



Implementación y gestión de una base de datos

Errores

Implementar la conexión a una base de datos PostgreSQL en una aplicación Node utilizando buenas prácticas.

Programar instrucciones para la obtención y manipulación de información desde una base de datos utilizando buenas prácticas, acorde al entorno Node.js.

{desafío}
latam_

- Unidad 1:
Implementación y gestión de una base de datos
- Unidad 2:
Transacciones y API REST
- Unidad 3:
Trabajo práctico



Te encuentras
aquí



¿Qué aprenderás en esta sesión?

- *Utiliza sentencias para la captura y procesamiento de errores de conexión y estados de base de datos utilizando el entorno Node.js*
- *Utiliza sentencias para la captura y manejo de errores durante las operaciones de manipulación acorde al entorno Node.js*

¿Alguna duda o
comentario sobre la
clase anterior?



/* Capturando errores de conexión */

Capturando errores de conexión

Los errores de conexión podemos capturarlos en el momento que ejecutamos una consulta a la base de datos, independientemente de la tabla que declaremos en la sentencia. Sabremos que se trata de un error de conexión por el código que nos devuelva el propio error estructurado por el paquete pg.

Ejercicio guiado: Capturando errores de conexión



Ejercicio guiado

Capturando errores de conexión

Consultar todos los registros de la tabla **ropa** de la base de datos **jeans**, pero voluntariamente escribir mal el nombre de la base de datos quitándole la “s” al final de la palabra e indispensablemente escribiremos un condicional, que detecte si existe el error y devuelva por consola el código de este.

Sigue los siguientes pasos para capturar el error de conexión:

Paso 1: Cambiar el nombre de la base de datos por uno incorrecto.

Paso 2: Usar trycatch para intentar ejecutar la consulta y capturar el error de la misma, extrayendo su código con destructuring.



/* Capturando errores de consultas */

Capturando errores de consultas

Al igual que los errores de conexión, los errores de consultas los podemos tomar por medio de sentencias trycatch. Esto quiere decir, que utilizaremos la misma forma para capturar cualquier error que ocurra en la comunicación con nuestras bases de datos.

Ejercicio guiado: Capturando errores en una consulta



Ejercicio guiado

Capturando errores en una consulta

Consultar a una tabla que no existe en la base de datos jeans. En este caso, simplemente agregarle una “s” al final del nombre de la tabla. En el siguiente código simplemente se cambió el nombre de la tabla que estamos consultando:

```
const consultar = async () => {  
  try {  
    const res = await pool.query("select *  
from ropas");  
    console.log(res.rows);  
  } catch (error) {  
    const { code } = error;  
    console.log(code)  
  }  
}  
  
consultar()
```



/* Error de conexión */

Error de conexión

Estos son los errores más comunes al generar una conexión a la base de datos PostgreSQL:

1. La IP o nombre del servidor es incorrecto.
2. El usuario existente en las credenciales de acceso no tiene los suficientes permisos para realizar operaciones en la Base de datos.
3. Los valores de las variables de entorno son incompatibles o no corresponden.
4. Protocolo SSL no soportado.
5. Se sobrepasó la máxima cantidad de conexiones.

/* Interpretación de errores PostgreSQL */

Interpretación de errores PostgreSQL

El servidor de base de datos PostgreSQL emite mensajes a los que se le asignan códigos de error de cinco caracteres que cumplen con convenciones del estándar SQL para los códigos SQLSTATE.

Es poco probable que estos códigos cambien en nuevas versiones de PostgreSQL. Según el estándar, los dos primeros caracteres corresponden a una clase de error y los últimos tres indican una condición específica en la clase.

**/* Clases de errores para poder
identificarlas vigente en la versión 12.x de
PostgreSQL */**

Clases de errores para poder identificarlas vigente en la versión 12.x de PostgreSQL

- Clase 00 - Finalización exitosa
- Clase 01 - Advertencia
- Clase 02 - Sin datos
- Clase 03 - Declaración SQL aún no completa
- Clase 08 - Excepción de conexión
- Clase 09 - Excepción de acción activada
- Clase 0A - Característica no admitida
- Clase 0F - Excepción del localizador
- Clase 0L - Concedente inválido
- Clase 0P - Especificación de rol no válida
- Clase 0Z - Excepción de diagnóstico
- Clase 20 - Caso no encontrado
- Clase 21 - Violación de cardinalidad
- Clase 22 - Excepción de datos
- Clase 23 - Infracción de restricción de integridad
- Clase 24 - Estado del cursor inválido
- Clase 25 - Estado de transacción no válido
- Clase 26 - Nombre de instrucción SQL no válido
- Clase 27 - Violación de cambio de datos desencadenada
- Clase 28 - Especificación de autorización no válida
- Clase 2B - Todavía existen descriptores de privilegios dependientes

Clases de errores para poder identificarlas vigente en la versión 12.x de PostgreSQL

- Clase 2D - Terminación de transacción no válida
- Clase 2F - Excepción de rutina SQL
- Clase 34 - Nombre de cursor inválido
- Clase 38 - Excepción de rutina externa
- Clase 39 - Excepción de invocación de rutina externa
- Clase 3B - Excepción de punto de rescate
- Clase 3D - Nombre de catálogo no válido
- Clase 3F - Nombre de esquema no válido
- Clase 40 - Reversión de transacciones
- Clase 42 - Error de sintaxis o infracción de la regla de acceso
- Clase 44 - CON OPCIÓN DE VERIFICACIÓN Infracción
- Clase 53 - Recursos insuficientes
- Clase 54 - Límite de programa excedido
- Clase 55 - Objeto no en estado de requisito previo
- Clase 57 - Intervención del operador
- Clase 58 - Error del sistema (errores externos al propio PostgreSQL)
- Clase 72 - Fracaso de instantánea
- Clase F0 - Error de archivo de configuración
- Clase HV - Error de contenedor de datos externos (SQL / MED)
- Clase P0 - Error PL / pgSQL
- Clase XX - Error interno

Clases de errores para poder identificarlas vigente en la versión 12.x de PostgreSQL

Como se pudo apreciar, el listado de códigos de error es muy largo, por lo tanto, se detallarán los códigos más comunes, incluyendo una breve descripción.

- Código 08003 - connection_does_not_exist
- Código 08006 - connection_failure
- Código 2F002 - modifying_sql_data_not_permitted
- Código 57P03 - cannot_connect_now
- Código 42601 - syntax_error
- Código 42501 - insufficient_privilege

- Código 42602 - invalid_name
- Código 42622 - name_too_long
- Código 42939 - reserved_name
- Código 42703 - undefined_column
- Código 42000 - syntax_error_or_access_rule_violation
- Código 42P01 - undefined_table
- Código 42P02 - undefined_parameter

Si quieres obtener el listado y detalle completo de estos errores, visita la [sección de errores en el sitio web oficial de PostgreSQL](#).

¿Qué diferencias pudiste
notar entre los errores que
vimos hoy?





Próxima sesión...

- *Guía de ejercicios - Implementación y gestión de una base de datos*

{desafío}
latam_

*Academia de
talentos digitales*

