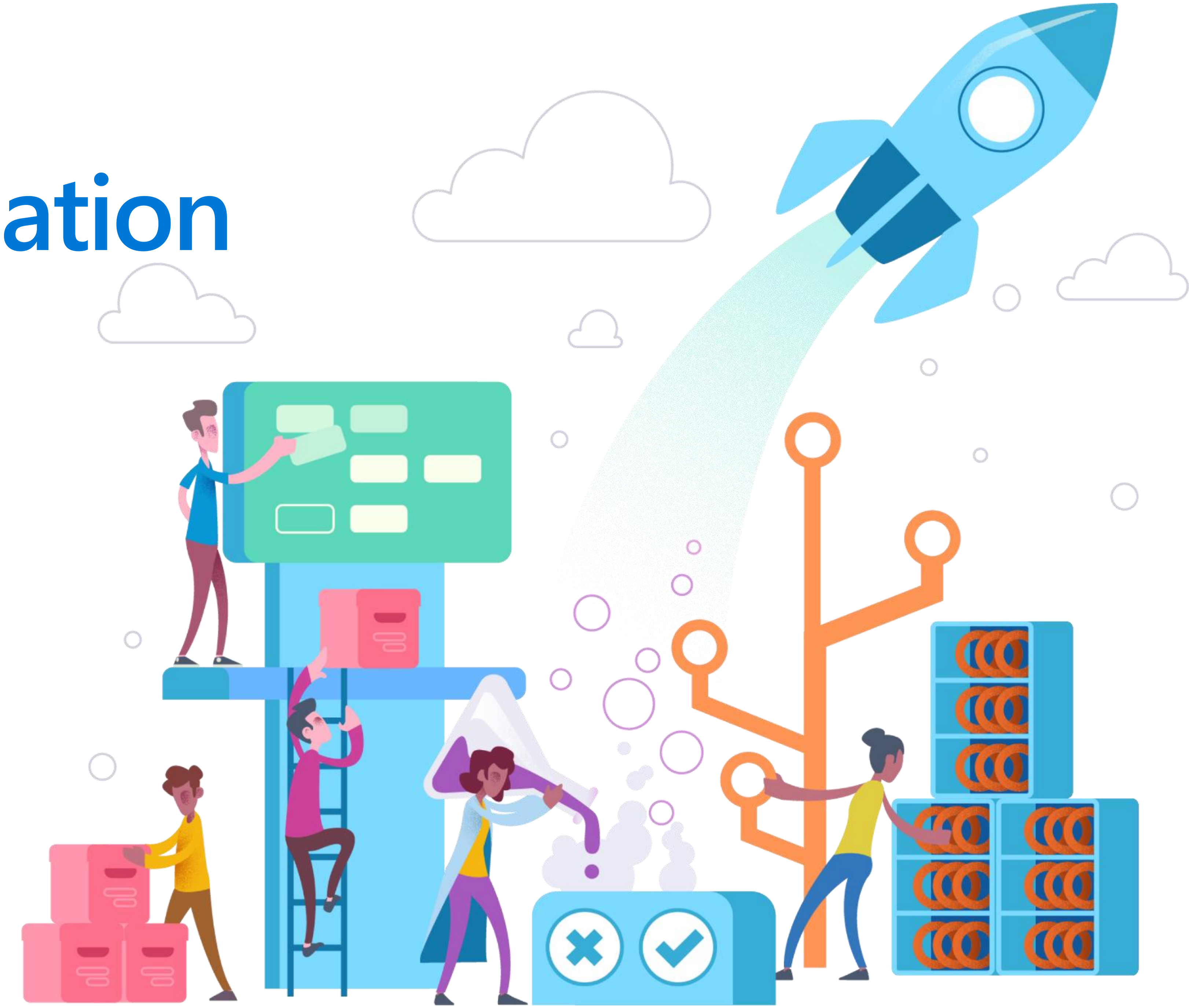
Microsoft

Testing Automation



Juan Osorio
Premier Field Engineer - Apps

Module: xUnit

Overview

- xUnit
- Dependency Injection Pattern
- Moq
- Code Analysis

xUnit

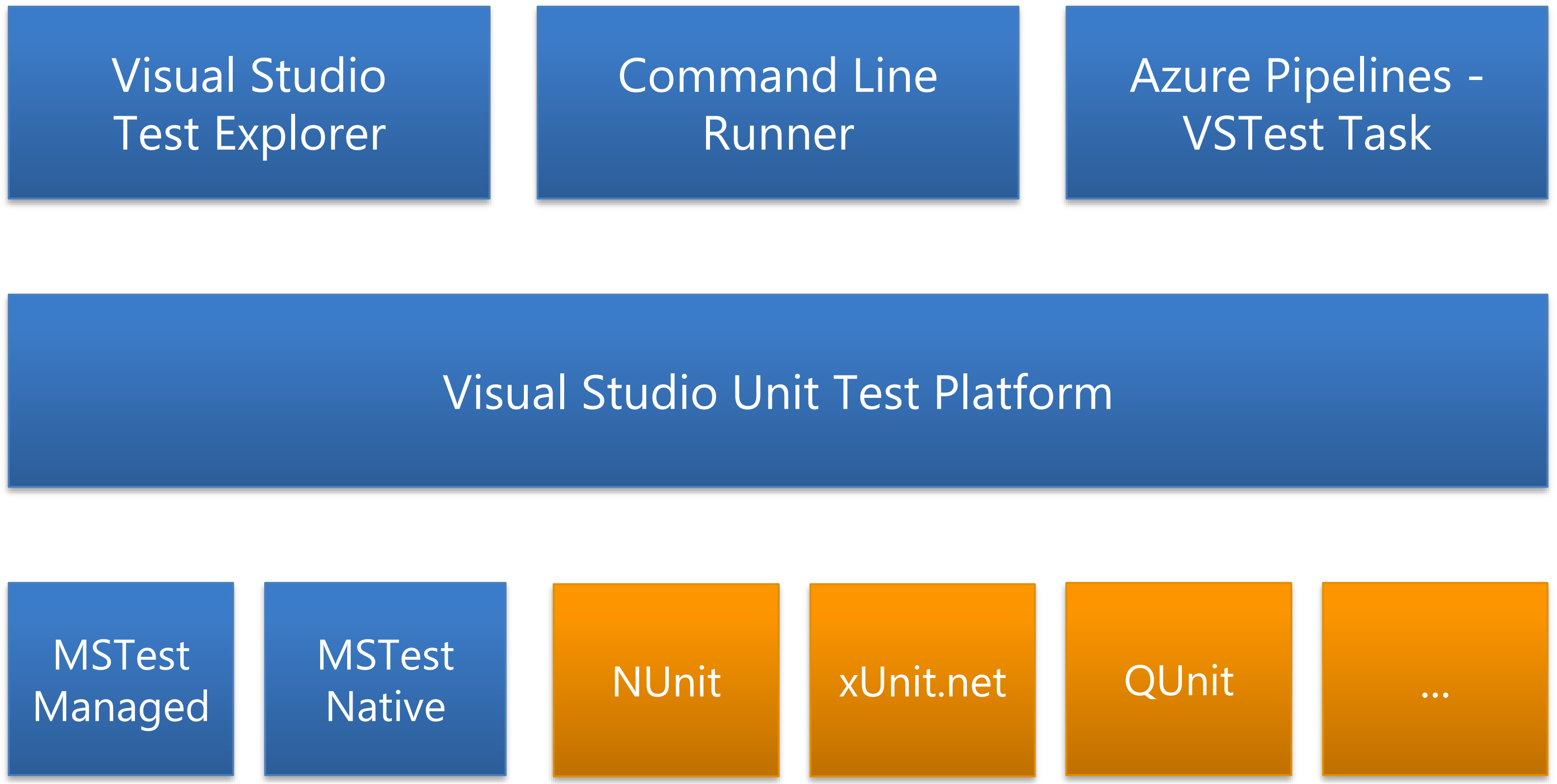
xUnit

xUnit.net is a free, open source, community-focused unit testing tool for the .NET Framework

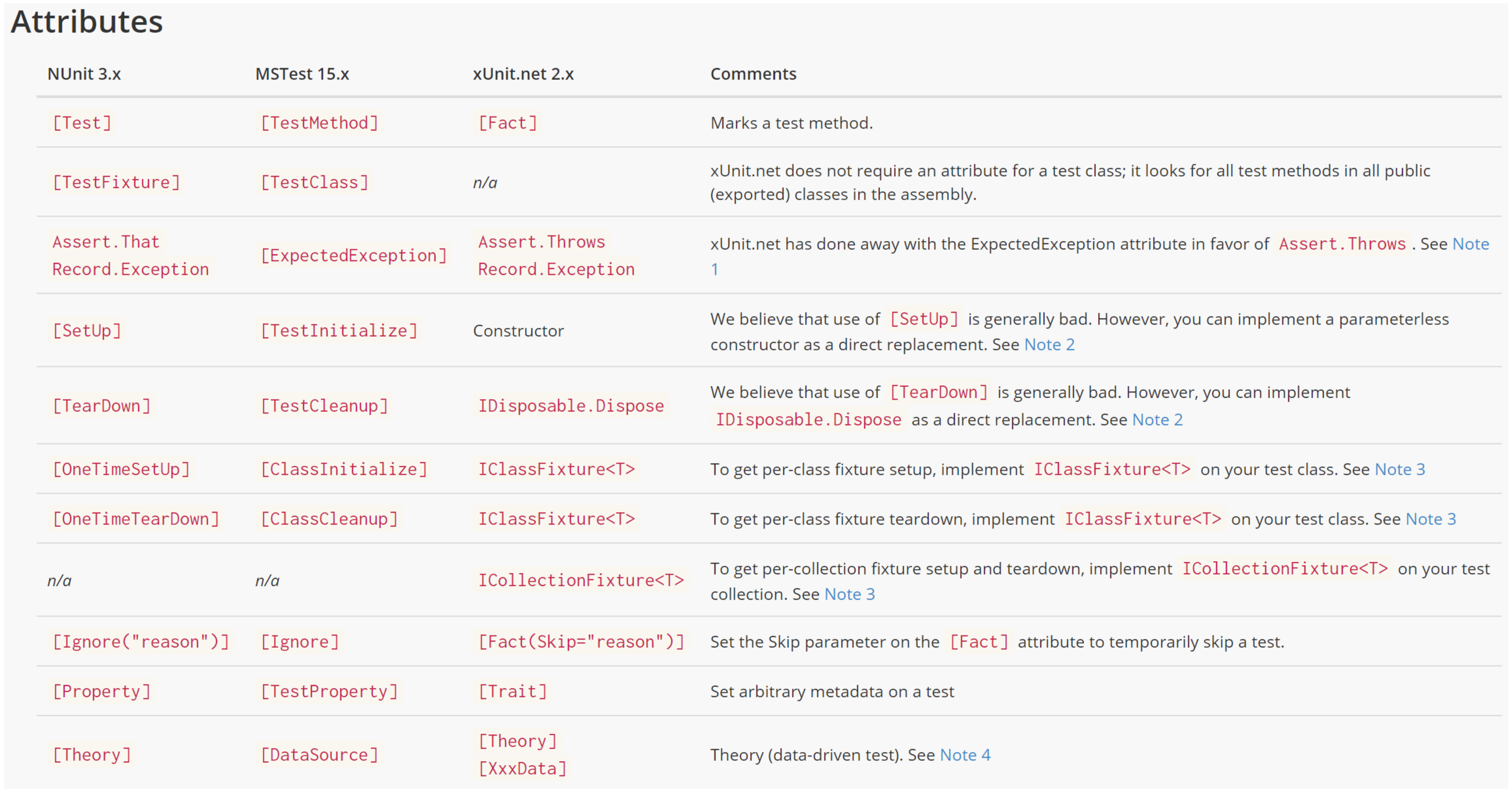
It is part of the .NET Foundation

Works with Visual Studio Architecture

Visual Studio Unit Test Architecture


Visual Studio Test Explorer
Command Line Runner
Azure Pipelines - VSTest Task
Visual Studio Unit Test Platform
MSTest Managed
MSTest Native
NUnit
xUnit.net
QUnit
...

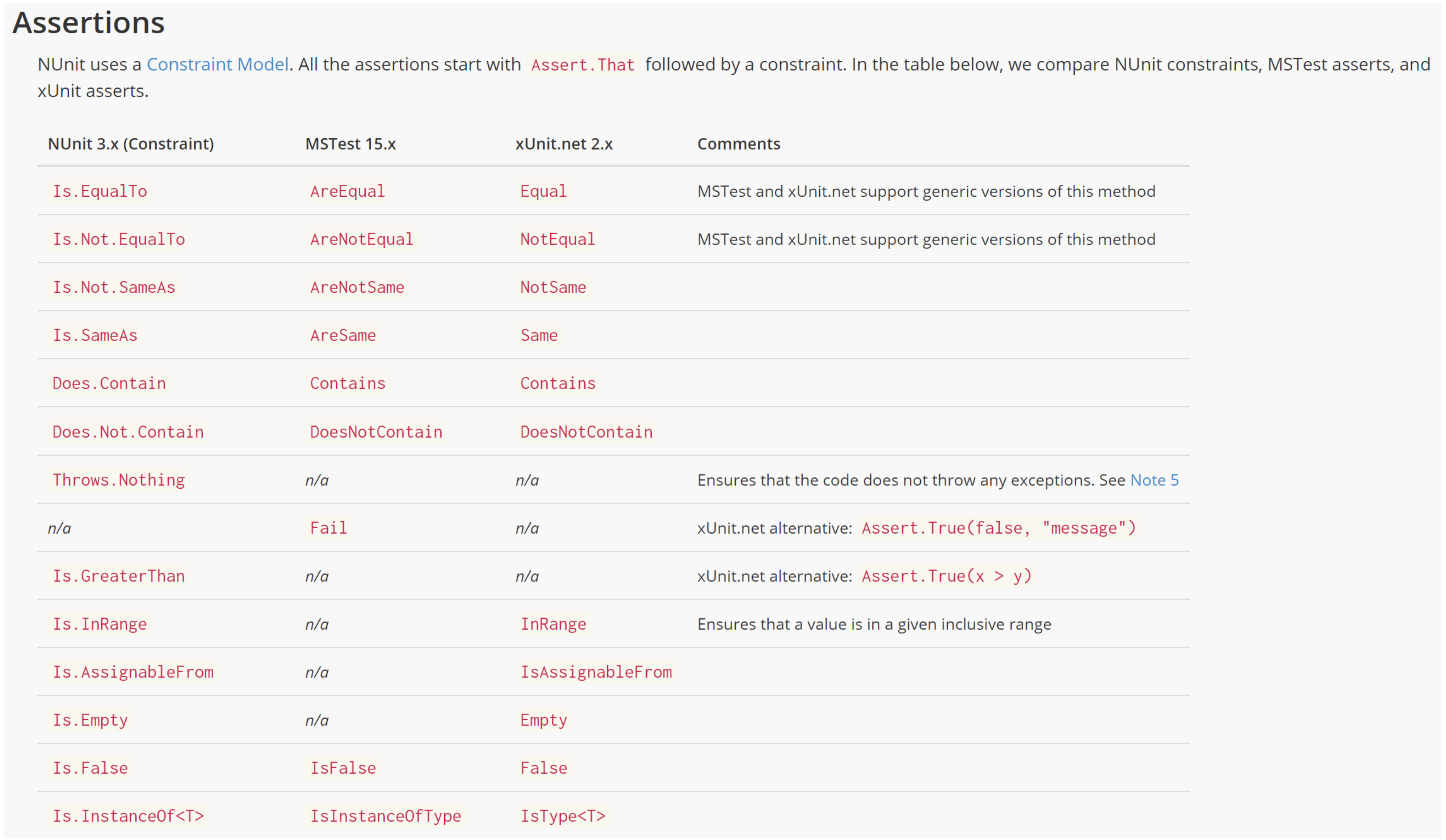
Comparing xUnit.net to other frameworks



Attributes

NUnit 3.x	MSTest 15.x	xUnit.net 2.x	Comments
`[Test]`	`[TestMethod]`	`[Fact]`	Marks a test method.
`[TestFixture]`	`[TestClass]`	*n/a*	xUnit.net does not require an attribute for a test class; it looks for all test methods in all public (exported) classes in the assembly.
`Assert.That` `Record.Exception`	`[ExpectedException]`	`Assert.Throws` `Record.Exception`	xUnit.net has done away with the ExpectedException attribute in favor of `Assert.Throws` . See Note 1
`[SetUp]`	`[TestInitialize]`	Constructor	We believe that use of `[SetUp]` is generally bad. However, you can implement a parameterless constructor as a direct replacement. See Note 2
`[TearDown]`	`[TestCleanup]`	`IDisposable.Dispose`	We believe that use of `[TearDown]` is generally bad. However, you can implement `IDisposable.Dispose` as a direct replacement. See Note 2
`[OneTimeSetUp]`	`[ClassInitialize]`	`IClassFixture<T>`	To get per-class fixture setup, implement `IClassFixture<T>` on your test class. See Note 3
`[OneTimeTearDown]`	`[ClassCleanup]`	`IClassFixture<T>`	To get per-class fixture teardown, implement `IClassFixture<T>` on your test class. See Note 3
n/a	*n/a*	`ICollectionFixture<T>`	To get per-collection fixture setup and teardown, implement `ICollectionFixture<T>` on your test collection. See Note 3
`[Ignore("reason")]`	`[Ignore]`	`[Fact(Skip="reason")]`	Set the Skip parameter on the `[Fact]` attribute to temporarily skip a test.
`[Property]`	`[TestProperty]`	`[Trait]`	Set arbitrary metadata on a test
`[Theory]`	`[DataSource]`	`[Theory]` `[XxxData]`	Theory (data-driven test). See Note 4

Comparing xUnit.net to other frameworks

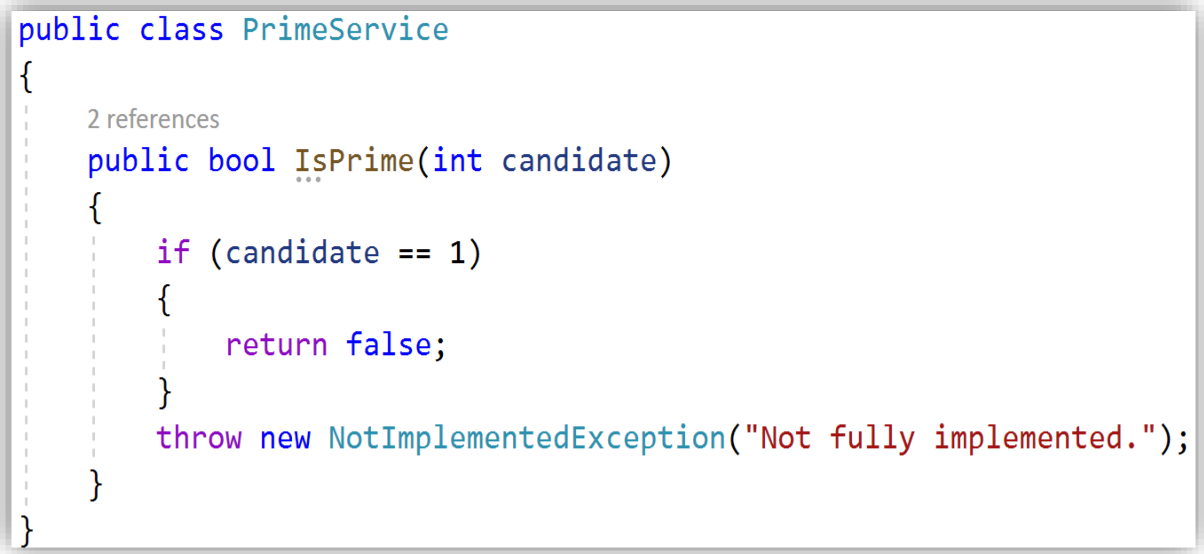


Assertions

NUnit uses a Constraint Model. All the assertions start with `Assert.That` followed by a constraint. In the table below, we compare NUnit constraints, MSTest asserts, and xUnit asserts.

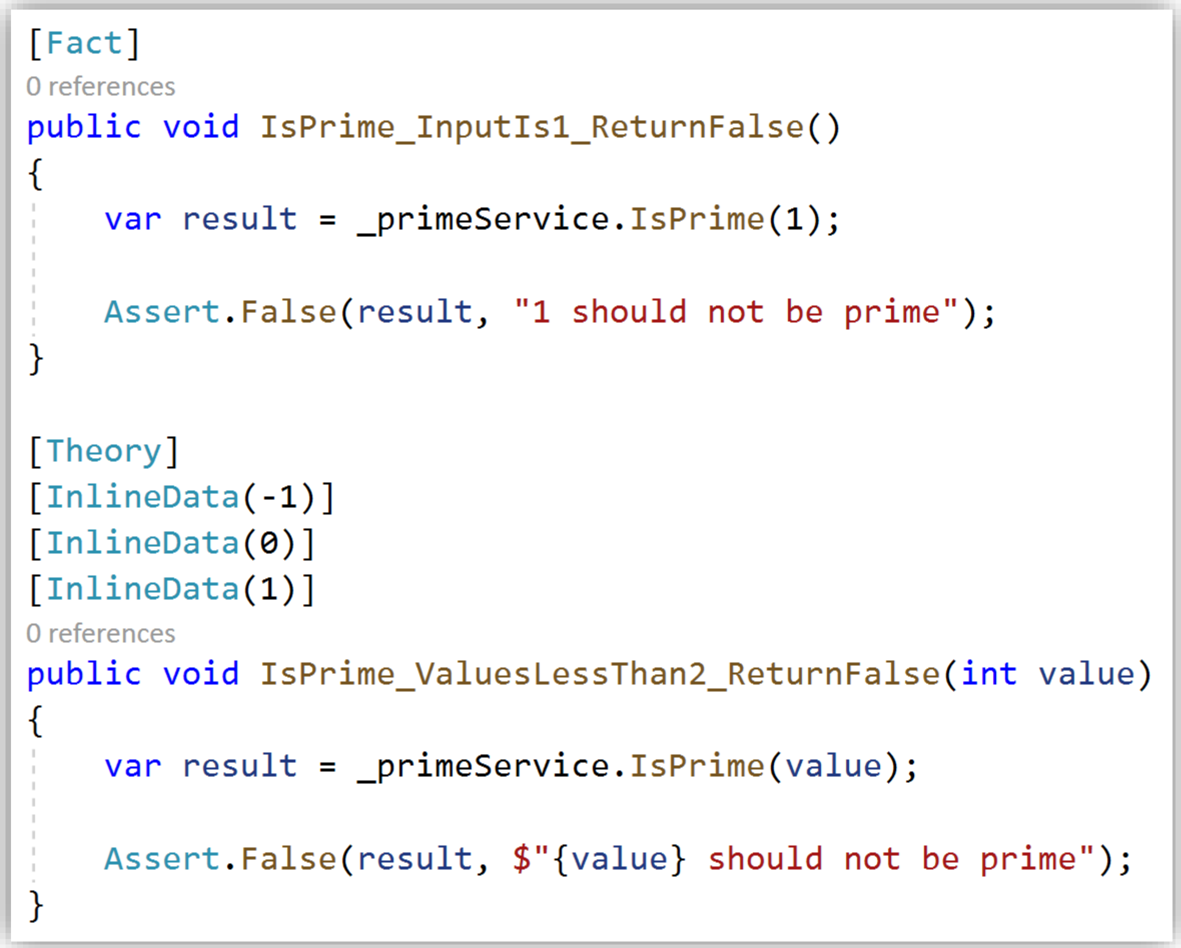
NUnit 3.x (Constraint)	MSTest 15.x	xUnit.net 2.x	Comments
`Is.EqualTo`	`AreEqual`	`Equal`	MSTest and xUnit.net support generic versions of this method
`Is.Not.EqualTo`	`AreNotEqual`	`NotEqual`	MSTest and xUnit.net support generic versions of this method
`Is.Not.SameAs`	`AreNotSame`	`NotSame`	
`Is.SameAs`	`AreSame`	`Same`	
`Does.Contain`	`Contains`	`Contains`	
`Does.Not.Contain`	`DoesNotContain`	`DoesNotContain`	
`Throws.Nothing`	*n/a*	*n/a*	Ensures that the code does not throw any exceptions. See Note 5
n/a	`Fail`	*n/a*	xUnit.net alternative: `Assert.True(false, "message")`
`Is.GreaterThan`	*n/a*	*n/a*	xUnit.net alternative: `Assert.True(x > y)`
`Is.InRange`	*n/a*	`InRange`	Ensures that a value is in a given inclusive range
`Is.AssignableFrom`	*n/a*	`IsAssignableFrom`	
`Is.Empty`	*n/a*	`Empty`	
`Is.False`	`IsFalse`	`False`	
`Is.InstanceOf<T>`	`IsInstanceOfType`	`IsType<T>`	

xUnit Unit Test Example



```
public class PrimeService
{
    2 references
    public bool IsPrime(int candidate)
    {
        if (candidate == 1)
        {
            return false;
        }
        throw new NotImplementedException("Not fully implemented.");
    }
}
```

PrimeService.cs



```
[Fact]
0 references
public void IsPrime_InputIs1_ReturnFalse()
{
    var result = _primeService.IsPrime(1);

    Assert.False(result, "1 should not be prime");
}

[Theory]
[InlineData(-1)]
[InlineData(0)]
[InlineData(1)]
0 references
public void IsPrime_ValuesLessThan2_ReturnFalse(int value)
{
    var result = _primeService.IsPrime(value);

    Assert.False(result, $"{value} should not be prime");
}
```

PrimeServiceTests.cs

Demo

xUnit in Visual Studio

Knowledge check

- Why xUnit is able to run on Visual Studio?

- Difference between Fact and Theory

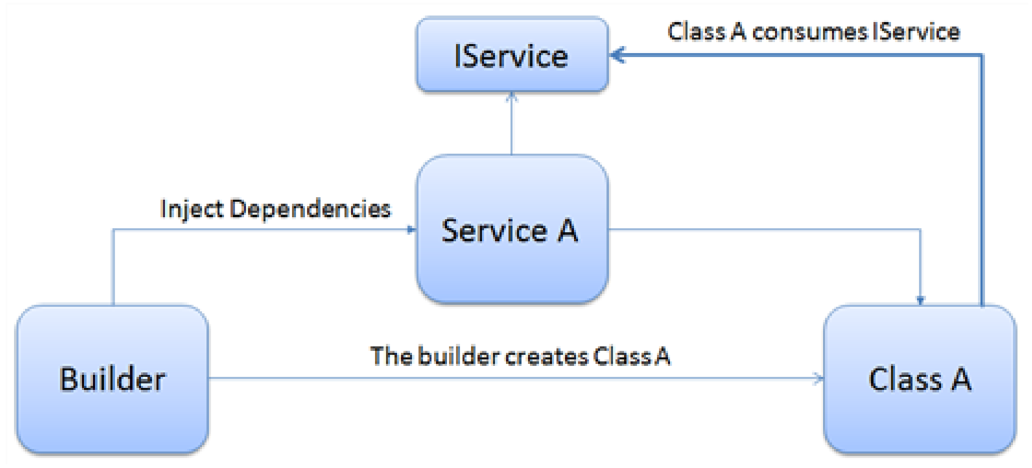
Dependency Injection Pattern

Dependency Injection Pattern

- Loose coupling
- Code more maintainable and reusable
- Unit Testing Possible
- Liskov Substitution Principle

Dependency Injection Implementation

- Constructor Injection
- Property Setter
- Method Injection



Demo

Dependency Injection example

Knowledge check

- Advantages of dependency injection
- Name two ways to implement DI

Moq

Moq

- The most popular and friendly mocking framework for .NET
- Mocking creates replacements objects that simulate the behavior of the real ones
- Minimalistic
- Nuget package (https://www.nuget.org/packages/moq/)

Moq

- Use object Mock
- SetUp method to mock operations
- Use Mock to verify behavior

Demo

Using Moq for DI with xUnit

Knowledge check

- Why is it possible to use Moq in Unit Tests

- What method is needed to count number of times a method is called?

Web Performance Tests with Visual Studio

Overview

Web Performance Test Scenarios

Creating a Web Performance Test

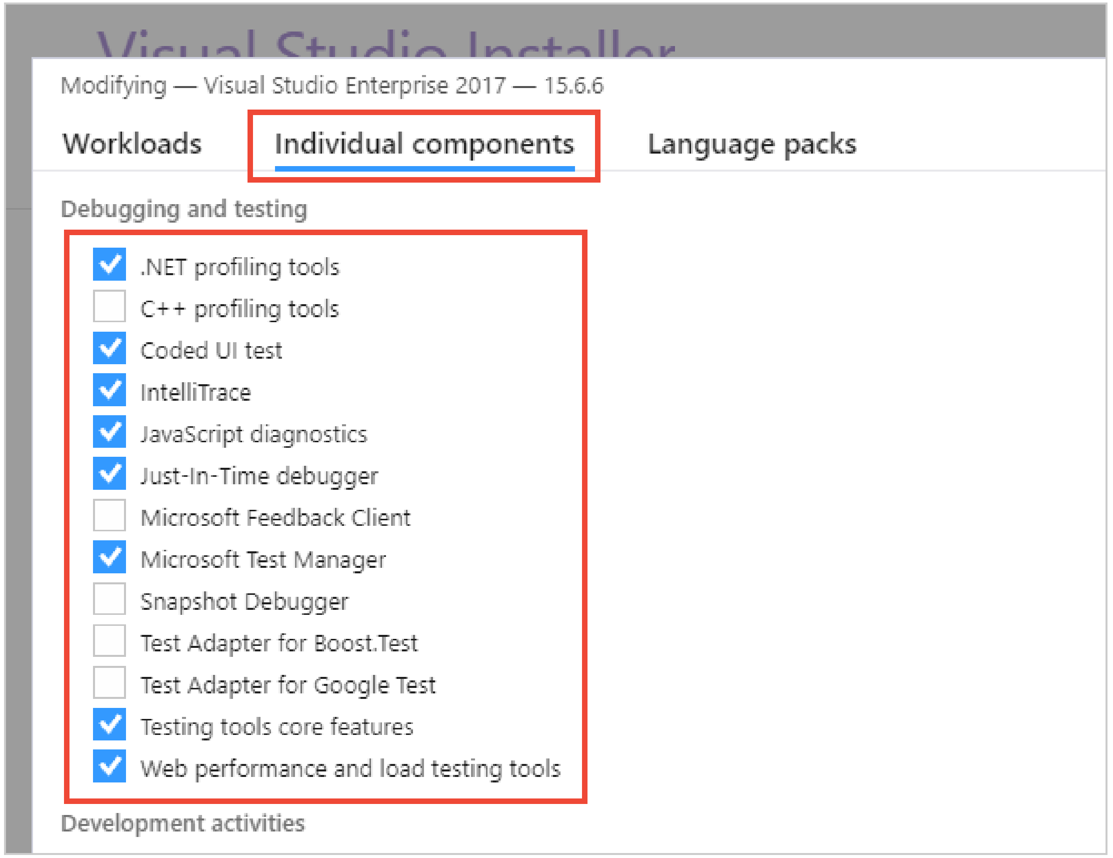
Web Performance Test Viewer

Editing Web Performance Tests

Prerequisites

Visual Studio Enterprise with *Web Performance and Load Testing Tools*

Visual Studio 2019 is the last version where web performance and load testing will be available



Modifying — Visual Studio Enterprise 2017 — 15.6.6

Debugging and testing

- [✓] .NET profiling tools
- [ ] C++ profiling tools
- [✓] Coded UI test
- [✓] IntelliTrace
- [✓] JavaScript diagnostics
- [✓] Just-In-Time debugger
- [ ] Microsoft Feedback Client
- [✓] Microsoft Test Manager
- [ ] Snapshot Debugger
- [ ] Test Adapter for Boost.Test
- [ ] Test Adapter for Google Test
- [✓] Testing tools core features
- [✓] Web performance and load testing tools

Development activities

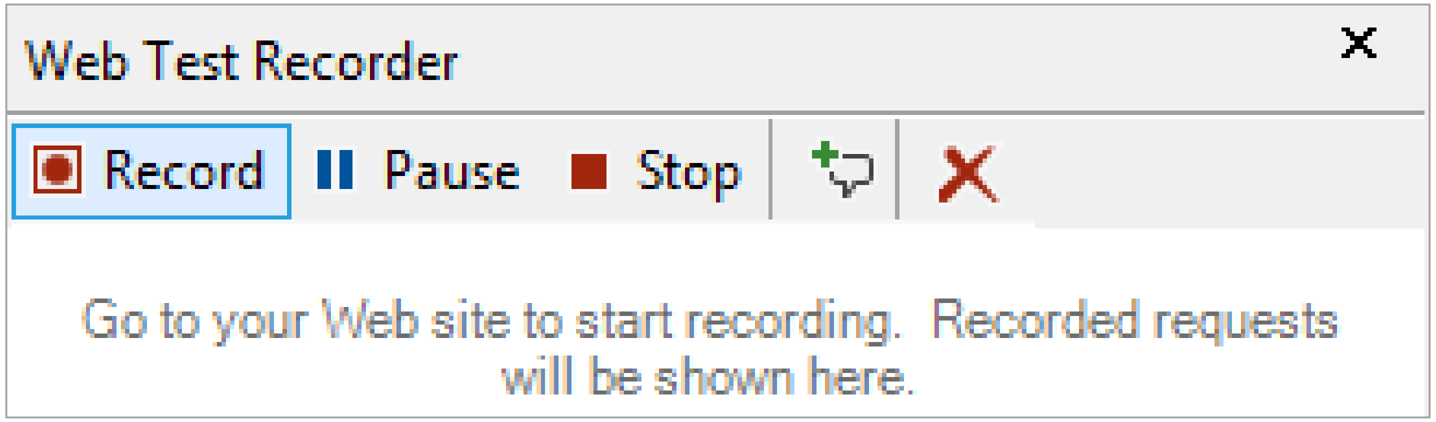
Web Performance Test Scenarios

- Simulates how an end user might interact with a web application
- Records HTTP requests
- Retrieves responses from servers with timing data
- Smoke Test
    - Validation
- Use web performance tests with load tests
    - Stress
    - Performance
    - Capacity Planning

Creating a Web Performance Test

Browse website normally, requests are recorded

Use Web Test Recorder or Fiddler

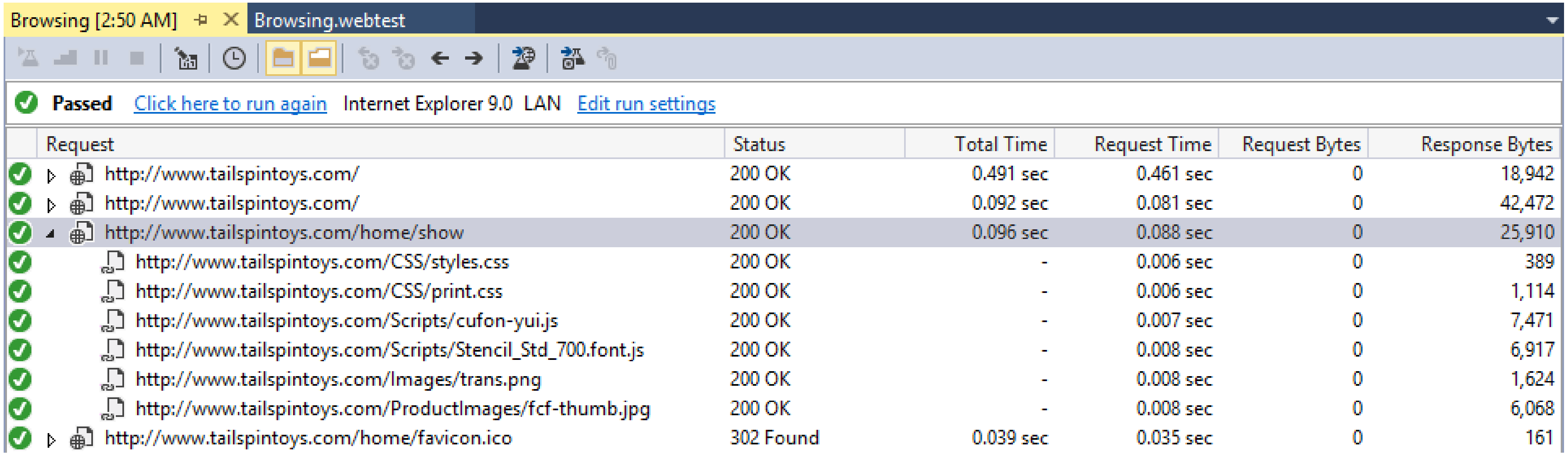


Go to your Web site to start recording. Recorded requests will be shown here.

Web Performance Test Viewer

Primary tool for running a web performance test

View details related to Request and Response



✓ **Passed** Click here to run again Internet Explorer 9.0 LAN Edit run settings

	Request	Status	Total Time	Request Time	Request Bytes	Response Bytes
✓	http://www.tailspintoys.com/	200 OK	0.491 sec	0.461 sec	0	18,942
✓	http://www.tailspintoys.com/	200 OK	0.092 sec	0.081 sec	0	42,472
✓	http://www.tailspintoys.com/home/show	200 OK	0.096 sec	0.088 sec	0	25,910
✓	http://www.tailspintoys.com/CSS/styles.css	200 OK	-	0.006 sec	0	389
✓	http://www.tailspintoys.com/CSS/print.css	200 OK	-	0.006 sec	0	1,114
✓	http://www.tailspintoys.com/Scripts/cufon-yui.js	200 OK	-	0.007 sec	0	7,471
✓	http://www.tailspintoys.com/Scripts/Stencil_Std_700.font.js	200 OK	-	0.008 sec	0	6,917
✓	http://www.tailspintoys.com/Images/trans.png	200 OK	-	0.008 sec	0	1,624
✓	http://www.tailspintoys.com/ProductImages/fcf-thumb.jpg	200 OK	-	0.008 sec	0	6,068
✓	http://www.tailspintoys.com/home/favicon.ico	302 Found	0.039 sec	0.035 sec	0	161

Web Performance Test Viewer – Bottom Pane

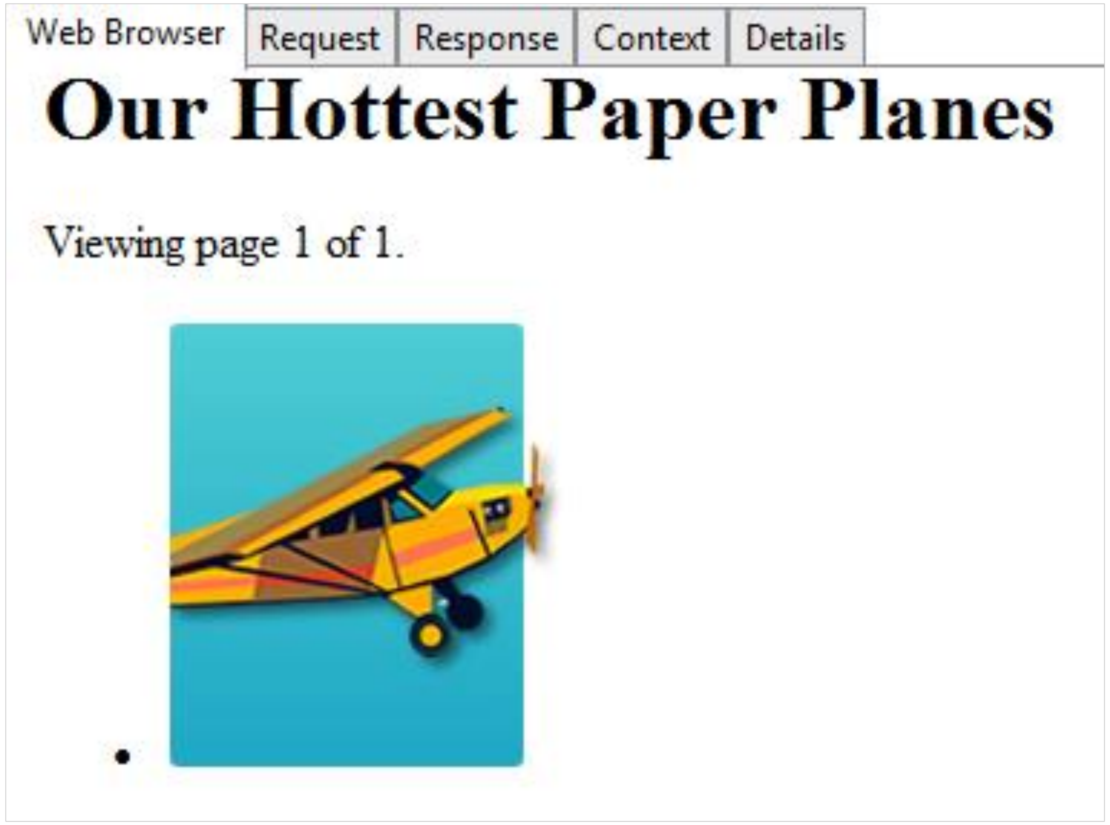
Web Browser: Displays the rendered page returned from the current HTTP request.

Request: Displays the contents of the current HTTP request, in two views, graphical and raw data.

Response: Displays the HTTP response received as a reply to the current HTTP request.

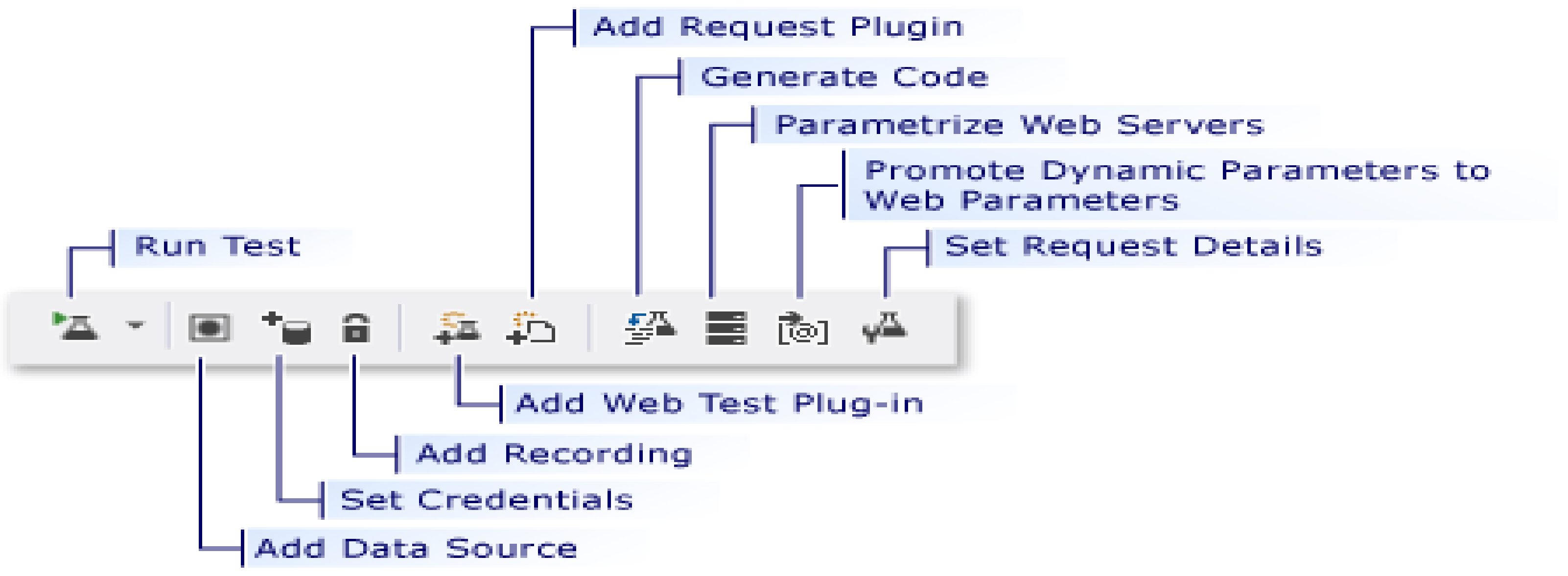
Context: Displays the collection of contexts for a Web performance test. The context collection is a set of name and value pairs that contains important information persisted during a Web performance test.

Details: Displays specific details about the currently displayed webpage, including any validation and extraction rules you have applied, and their results.



Web Performance Test Editor - Overview

Use the Web Performance Test Editor to edit your web performance tests after you have recorded them.

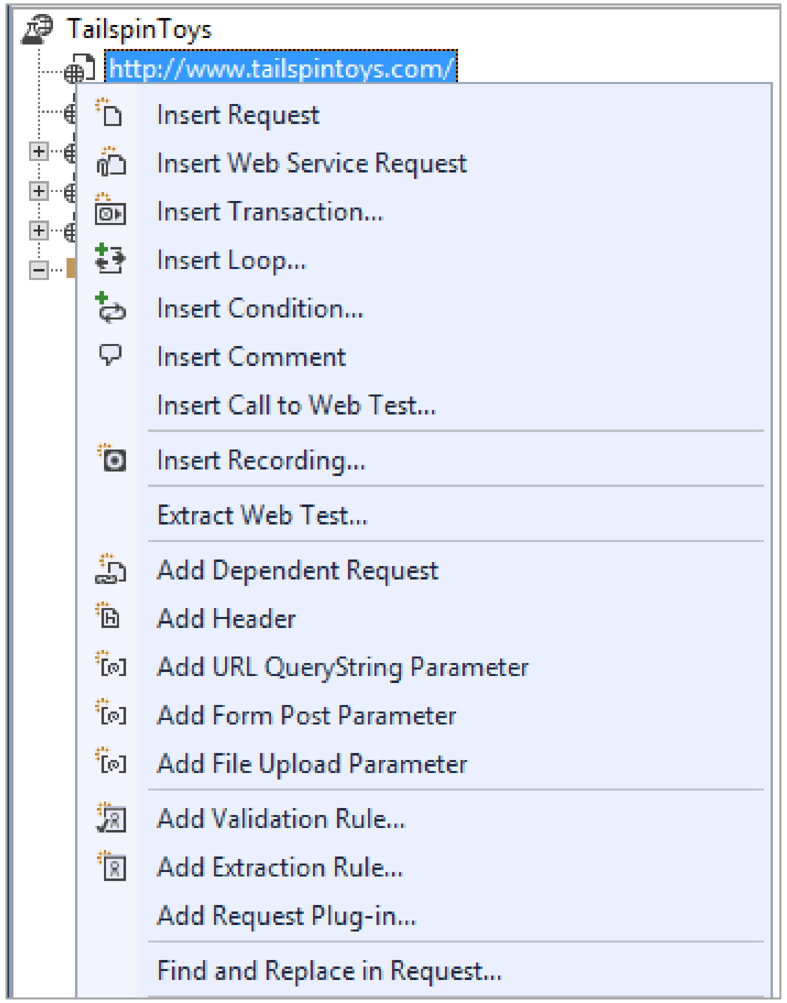


Web Performance Test Editor - Tree Nodes

View the Web Performance Test Editor nodes

- Hierarchy tree of HTTP requests

Right-click a node to view options for editing



TailspinToys
http://www.tailspintoys.com/

Insert Request
Insert Web Service Request
Insert Transaction...
Insert Loop...
Insert Condition...
Insert Comment
Insert Call to Web Test...
Insert Recording...
Extract Web Test...
Add Dependent Request
Add Header
Add URL QueryString Parameter
Add Form Post Parameter
Add File Upload Parameter
Add Validation Rule...
Add Extraction Rule...
Add Request Plug-in...
Find and Replace in Request...

Context Parameters

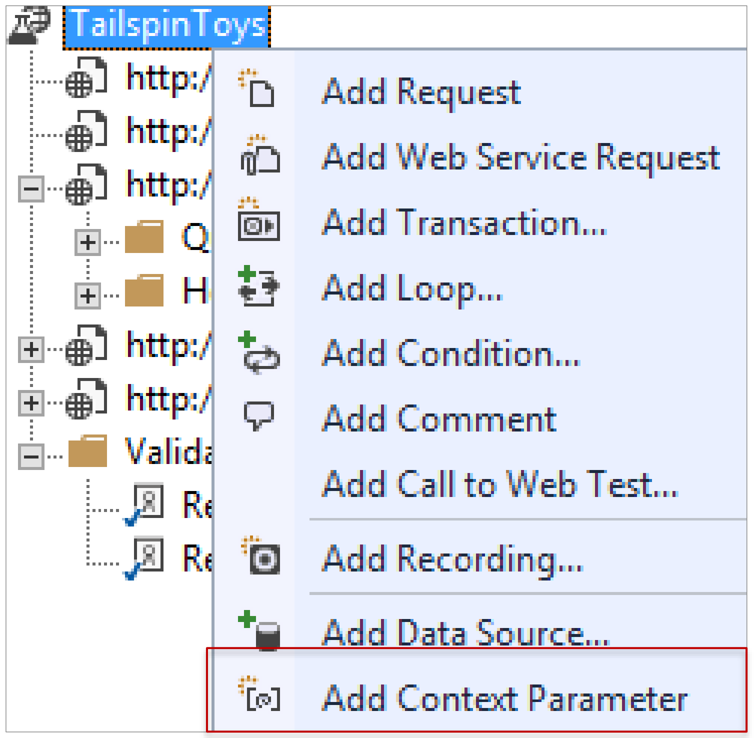
Allows you to parameterize a string

Example: Parameterize web servers

- Move tests into different environments such as Developer, Quality Assurance, and Production

Set context parameters:

- Manually
- Extraction rules
- Load Test Run Settings
- Environment variables



Dynamic Parameters

A dynamic parameter is a parameter whose value is regenerated every time that a user runs the application. Example: Session ID.

The web performance test recorder and playback engine automatically handles the most common types of dynamic parameters:

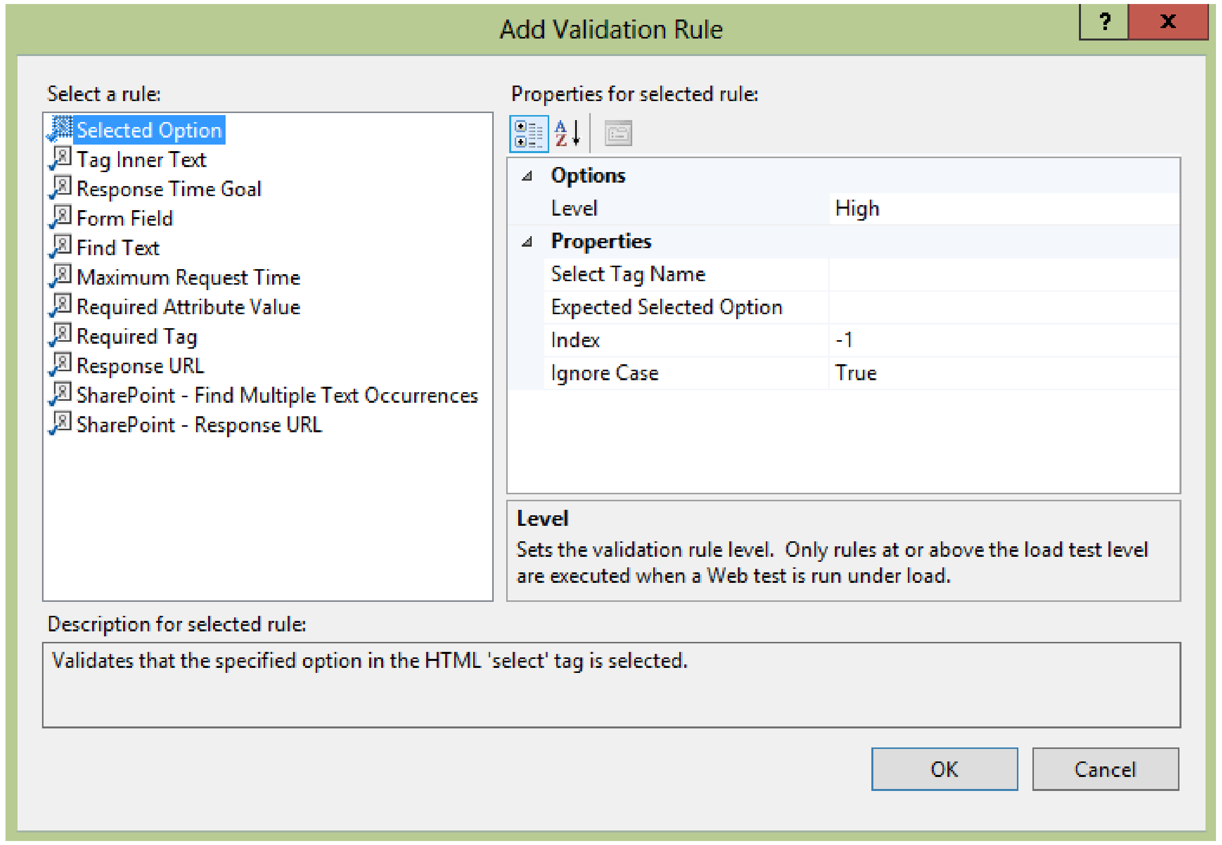
- Values that are set in a cookie value
- Values that are set in hidden fields on HTML pages, such as ASP.NET view state
- Values that are set as query string or form post parameters

For dynamic parameters that are not detected, use Extraction Rules

Validation Rules

Verify by validating text, tag, and attributes

Use built-in rules or create your own



Add Validation Rule

Select a rule:

- Selected Option
- Tag Inner Text
- Response Time Goal
- Form Field
- Find Text
- Maximum Request Time
- Required Attribute Value
- Required Tag
- Response URL
- SharePoint - Find Multiple Text Occurrences
- SharePoint - Response URL

Properties for selected rule:

Options	
Level	High
Properties	
Select Tag Name	
Expected Selected Option	
Index	-1
Ignore Case	True

Level
Sets the validation rule level. Only rules at or above the load test level are executed when a Web test is run under load.

Description for selected rule:

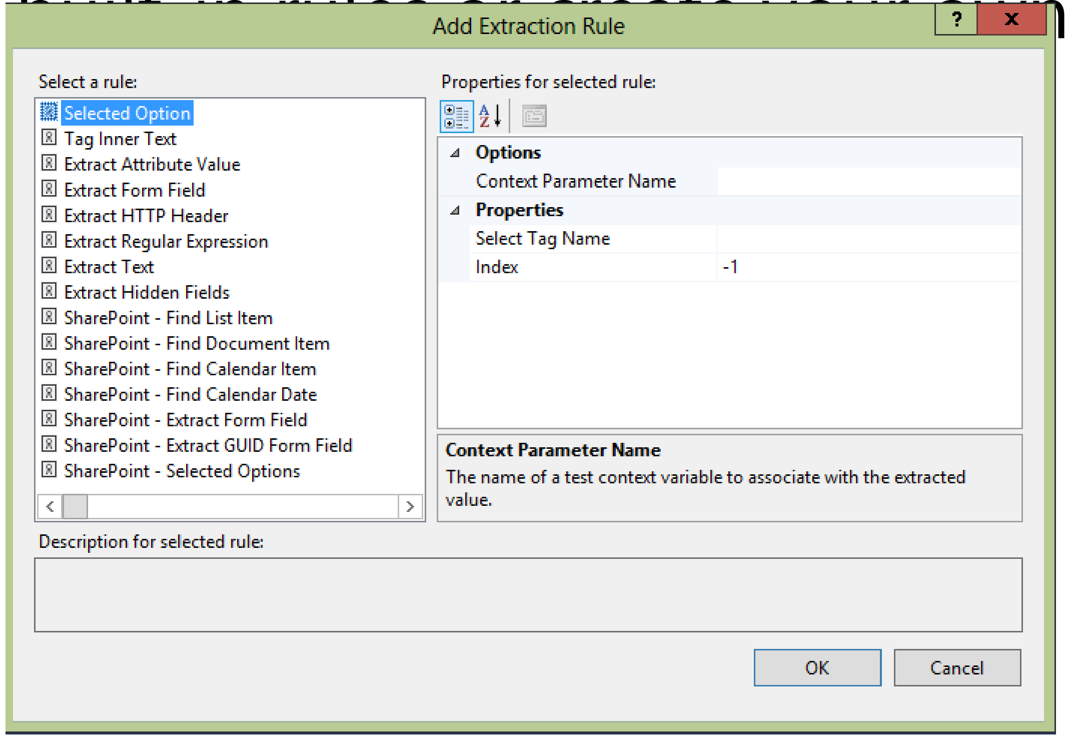
Validates that the specified option in the HTML 'select' tag is selected.

Extraction Rules

Extracts data from the responses

Stores results in context parameter as name/value pairs

Use built-in rules or create your own



Add Extraction Rule

Select a rule:

- Selected Option
- Tag Inner Text
- Extract Attribute Value
- Extract Form Field
- Extract HTTP Header
- Extract Regular Expression
- Extract Text
- Extract Hidden Fields
- SharePoint - Find List Item
- SharePoint - Find Document Item
- SharePoint - Find Calendar Item
- SharePoint - Find Calendar Date
- SharePoint - Extract Form Field
- SharePoint - Extract GUID Form Field
- SharePoint - Selected Options

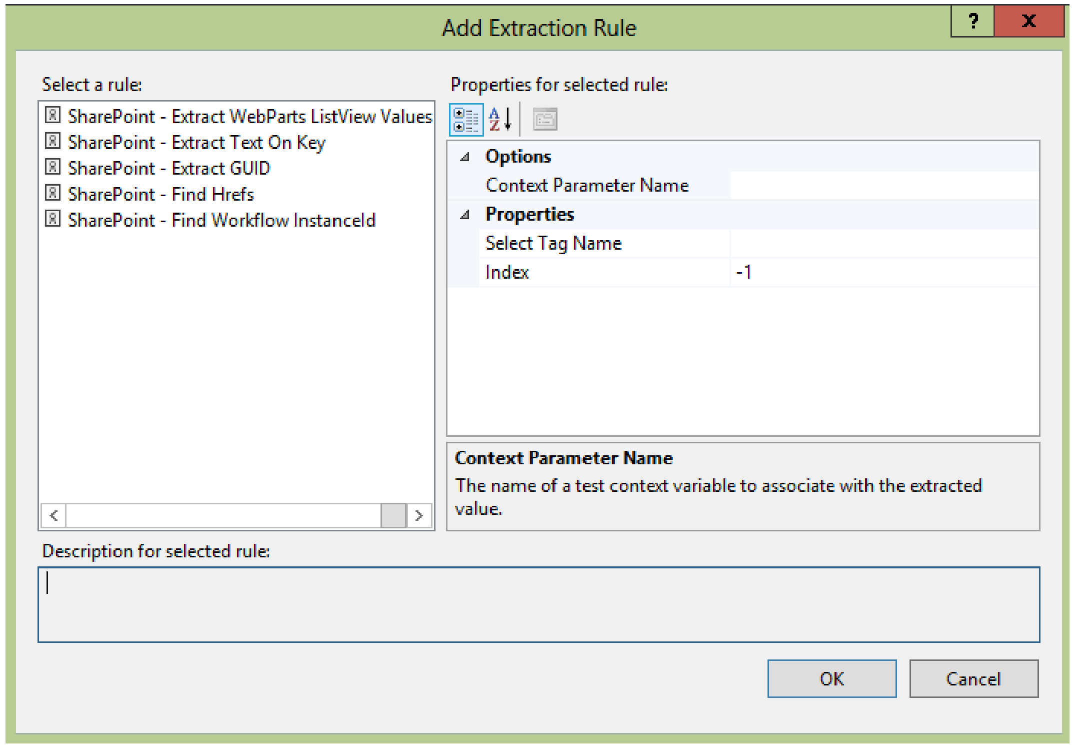
Properties for selected rule:

Options	
Context Parameter Name	
Properties	
Select Tag Name	
Index	-1

Context Parameter Name
The name of a test context variable to associate with the extracted value.

Description for selected rule:

OK Cancel



Add Extraction Rule

Select a rule:

- SharePoint - Extract WebParts ListView Values
- SharePoint - Extract Text On Key
- SharePoint - Extract GUID
- SharePoint - Find Hrefs
- SharePoint - Find Workflow InstanceId

Properties for selected rule:

Options	
Context Parameter Name	
Properties	
Select Tag Name	
Index	-1

Context Parameter Name
The name of a test context variable to associate with the extracted value.

Description for selected rule:

OK Cancel

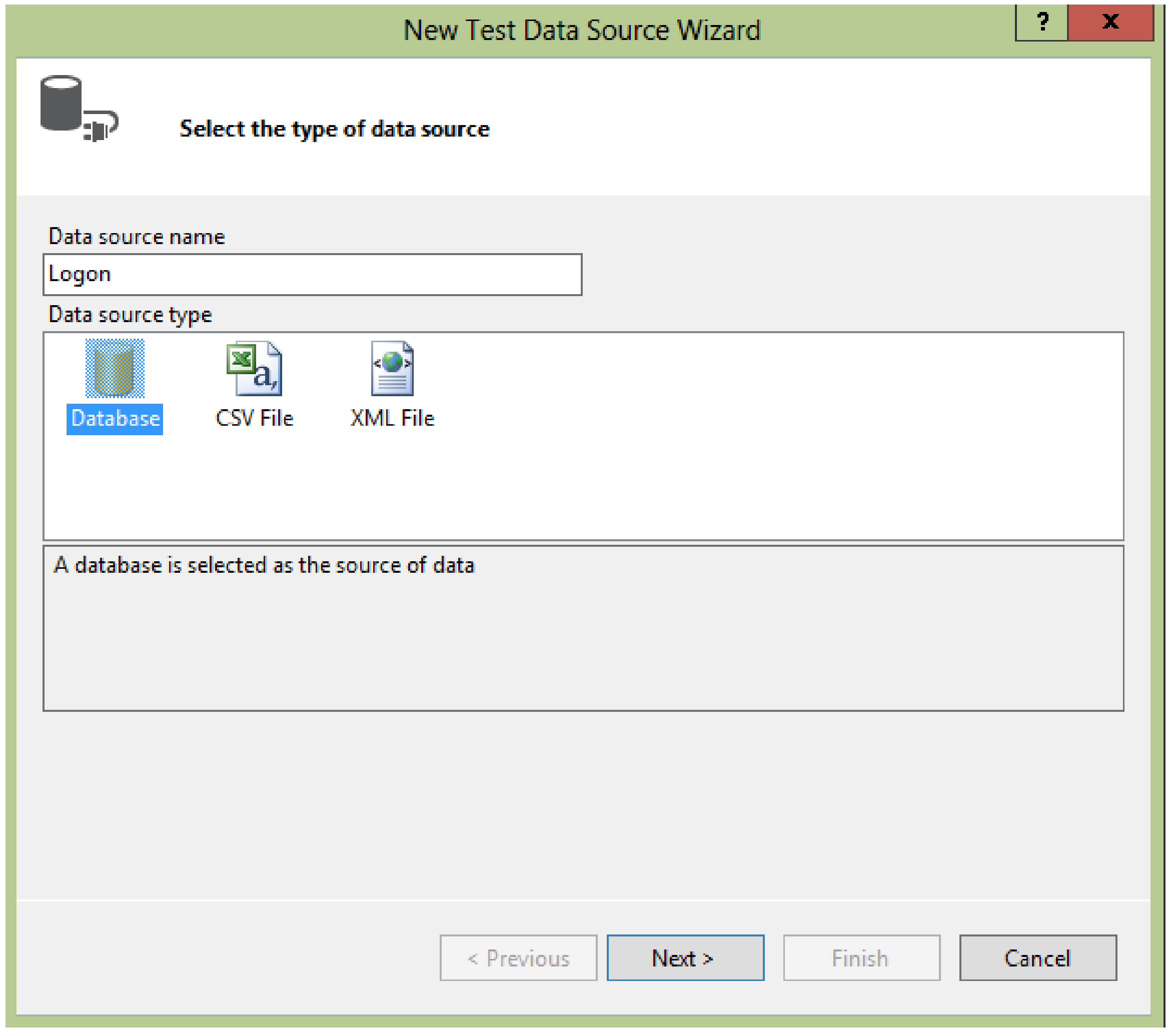
Data Binding Web Performance Test

Provides input to HTTP requests

- Credentials (usernames and passwords)
- Query string parameters
- Form field parameters
- Request URL

Data source

- Object Linking and Embedding Database (OLE DB)
- Comma-separated value file (CSV), XML file, Microsoft Excel file, Microsoft Access database or Microsoft SQL Server database


New Test Data Source Wizard

Select the type of data source

Data source name: Logon

Data source type: Database

A database is selected as the source of data

Use data binding for:

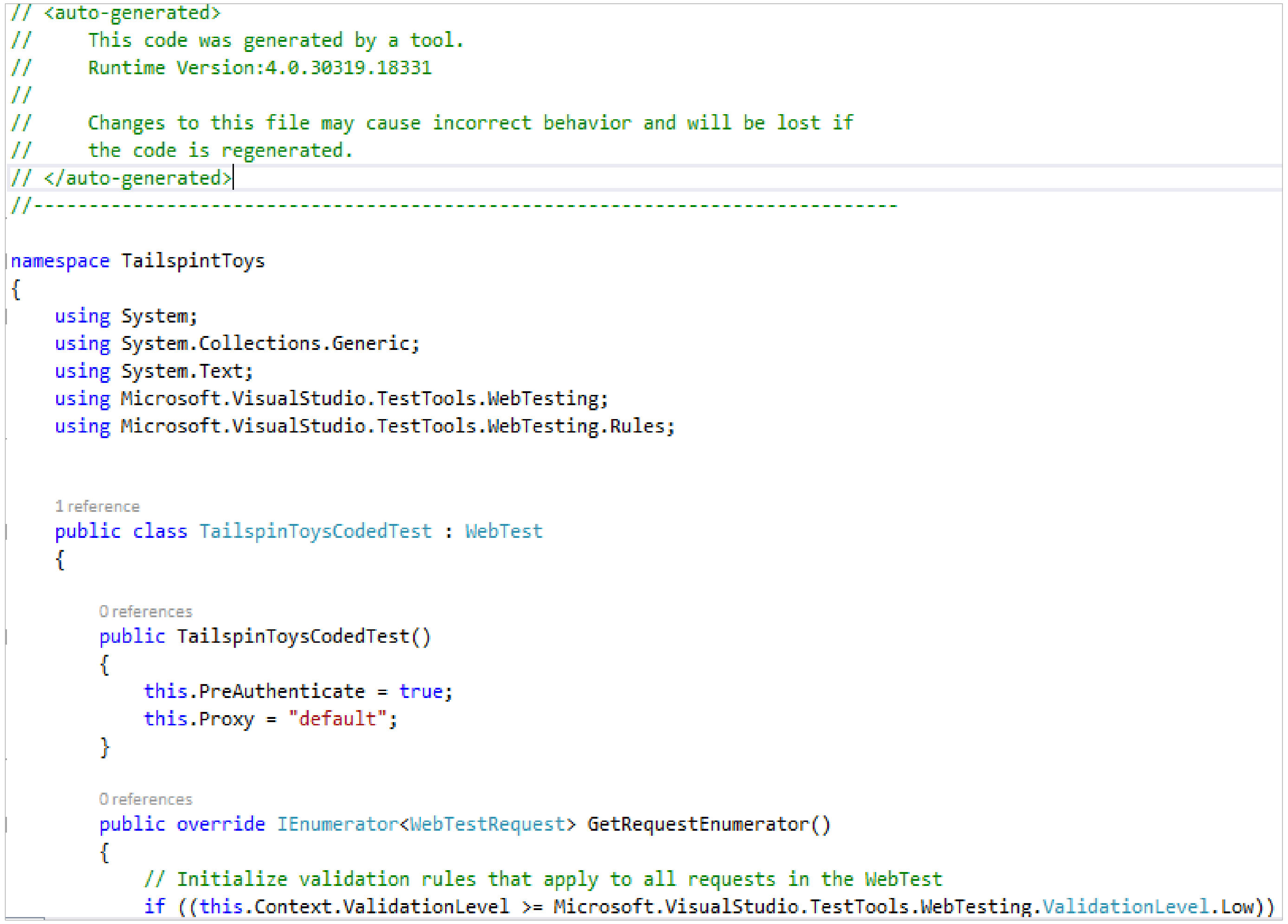
Coded Web Performance Test

Convert existing recorded test to coded web performance test

Create and edit manually

Scriptable in C# and Visual Basic .NET

Allows for flexible scenarios



```
// <auto-generated>
//     This code was generated by a tool.
//     Runtime Version:4.0.30319.18331
//
//     Changes to this file may cause incorrect behavior and will be lost if
//     the code is regenerated.
// </auto-generated>
//------------------------------------------------------------------------------

namespace TailspintToys
{
    using System;
    using System.Collections.Generic;
    using System.Text;
    using Microsoft.VisualStudio.TestTools.WebTesting;
    using Microsoft.VisualStudio.TestTools.WebTesting.Rules;


    1 reference
    public class TailspinToysCodedTest : WebTest
    {

        0 references
        public TailspinToysCodedTest()
        {
            this.PreAuthenticate = true;
            this.Proxy = "default";
        }

        0 references
        public override IEnumerator<WebTestRequest> GetRequestEnumerator()
        {
            // Initialize validation rules that apply to all requests in the WebTest
            if ((this.Context.ValidationLevel >= Microsoft.VisualStudio.TestTools.WebTesting.ValidationLevel.Low))
```

Web Performance Tests: Advanced Topics

- Insert
  - Loop
  - Condition
  - Web Service Request
  - Transaction
- Extract Web Test
- If/then branch conditions
- Run test from command line
- Debug via Results Viewer
- Custom plug-ins

Demo 1: Creating a Web Performance Test in Visual Studio

Lesson Knowledge Check

1. Name two ways to create a web performance test
2. Name two scenarios where you might use a web performance test
3. Name two data sources that can be bound to a web performance test

Microsoft Azure