

Explicación del código

1. Interfaz Statement

Mediante esta interfaz, se envían las órdenes SQL individuales a la base de datos a través del controlador JDBC y se recogen los resultados de estas. Cabe destacar que ésta nos protege de las Inyecciones SQL. Así podemos evitar que cualquier usuario pueda introducir sentencias no deseadas en nuestra BBDD.

2. Clase Pizza

En la clase pizza definimos los atributos correspondientes a la misma, donde tenemos el atributo name y url. El Id en este caso lo hemos definido en la clase Entity que explicaré posteriormente para poder adaptarlo a un UUID.

3. Interfaz Runables

Esta interfaz tiene dos métodos, el primero de ellos "getSQL()" sirve para llamar a la transacción (consulta) SQL. Por otra parte, el segundo método, preparedStatement, se usará para ejecutar las consultas de manera segura, y si hay algún error, se lanzará una excepción.

4. Implementación Runables

Aquí implementamos la interfaz runables descrita previamente. Esta clase es genérica, ya que puede tomar el valor de cualquier cosa que se le indique. Se definen los atributos sql, entity y statement. También el constructor. Luego tenemos dos métodos que uno devuelve el atributo SQL y el siguiente no devuelve nada porque es void ya que hace un preparedStatement de los atributos entity y statement.

5. Interfaz Resultset

En esta interfaz obtenemos por parámetros un ResultSet y una entity genérica. Su función es encargarse de insertar los datos obtenidos del ResultSet a la entity genérica usando Lambda.

6. Clase Abstracta Entity

En esta clase definimos la creación del atributo UUID, los getters y setters del mismo y la validación a la hora de pasarlo en el Select para que se pueda leer a través de la BBDD. También definimos los métodos equals y hashCode para devolvernos el ID y realizar la comparación correspondiente y comprobar si se cumple la igualdad.

7. Clase Converter

Esta clase sirve para convertir un UUID a ByteArray para poder usarlo posteriormente. También definimos el caso contrario.

8. Interfaz Entity Manager

Aquí definimos 4 metodos, un método save en el que se guardaran las transacciones. Los siguientes 3 metodos son genéricos, los dos primeros corresponden a la misma interfaz, con el addStatement añades un statement a una lista que se va a ejecutar para la base de datos (runables) y con el addRangeStatement, le pasas varios statement en vez de uno. El ultimo es el método optional Resultset, el cual usamos para definir el select.

9. Implementación EntityManager

Para empezar, tenemos un ArrayList correspondiente a la interfaz runables y un atributo inicializado a null que corresponde a la interfaz configuration. Siguiendo el código, aparece un método Save donde se inicializa una conexión a null. Mediante el try and catch se establece la conexión a la BBDD usando el DriverManager y se llaman a los métodos de la interfaz configuración url, user y password. Luego se realiza un bucle for para recorrer el arraylist y ejecutar la sentencia SQL. Y en el resto de las líneas de código se realiza un catch para ejecutar las correspondientes excepciones. Así pues, este método save ejecuta la sentencia SQL para escribir, modificar o borrar y lo guarda en la BBDD.

También aparece el método Select donde se realiza una operación similar a lo anteriormente explicado, pero en este caso para devolverme y consultar un objeto de la BBDD.

Posteriormente encontramos los métodos addStatement y addRangeStatement que explicamos en la interfaz. Por último, el método buildConnection es estático donde se le pasa una variable de la interfaz Configuration y retorna una nueva instancia de la variable pasada como parámetro, y construye una nueva conexión.

10. Interfaz Configuración

Aquí se definen tres métodos, getUser, getPassword y getUrl para obtener dichos atributos.

11. Implementación Configuración

Aquí implementamos la interfaz configuración descrita previamente. Además, la inicializamos a null. Definimos un método de la interfaz configuración donde si fuera nulo, se crea una nueva configuración y se retorna esa misma configuración. Por último, definimos nuestras variables de entorno donde se introducen el usuario, el password y la url de nuestra base de datos para ejecutar la conexión a la misma.

12. Clase App

Aquí se crea el main donde se va a ejecutar el programa. En primer lugar, creamos un nuevo objeto Pizza (en el caso de crear una Pizza, ya que sirve igual para ingredientes, usuarios, etc) y le definimos los atributos id, name y url. Después llamamos a la implementación EntityManager y Configuration para construir la conexión con la base de datos y añadir la sentencia correspondiente a ejecutar. En el caso del save, vamos a guardar en la base de datos una nueva pizza, donde con los statements correspondientes, introducimos el orden en el cual guardamos los valores de id, name y url. Dicho id lo transformamos con un método de UUID a ByteArray para poderse leer ya que es del tipo Binary. Por último, ejecutamos el método Save para guardarla. En el caso del Select el código sería similar, solo que usamos la interfaz ResultSet definida anteriormente para sacar a través de la clase Entity los atributos id, name y url.