id } = req.params;
    const [ rows ] = await db.query('DELETE FROM cuentabancaria WHERE id=?',[id]);
    res.json(rows);
  } catch (error) {
    res.status(500).json({ error: 'Error interno del servidor' });
  }
})
```

04

... Get Cuentas Bancarias

```
// Get Cuentas Bancarias
app.get('/cuentabancaria', async (req, res) => {
  try {
    const [rows] = await db.query('SELECT * FROM cuentabancaria');
    res.json(rows);
  } catch (error) {
    console.error('Error en la consulta de usuarios:', error.message);
    res.status(500).json({ error: 'Error interno del servidor' });
  }
});
```

04

... Delete Cuentas Bancarias

```
// Delete Cuentas Bancarias
app.delete('/cuentabancaria/:id', async ( req ,res ) => {
  try {
    const { id } = req.params;
    const [ rows ] = await db.query('DELETE FROM cuentabancaria WHERE id=?',[id]);
    res.json(rows);
  } catch (error) {
    res.status(500).json({ error: 'Error interno del servidor' });
  }
})
```

04

... Post Cuentas Bancarias

```
// Delete Cuentas Bancarias
app.delete('/cuentabancaria/:id', async ( req ,res ) => {
  try {
    const { id } = req.params;
    const [ rows ] = await db.query('DELETE FROM cuentabancaria WHERE id=?',[id]);
    res.json(rows);
  } catch (error) {
    res.status(500).json({ error: 'Error interno del servidor' });
  }
})
```

Prueba de API

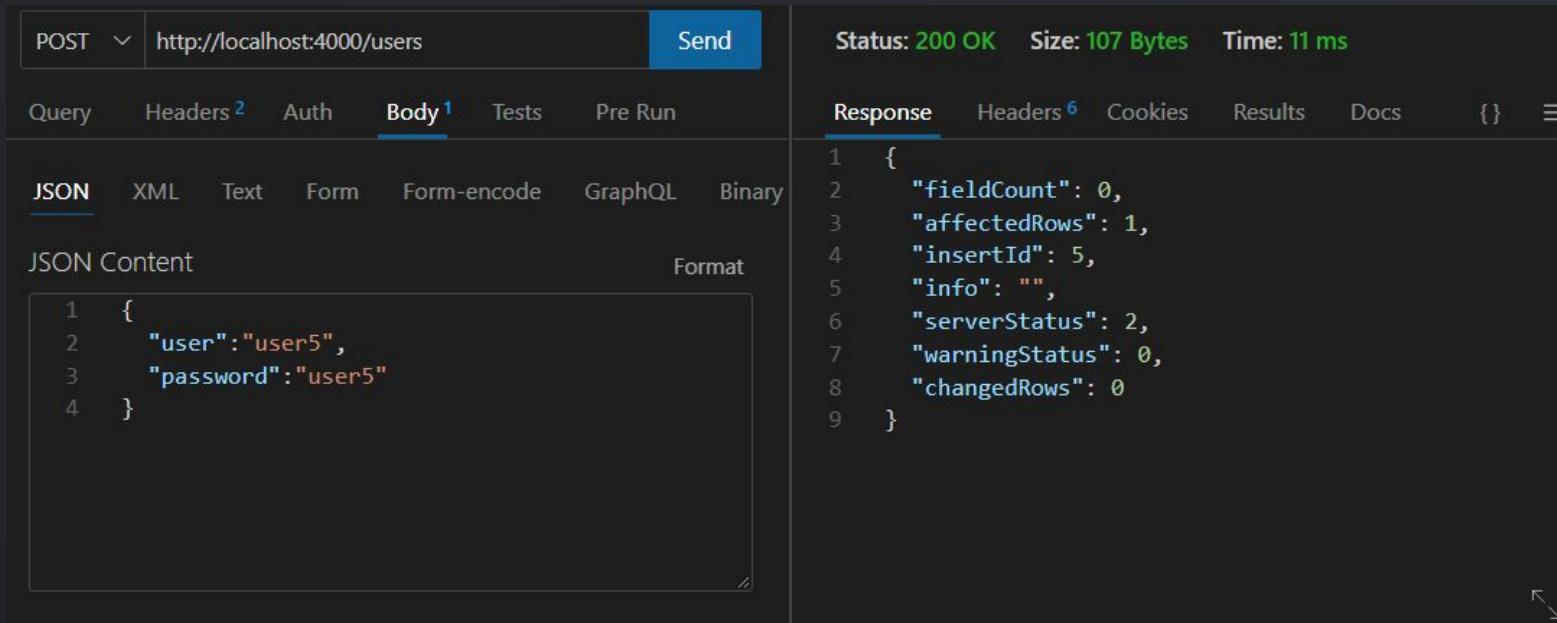
05

... Get Users

GET	▼	http://localhost:4000/users	Send			
Query	Headers 2	Auth	Body 1	Tests	Pre Run	Status: 200 OK Size: 132 Bytes Time: 8 ms
						<u>Response</u> Headers 6 Cookies Results Docs { } ⌂
						<pre>1 [2 { 3 "id": 2, 4 "user": "user2", 5 "password": "user2" 6 }, 7 { 8 "id": 3, 9 "user": "Silvio", 10 "password": "Chavez" 11 }, 12 { 13 "id": 4, 14 "user": "user3", 15 "password": "user3"</pre>

05

... Post Users



POST http://localhost:4000/users

Query Headers ² Auth Body ¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {  
2   "user": "user5",  
3   "password": "user5"  
4 }
```

Status: 200 OK Size: 107 Bytes Time: 11 ms

Response Headers ⁶ Cookies Results Docs { } 

```
1 {  
2   "fieldCount": 0,  
3   "affectedRows": 1,  
4   "insertId": 5,  
5   "info": "",  
6   "serverStatus": 2,  
7   "warningStatus": 0,  
8   "changedRows": 0  
9 }
```

05

... Delete Users

DELETE http://localhost:4000/users/3

Query Headers ² Auth Body ¹ Tests Pre Run

Query Parameters

<input type="checkbox"/> parameter	value
------------------------------------	-------

Status: 200 OK Size: 107 Bytes Time: 11 ms

Response Headers ⁶ Cookies Results Docs { } 

```
1  {
2    "fieldCount": 0,
3    "affectedRows": 1,
4    "insertId": 0,
5    "info": "",
6    "serverStatus": 2,
7    "warningStatus": 0,
8    "changedRows": 0
9 }
```



05

... Get cuentabancaria

GET http://localhost:4000/cuentabancaria

Query Headers 2 Auth Body 1 Tests Pre Run

Query Parameters

<input type="checkbox"/> parameter	value

Status: 200 OK Size: 568 Bytes Time: 17 ms

Response Headers 6 Cookies Results Docs { } ≡

```
1  [
2   {
3     "id": 2,
4     "cuenta": "34323432432",
5     "cbu": "000100001010121366",
6     "dni": "34453453",
7     "nombre": "Maria",
8     "apellido": "Muller",
9     "pesos": 1000000,
10    "dolares": 24000
11  },
12  {
13    "id": 3,
14    "cuenta": "54344343",
15    "cbu": "012131231232312321",
```

05

... Post cuentabancaria

POST http://localhost:4000/cuentabancaria

Status: 200 OK Size: 107 Bytes Time: 17 ms

Query	Headers <small>2</small>	Auth	Body <small>1</small>	Tests	Pre Run	Response	Headers <small>6</small>	Cookies	Results	Docs	{}	=	
<input checked="" type="radio" value="JSON"/> JSON	<input type="radio" value="XML"/> XML	<input type="radio" value="Text"/> Text	<input type="radio" value="Form"/> Form	<input type="radio" value="Form-encode"/> Form-encode	<input type="radio" value="GraphQL"/> GraphQL	<input type="radio" value="Binary"/> Binary	<pre>1 { 2 "cuenta": "767344343", 3 "cbu": "345341231232312321", 4 "dni": "6242324", 5 "nombre": "Ximena", 6 "apellido": "Xitla", 7 "pesos": "30000", 8 "dolares": "20000" 9 }</pre>	<pre>1 { 2 "fieldCount": 0, 3 "affectedRows": 1, 4 "insertId": 5, 5 "info": "", 6 "serverStatus": 2, 7 "warningStatus": 0, 8 "changedRows": 0 9 }</pre>					

JSON Content Format

```
1 {  
2   "cuenta": "767344343",  
3   "cbu": "345341231232312321",  
4   "dni": "6242324",  
5   "nombre": "Ximena",  
6   "apellido": "Xitla",  
7   "pesos": "30000",  
8   "dolares": "20000"  
9 }
```

05

... Delete cuentabancaria

DELETE http://localhost:4000/cuentabancaria/3

Query Headers ² Auth Body ¹ Tests Pre Run

Query Parameters

<input type="checkbox"/> parameter	value

Status: 200 OK Size: 107 Bytes Time: 12 ms

Response Headers ⁶ Cookies Results Docs { } ≡

```
1  {
2    "fieldCount": 0,
3    "affectedRows": 1,
4    "insertId": 0,
5    "info": "",
6    "serverStatus": 2,
7    "warningStatus": 0,
8    "changedRows": 0
9 }
```

¡Muchas gracias!

