Programación Go

Storage Implementation

// Práctica clase 2 - Storage Implementation

Objetivo

Escribir los tests necesarios para asegurar la calidad de la implementación del repository sql. Utilizando las librerías vistas en clase, generar una suite de tests para poder testear el correcto funcionamiento de la implementación.

Forma de trabajo

Generar un nuevo proyecto. Con los recursos dados en la clase hacer la creación de la DB MySql llamada “mydb”. Descargar e incluir en el proyecto los archivos que contienen la interfaz del repository en conjunto con su implementación.

Are you ready? 😎👍

Ejercicio 1 - Testear Store() y GetOne()

Suponiendo la correcta implementación de los métodos Store y GetOne. Escribir un test utilizando go-txdb en donde se guarde un User y luego se obtenga, en el caso de que no exista GetOne debe devolver un resultado nulo. No hay límite de cantidad de tests ni casos, crear los que considere necesarios para asegurar el correcto funcionamiento de las distintas variaciones.

Ejercicio 2 - Testear Update() y Delete()

Generar tests para update, en donde se verifique que luego de modificarse un modelo, al obtenerlo el mismo posea los cambios realizados.
Generar tests para delete para verificar que un registro fue borrado correctamente y ya no se puede obtener ni utilizando GetOne ni al llamar a GetAll.

Ejercicio 3 - Replicar tests anteriores utilizando mocks

Tomar alguno de los tests realizados en los ejercicios anteriores (o todos) y replicarlos utilizando go-sqlmock.

Ejercicio 4 - Testear que sucede en el caso de que falle una query

Seleccione alguno de los métodos (Store, GetOne, GetAll, Update o Delete) y simule un error al ejecutar la consulta. El test debe pasar si ocurre un error. (assert.Error). El objetivo de este ejercicio es conocer la forma de ver como reacciona nuestra implementación frente a una falla.