Backend code test - banner

Solution structure

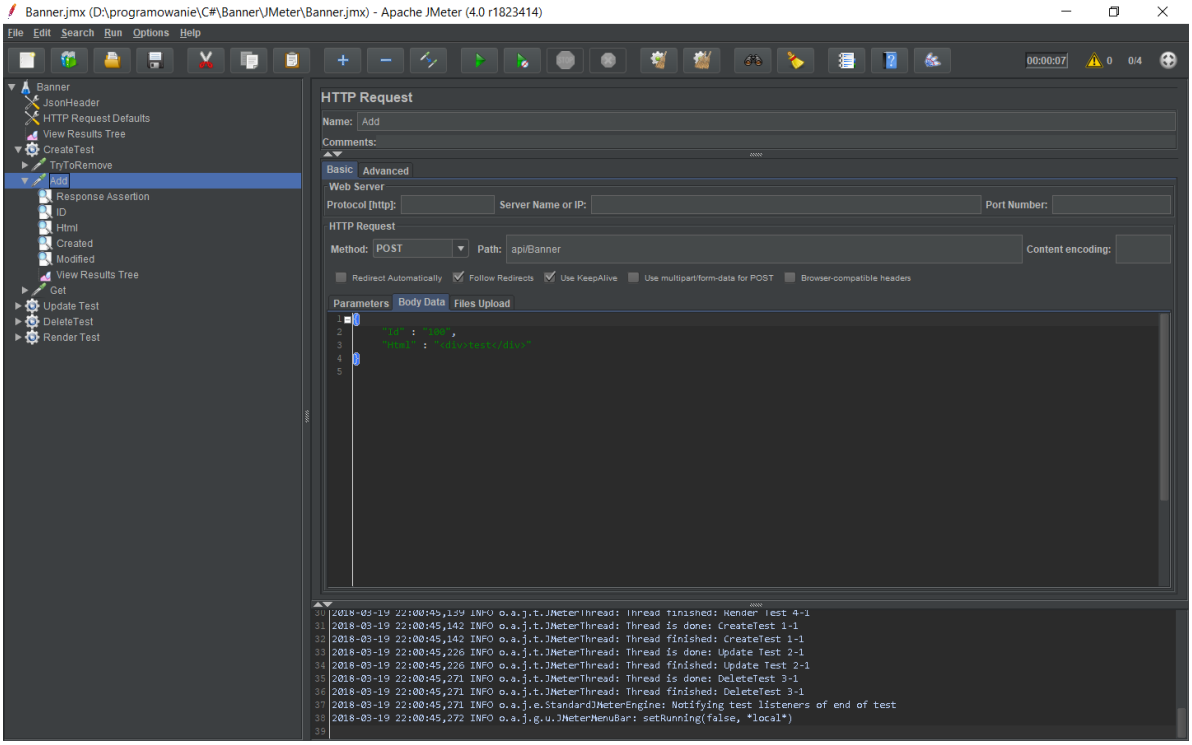
Solution consists of 9 projects. They are grouped into 4 categories:

- Application
  - Web
    It's ASP.NET Web Api 2 project that have all CRUD operations and is able to render our banner.
  - ConsoleTest
    Project which was sometimes used to test something
- Project Specific
  - Infrastructure
    Containts all interfaces and data objects.
  - Mongo
    Contains repositories and unit of work.
  - Services
    Contains services used in this project. It has only banner service.
- Tests
  - UnitTests
  - AcceptanceTests
  - TestsCommon
- Other
  - Common
    Common files used through the project.

Api description

API	Description
GET Api/Banner/Render/{id}	Displays banner HTML as web page.
GET api/Banner/{id}	Gets data about specific banner.
GET api/Banner?pageNumber={pageNumber}&pageSize={pageSize}	Let's you see paged information about all banners in system.
POST api/Banner	Creates new banner with given Id.
DELETE api/Banner/{id}	Removes banner from the server.

Web Api testing process



I've used JMeter software to test backend REST Api. It was my first time using JMeter so I am not quite used to it and might not know all best practices.

It tests whether Create, Update, Delete and render actions provide correct results. I did not created Read test because it is tested along the way with other tests.

I am using endpoint created by me at Azure Portal with address http://bannerrest.azurewebsites.net/ .

JMeter configuration file lies inside JMeter folder in repository.

Most of the tests have simple structure like this:

- Try to remove old banner if exists
- Make tested action and assert result if able
- Assert result from get action

C# Tests

For testing I used XUnit framework as it has tests with parameters. It helps to not produce redundancy code. I used typical AAA layout for all tests.

For mocking pursposes framework Moq was used. I am used to it and it's easy in use for me.

Acceptance tests used in BannerControllerTests are overlapping with JMeter tests but they are different in nature. They are just using BannerController not the whole IIS server.

Connection strings

Connection strings are very vulnerable information and are not stored inside repository. They are automatically added to the web.config (or app.config for tests) from connection.config file. I attached file content in the email.

Links:

Repository

Api