

Para empezar, la clase Configuration que implementa la interfaz IConfiguration, que contiene los métodos que retornan el valor de las variables de entorno que dan el acceso a la base de datos, con el método getConfiguration, si es nula, inicializa una nueva Configuration y la devuelve.

Las interfaces IEntityManager e IRunables contienen los métodos que van a usar sus implementaciones, algo que explicaré más adelante.

La interfaz Statement está parametrizada con T que se acaba usando para darle un tipo a la entidad en el método run

La clase Runables<T> que implementa la interfaz IRunables tiene un constructor que recibe sus 3 atributos por parámetro (Una cadena para la sentencia, la entidad de tipo genérico y un Statement<T>). Un getter para la cadena sql y un método run que recibe por parámetro un PreparedStatement que lo envía junto a la entidad genérica al método run del Statement<T> que ha inicializado en el constructor.

En EntityManager obviando el método select que tenemos que implementar por nuestra cuenta, tiene 4 métodos y un constructor:

- El método estático buildConnection recibe una IConfiguration y esta se la pasa al constructor para que lo guarde en el IConfiguration que tiene como atributo.
- El método addStatement, al igual que el constructor de Runables, recibe por parámetro una entidad genérica, una cadena sql y un Statement<T>, estos se los guarda por constructor a un nuevo IRunables que se añade a una lista de IRunables, después de esto se devuelve a sí mismo en IEntityManager para poder seguir con las operaciones.
- El método addRangeStatement, recibe los mismo que addStatement, a excepción de un Iterable de una entidad genérica y por cada entidad que haya dentro del iterable hará el proceso anterior, después de esto se devolverá a sí mismo.
- El método save, crea una conexión con los datos de las variables de entorno almacenados en configuration, desactiva el autocommit y por cada runnable de la lista runables, crea un PreparedStatement con la cadena sql almacenada en el runnable, lo usa en el método run de Runables y lo ejecuta, al acabar todas las operaciones las commitará; si por algún caso el commit fallase, con un catch hará un rollback para descartar las sentencias y al acabar con un finally cerrará la conexión.

Finalmente, en App al EntityManager (después de construir la conexión) al añadir un statement tiene que añadir un Statement<T> por expresión lambda define que quiere que haga ese objeto, en este caso con el statement y el entity del objeto Statement<T> añade al statement el id de entity, esto ocurrirá cuando en EntityManager, en save(), cuando ejecuta el run de runnable, que este ejecuta el metodo run de Statement<T> este recibe el PreparedStatement y la entidad ya especificada y entonces hace la función que le es definida por lambda, es decir le añade al PreparedStatement el id de la entidad especificada