

INTEGRATION TESTING EXERCISES

CupCakeShop project consists of some entities (bottom / top / cake), a databaseconnector, a datasource and a datamapper that makes it possible to create, update, delete and get tops, bottoms and cakes

A sql script for creating the database for the CakeMapper is included in the SQL folder

A xml file for DBUnit is included in the DATASETS folder

In the main.Main.java file, it can be seen how the CakeMapper is to be used

Create tests for the CakeMapper in CupCakeShop project

Create test files in 4 different ways, with MySQL, DBUnit, H2 and Mockito

Create new datasources for the different tests if needed in the test files

The tests should assert that the name of the cake with id 1 is "ChoccoNut" by using `CakeMapper.getCakeById`, then update the name to "ChocolateNuts" by using `CakeMapper.updateCake` and finally assert that the new name of the cake with id 1 is "ChocolateNuts" by using `CakeMapper.getCakeById`

The end result should be the same `@Test` in each of the different test files, but different ways of setting up the CakeMapper before the tests are executed

MYSQL

Create tests where a connection to a MySQL test database is created and used to execute a sql script file that resets the MySQL test database to a known state before the tests

DBUNIT

Use DBUnit to clean the MySQL test database before the tests with a XML file instead of executing a sql script file

H2

Use embedded / in-memory or network database instead of mysql test database for the tests

MOCKITO

Use Mockito to create mocks needed to test the CakeMapper