

mysql2

Índice

- [Índice](#)
- [Introducción](#)
- [Instalación](#)
- [Conexión](#)
- [Consultas \(Queries\)](#)
 - [Consultas sin parámetros](#)
 - [Consultas con parámetros](#)

Introducción

`mysql2` es una librería de Node.js que permite realizar una conexión a una base de datos. En este documento utilizaremos en específico la parte de la librería que utiliza promesas para manejar la asincronía (`mysql2/promise`).

Instalación

Para instalar `mysql2` deberemos ejecutar el siguiente comando (dentro de un proyecto de Node.js):

```
npm install mysql2
```

Conexión

Para conectarnos a la base de datos deberemos añadir el siguiente código:

```
import mysql from "mysql2/promise";

const connection = await mysql.createConnection({
  host: "<host>",
  user: "<usuario>",
  password: "<contraseña>",
  database: "<nombre_de_la_db>",
});
```

En `connection` tendremos un objeto que nos permitirá hacer consultas a la base de datos.

Aclaración: (recordar que para poder importar las librerías con `import` y para poder utilizar `await` fuera de una función deben utilizar `"type": "module"` en el `package.json`)

Consultas (Queries)

Consultas sin parámetros

Para hacer consultas a la base de datos deberemos añadir el siguiente código:

```
const [result, fields] = await connection.query("ACÁ VA LA CONSULTA SQL");
```

En el caso de que la consulta sea un `SELECT`, en `result` tendremos un array con los resultados de la consulta. En caso de que sea un `INSERT`, `UPDATE` o `DELETE`, en `result` tendremos un objeto con información sobre la consulta.

En `fields` tendremos información sobre las columnas de la tabla, como el nombre, el tipo, etc.

Consultas con parámetros

Para hacer consultas a la base de datos con parámetros deberemos añadir el siguiente código:

```
const [result, fields] = await connection.query(  
  "ACÁ VA LA CONSULTA SQL CON ? Y ?",  
  [parametro1, parametro2]  
);
```

Este código funciona de la misma manera que el anterior, pero en este caso, los `?` serán reemplazados por los parámetros que pasemos en el array. Es decir, la query final que se ejecutará será "ACÁ VA LA CONSULTA SQL CON parametro1 Y parametro2".

Esto es útil para evitar inyecciones SQL, haciendo mucho más seguro nuestro código.

Aclaración: Los parámetros deben ser pasados en el mismo orden que los `?` en la consulta SQL. Porque el primer `?` será reemplazado por el primer elemento del array, el segundo `?` por el segundo elemento, y así sucesivamente.