AMF .Net Tools Manual Tests

Instructions on doing manual tests with the AMF .Net Tools extensions

Pre-requisites:

- Windows box

Set up instructions:

1. Download Visual Studio 2017 Community Edition from https://visualstudio.microsoft.com/downloads/
2. Download Visual Studio extension from https://github.com/mulesoft/amf-dotnet-tools/releases
3. Close Visual Studio and run the vsix file

Test scenarios:

1. Client Proxy:
    a. Create a new console project in Visual Studio
    b. Right click on the “References” or “Dependencies” node on the solution explorer
    c. Choose “Add RAML/OAS reference…” command
    d. Select an RAML or OAS file from disk, you have many available here: https://github.com/mulesoft/amf-dotnet-tools/tree/develop/src/tools/AMF.Tools.Tests/files or choose an public API from Exchange.
    e. Once the code is generated, build the solution to check it compiles successfully.

2. Server scaffolding:
    a. Create a new Asp.Net Web Api or Asp.Net Core project in Visual Studio
    b. Right click on the web project node on the solution explorer
    c. Choose “Add RAML/OAS contract…” command
    d. Select an RAML or OAS file from disk, you have many available here: https://github.com/mulesoft/amf-dotnet-tools/tree/develop/src/tools/AMF.Tools.Tests/files or choose an public API from Exchange.
    e. Once the code is generated, build the solution to check it compiles successfully.

3. Get RAML from existing Web API implementation:
    a. Open an existing Asp.Net Web Api or Asp.Net Core with Visual Studio (you can also create the default solution and use the auto generated code).
    b. Right click on the web project node on the solution explorer
    c. Choose “Extract RAML” command
    d. Run your web project by hitting the F5 button. This opens the web browser. Add /raml to your web project URL to navigate to the API console.

For more info on these scenarios and the tools, check the readme at:
https://github.com/mulesoft/amf-dotnet-tools