



This repository ▾ Search or type a command ⓘ

Explore Gist Blog Help

 rofr + ▾ ✂ 📄

 DevrexLabs / OrigoDB.Workshop

👁 Unwatch ▾ 1 ⭐ Star 0 🍴 Fork 0

03 Testing

rofr edited this page 10 hours ago · 3 commits

This module explores automated testing alternatives for your data models.

Notes: Uses NUnit, assumes you have a test runner available. There is an extension available for VS, also resharper supports NUnit out of the box.

Exercise Task 1 - Setup NUnit

1. Open the Todo project in Visual Studio from the Ex01 folder
2. Add a class library to the solution called `Todo.Test`
3. In the new library, add a nuget reference to `NUnit`
4. Add a reference to the `Todo.Core` project
5. Rename the `Class1` to `TodoTests` and add `TestFixture` attribute.

Exercise Task 2 - Create a simple test

1. Create a test by adding the following method:

```
[Test]
public void Todo_Constructor_args_are_picked_up()
{
    const int expectedId = 42;
    const string expectedTodo = "Give the dog a dose of meth";

    var todo = new Core.Todo(expectedId, expectedTodo);
    Assert.AreEqual(expectedId, todo.Id);
    Assert.AreEqual(expectedTodo, todo.Title);
}
```

2. Execute the test

Exercise Task 3 - Write more tests

You're on your own now, create tests for the following:

1. That `TodoModel.SetCategory(int,category)` creates a category when missing
2. That `SetCompletedCommand` has the desired effect
3. Create a general smoke test which creates an engine, executes some commands and queries. Tip: You can create an engine for isolated test and in-memory storage using the following syntax:

```
var config = new EngineConfiguration().ForIsolatedTest();
var engine = Engine.LoadOrCreate<TodoModel>(config);
```

▼ Pages 8

01 Introduction
02 Modeling
03 Testing
04 Hosting
05 Configuration
06 Immutability
07 Server
Home

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/DevrexLabs/OrigoDB.Workshop/wiki/03-Testing> 📄

📄 Clone in Desktop

+ Add a custom footer

