UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
Centro de e-Learning
Secretaría de Cultura y Extensión Universitaria

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Diplomatura en programación web full stack con React JS

UTN.BA | Centro de e-Learning
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
Secretaría de Cultura y Extensión Universitaria

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Módulo 5:

Base de datos, integración con Node.js

Unidad 3:

Integración de Node.js con Base de datos (parte 1)

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Presentación

En esta unidad realizamos la conexión entre Node.js y MySQL mediante el uso de una librería de código abierto, vemos cómo guardar los datos de esta conexión en un archivo externo y cómo hacer consultas a la base de datos desde nuestra aplicación.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Objetivos

Que los participantes logren...

- Instalar y configurar las librerías necesarias para conectar Node.js con MySQL.
- Comprender el uso de las variables de entorno en una aplicación web.
- Realizar consultas a una base de datos desde Node.js.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bloques temáticos

1. Instalación de las dependencias.
2. Conexión a la BD.
3. Ejecución de consultas.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

1. Instalación de las dependencias

La manera más simple de instalar un módulo de Node.js de forma local es usar el comando **npm install** en la carpeta en la cual vamos a trabajar.

Esto combina dos pasos:

- Marca la última versión del módulo como una dependencia en tu archivo **package.json.**
- Descarga el módulo en tu directorio **node_modules**. Esto te permite usar el módulo cuando desarrollas de forma local.

Carga módulos Node.js

Utilizamos la función require() de Node.js para cargar cualquier módulo de Node.js que instalemos. También podemos usar la función **require()** para importar archivos locales que implementemos junto con nuestra función.

Dependencia mysql

Una de las maneras más fáciles de conectarse a MySQL es mediante el uso del módulo **mysql** . Esta dependencia maneja la conexión entre la aplicación Node.js y el servidor MySQL. Se instala así:

```
npm i mysql
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Dependencia util

La dependencia util nos va a permitir, entre otras cosas, convertir ciertas funciones que utilizan callbacks al modelo de promesas asincrónico.

```
npm i util
```

:

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

2. Conexión a la BD

Archivo .env y variables de entorno

Comúnmente usamos archivos sin nombre y con la extensión **.env** para almacenar datos sensibles como contraseñas, claves de API o datos de conexión de nuestra aplicación sin tener que escribirlos en el código y de esa forma hacerlos públicos. Estos archivos no suelen incluirse en sistemas de control de versiones como git.

Para utilizar los valores almacenados en este archivo instalaremos una librería llamada **dotenv**, cuya función es incorporar los contenidos de este archivo a las variables de entorno y de esta forma hacerlas disponibles para nuestra aplicación accediendo a las mismas mediante process.env..nombre_de_la_variable.

En este caso incluiremos los siguientes valores en nuestro archivo .env.

```
MYSQL_HOST=locahost
MYSQL_DB_NAME=nombre_de_las_base_de_datos
MYSQL_USER=root
MYSQL_PASSWORD=
```

Estos valores representan:

- **MYSQL_HOST:** el nombre de host de la base de datos a la que se está conectando.
- **MYSQL_DB_NAME:** nombre de la base de datos que se utilizará para esta conexión.
- **MYSQL_USER:** el usuario de MySQL con el que autenticarse.
- **MYSQL_PASSWORD:** es la contraseña de ese usuario de MySQL.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Conectándonos a la base de datos

Crearemos un archivo llamado **bd.js** en el cual utilizamos la función **require()** para cargar los módulos de mysql y util.

Todas las consultas en la conexión de MySQL se realizan una tras otra. Esto significa que si desea hacer 10 consultas y cada consulta tarda 2 segundos, se tardará 20 segundos en completar toda la ejecución. La solución es crear 10 conexiones y ejecutar cada consulta en una conexión diferente. Esto se puede hacer automáticamente utilizando el conjunto de conexiones y se ejecutarán todas las 10 consultas en paralelo.

Para esto, la librería de node.js mysql nos permite crear un pool o grupo de conexiones usando los datos de conexión que cargamos previamente.

Cuando usamos **pool** ya no necesitamos la conexión. Podemos consultar directamente al grupo de conexiones y el módulo MySQL buscará la siguiente conexión libre para ejecutar nuestra consulta.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
var mysql = require('mysql');
var util = require('util');


var pool = mysql.createPool({
    connectionLimit: 10,
    host: process.env.MYSQL_HOST,
    user: process.env.MYSQL_USER,
    password: process.env.MYSQL_PASSWORD,
    database: process.env.MYSQL_DB_NAME
})

pool.query = util.promisify(pool.query);


module.exports = pool;
```

En este ejemplo creamos una variable llamada pool, donde guardamos la referencia al grupo de conexiones, utilizando los datos que la librería dotenv extrajo de nuestro archivo .env.

Por último, tomamos el método query de pool y lo convertimos al modelo de promesas asincrónicas usando la libreria util.

Después de esto solo queda exportar la variable pool para incluirla donde necesitemos hacer uso de la base de datos en nuestra aplicación.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

3. Ejecución de consultas

SELECT

```
var pool = require('./bd');

pool.query("select * from alumnos").then(function(resultados){
    console.log(resultados);
});
```

INSERT

```
var pool = require('./bd');

var obj = {
    nombre: 'Juan',
    apellido: 'Lopez'
}

pool.query("insert into alumnos set ?", [obj]).then(function(resultados) {
    console.log(resultados);
});
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

UPDATE

```
var pool = require('./bd');

var id = 1;
var obj = {
    nombre: 'Pablo',
    apellido: 'Gomez'
}

pool.query("update alumnos set ? where id=?", [obj, id]).then(function(resultados) {

    console.log(resultados);
});
```

DELETE

```
var pool = require('./bd');

var id = 1;

pool.query("delete from alumnos where id = ?", [id]).then(function(resultados) {
    console.log(resultados);
});
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bibliografía utilizada y sugerida

Artículos de revista en formato electrónico:

Google Cloud. Disponible desde la URL: https://cloud.google.com/functions/docs/writing/specifying-dependencies-nodejs?hl=es-419

npmjs. Disponible desde la URL: https://www.npmjs.com/