

Documento explicativo DAO

Este documento vamos a explicar el funcionamiento del DAO mediante la aplicación que hemos creado para hacer de ejemplo.

App

En la clase main App tenemos el mock del ingrediente “cheese” y las funciones lambda.

Converter

En esta clase tenemos un conversor para pasar el UUID a Bytes y viceversa.

Configuration

En la interfaz configuration tenemos tres métodos definidos `getUser()`, `getPassword()`, `getUrl()`. Estos los usaremos en la clase `ConfigurationImp`.

ConfigurationImp

En esta clase utilizamos el patrón Singleton para crear una conexión con la BBDD. Con la implementación de la interfaz configuration utilizamos los métodos mencionados anteriormente para acceder a las variables de entorno de nuestro equipo que usaremos para acceder a la BBDD de forma segura.

Ingredient

Es la clase que estamos utilizando para hacer la prueba, tiene los atributos `name`, `price` e `id` el cual obtenemos al extender de la clase abstracta `entity`. Además, tenemos `getters`, `setters` y dos constructores.

Entity

Como comentábamos en `Ingredient` aquí tenemos el atributo `id` junto a su `getter` y `setter`. También tenemos el método `validate` para comprobar la `id` y el `override` de los métodos `equals` y `hashCode`.

ResultSet

Esta interfaz funcional recibe por parámetro un ResultSet y un Generic para insertar los datos obtenidos del ResultSet en el Generic y poder utilizarlo mediante una función lambda.

Statement

Esta interfaz funcional recibe por parámetro un PreparedStatement y un Generic. La utilizamos para obtener los datos del Generic que luego insertaremos en el PreparedStatement y completar así la Query.

Runables

En esta interfaz tenemos el método getSQL() que utilizamos para obtener las Querys y el método run() que utilizamos para ejecutar el PreparedStatement.

RunablesImp

En esta clase implementamos la interfaz Runables, utilizamos la clase para almacenar en variables de tipo final querys, un Generic y la interfaz funcional Statement. Con el método getSQL() nos devuelve el valor de la variable junto a la query y el método run() que ejecuta la interfaz funcional pasándole por parámetro el PreparedStatement que hemos obtenido del método de la interfaz y la entidad que nos hemos guardado en la variable final.

EntityManager

Es la interfaz donde definimos los métodos buildConnection(), addStatement(), addRangeStatement(), save() y select().

EntityManagerImp

Esta clase implementa la interfaz EntityManager.

builtConnection: Utilizamos este metodo para crear la conexión con la BBDD.

addStatement: Nos crea un runnable con una única Query y una entidad (Generic) usando las funciones lambda que será añadida a la lista de runables.

AddRangeStatement: Como addStatement solo que creamos varios runables con una única Query y pasando por parámetro un Iterable junto a la query.

Select: Conecta con la BBDD y nos devuelve un Optional <T> obtenido a través de una query de lectura que será la primera en la lista de runables, ahí hará uso del ResultSet para crear una instancia de clase <T> que pasamos por parámetro al método y ejecutamos la interfaz funcional ResultSet <T> para sacar los datos del ResultSet e introducirlos en la entidad, luego devolvemos el Optional de la entidad (en nuestro caso ingredient). Por último, cerramos la conexión con la BBDD y vaciamos la lista de runables.

Save: Conecta con la BBDD y ejecuta la lista de runables. Por último, cerramos la conexión con la BBDD y vaciamos la lista de runables.