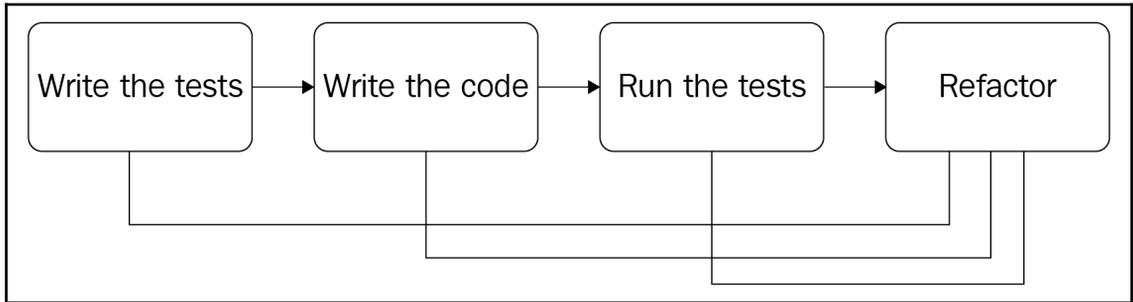
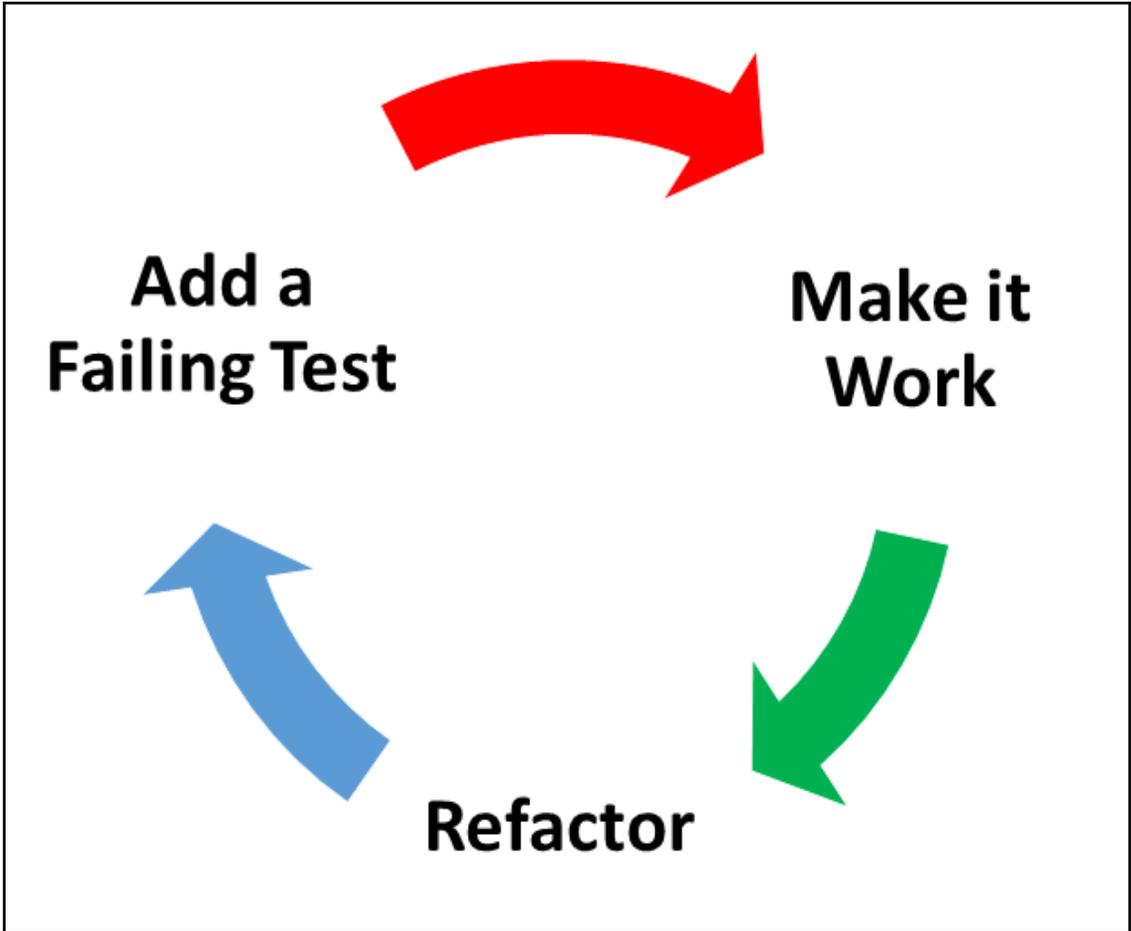
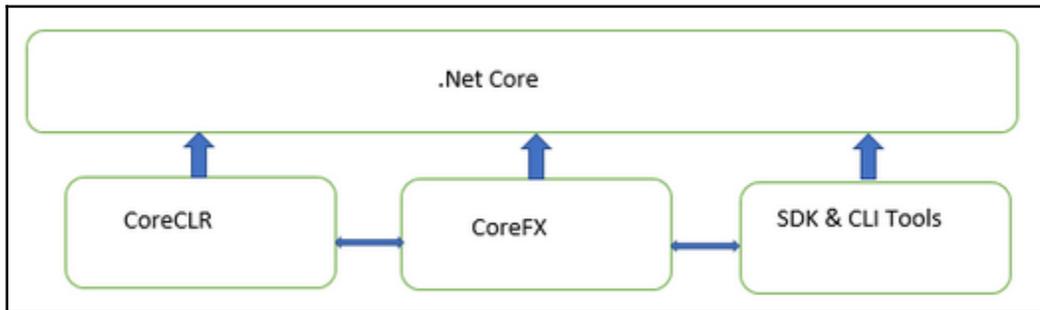
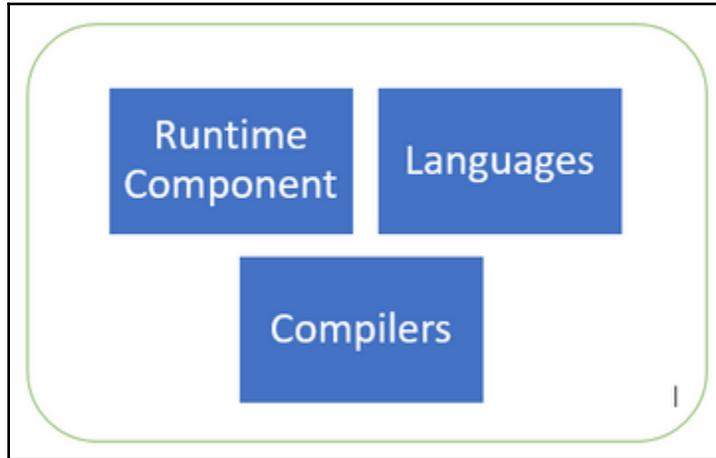


# Chapter 1: Exploring Test-Driven Development





## Chapter 2: Getting Started with .NET Core



```
ayobami@ayobami-VirtualBox: ~
Unpacking dotnet-runtime-2.0.0 (2.0.0-1) ...
Selecting previously unselected package dotnet-sdk-2.0.2.
Preparing to unpack .../dotnet-sdk-2.0.2_2.0.2-1_amd64.deb ...
Unpacking dotnet-sdk-2.0.2 (2.0.2-1) ...
Setting up dotnet-host (2.0.0-1) ...
Setting up aspnetcore-store-2.0.0 (2.0.0-1) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up dotnet-hostfxr-2.0.0 (2.0.0-1) ...
Setting up dotnet-runtime-2.0.0 (2.0.0-1) ...
Setting up dotnet-sdk-2.0.2 (2.0.2-1) ...
This software may collect information about you and your use of the software, and send that to Microsoft.
Please visit http://aka.ms/dotnet-cli-eula for more information.
Welcome to .NET Core!
-----
Learn more about .NET Core @ https://aka.ms/dotnet-docs. Use dotnet --help to see available commands or go to https://aka.ms/dotnet-cli-docs.

.NET Core Tools Telemetry
-----
The .NET Core Tools include a telemetry feature that collects usage information. It is important that the .NET Team understands how the tools are being used so that we can improve them.

The data collected is anonymous and will be published in an aggregated form for use by both Microsoft and community engineers under the Creative Commons Attribution License.

The .NET Core Tools telemetry feature is enabled by default. You can opt-out of the telemetry feature by setting an environment variable DOTNET_CLI_TELEMETRY_OPTOUT (for example, 'export' on macOS/Linux, 'set' on Windows) to true (for example, 'true', '1'). You can read more about .NET Core tools telemetry at https://aka.ms/dotnet-cli-telemetry.

Installation Note
-----
A command will be run during the install process that will improve project restore speed and enable offline access. It will take up to a minute to complete.
ayobami@ayobami-VirtualBox:~$
```

```
ayobami@ayobami-VirtualBox:~/Documents/testapp
ayobami@ayobami-VirtualBox:/$ cd /home/ayobami/Documents/testapp/
ayobami@ayobami-VirtualBox:~/Documents/testapp$ dotnet new console

Welcome to .NET Core!
-----
Learn more about .NET Core @ https://aka.ms/dotnet-docs. Use dotnet --help to see available commands or go to https://aka.ms/dotnet-cli-docs.

Telemetry
-----
The .NET Core tools collect usage data in order to improve your experience. The data is anonymous and does not include command-line arguments. The data is collected by Microsoft and shared with the community.
You can opt out of telemetry by setting a DOTNET_CLI_TELEMETRY_OPTOUT environment variable to 1 using your favorite shell.
You can read more about .NET Core tools telemetry @ https://aka.ms/dotnet-cli-telemetry.
Getting ready...
The template "Console Application" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on /home/ayobami/Documents/testapp/testapp.csproj...
  Restoring packages for /home/ayobami/Documents/testapp/testapp.csproj...
  Generating MSBuild file /home/ayobami/Documents/testapp/obj/testapp.csproj.nuget.g.props.
  Generating MSBuild file /home/ayobami/Documents/testapp/obj/testapp.csproj.nuget.g.targets.
  Restore completed in 300.02 ms for /home/ayobami/Documents/testapp/testapp.csproj.

Restore succeeded.

ayobami@ayobami-VirtualBox:~/Documents/testapp$
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL C#
Updating C# dependencies...
Platform: linux, x86_64, name=ubuntu, version=17.04

Downloading package 'OmniSharp for Linux (x64)' (24595 KB) ..... Done!
Downloading package '.NET Core Debugger (linux / x64)' (54950 KB) ..... Done!

Installing package 'OmniSharp for Linux (x64)'
Installing package '.NET Core Debugger (linux / x64)'

Finished
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash
ayobami@ayobami-VirtualBox:~$
```

```
Program.cs x
1 using System;
2
3 namespace testapp
4 {
5     0 references
6     class Program
7     {
8         0 references
9         static void Main(string[] args)
10        {
11            Console.WriteLine("Hello World!");
12        }
13    }

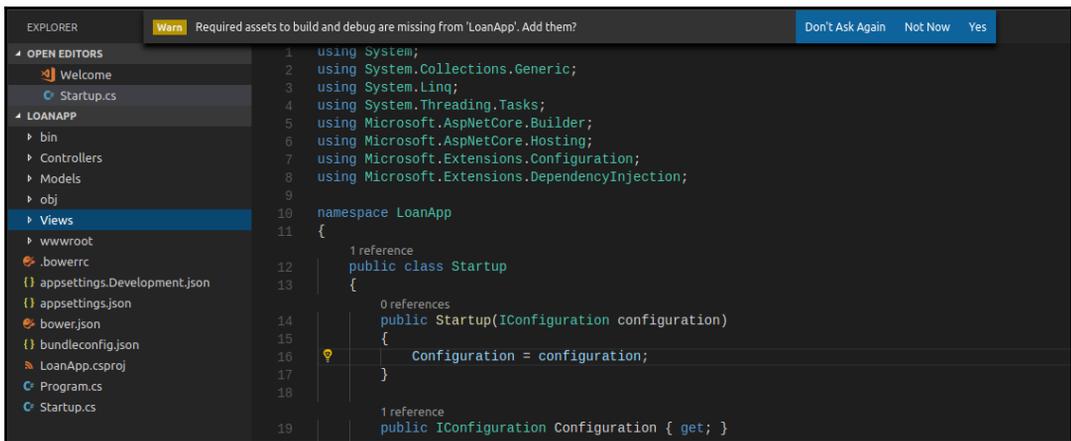
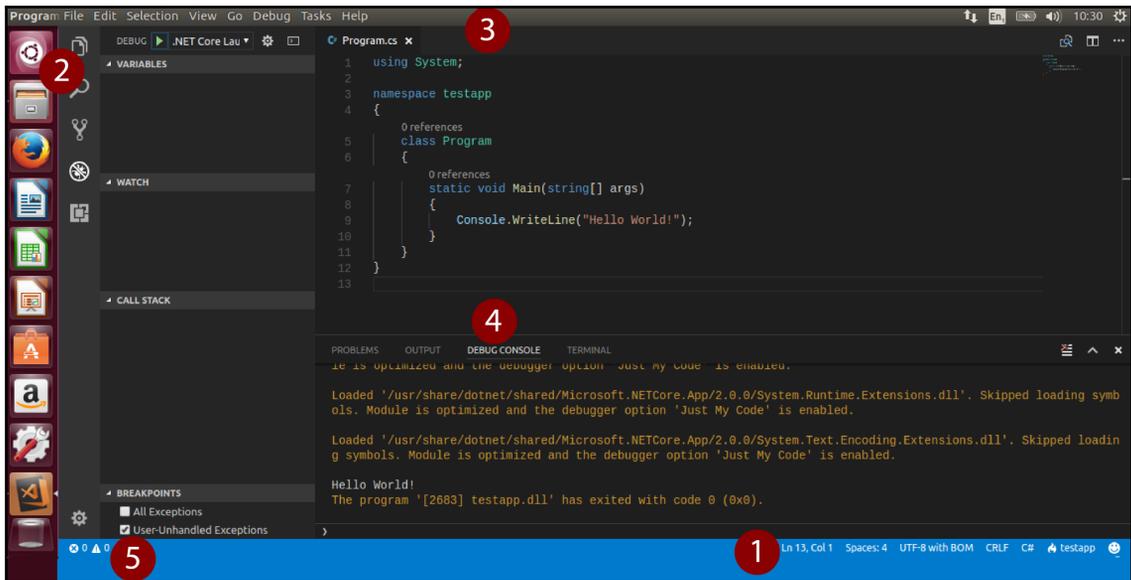
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
The program is optimized and the debugger option 'Just My Code' is enabled.

Loaded '/usr/share/dotnet/shared/Microsoft.NETCore.App/2.0.0/System.Runtime.Extensions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.

Loaded '/usr/share/dotnet/shared/Microsoft.NETCore.App/2.0.0/System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.

Hello World!
The program '[2683] testapp.dll' has exited with code 0 (0x0).
```



Firefox Web Browser

Home Page - LoanApp - Mozilla Firefox

Home Page - LoanApp x +

localhost:5000

LoanApp Home About Contact

# Microsoft Azure

Learn how Microsoft's Azure cloud platform allows you to build, deploy, and scale web apps. [Learn More](#)

## Application uses

- Sample pages using ASP.NET Core MVC
- [Bower](#) for managing client-side libraries
- Theming using [Bootstrap](#)

## How to

- Add a Controller and View
- Manage User Secrets using Secret Manager.
- Use logging to log a message.
- Add packages using

## Overview

- Conceptual overview of what is ASP.NET Core
- Fundamentals of ASP.NET Core such as Startup and middleware.
- Working with Data

## Run & Deploy

- Run your app
- Run tools such as EF migrations and more
- Publish to Microsoft Azure Web Apps

# Chapter 3: Writing Testable Code

No images

# Chapter 4: .NET Core Unit Testing

Test Explorer

TestCheckWordLength\_ShouldReturnBoolean (1 test) Failed: 1 test failed

- TestCheckWordLength\_ShouldReturnBoolean(word: "name", ???: 67) Failed: System.InvalidOperationException : The test method expected 1 parameter value, but 2 parameter values were provided.

Test Explorer

Run All | Run... | Playlist: All Tests

Failed Tests (1)

Passed Tests (4)

- LoanApplication.Tests.Unit.TheoryTest.TestCheckWordLength\_ShouldReturnBoolean(word: "city") 12 ms
- LoanApplication.Tests.Unit.TheoryTest.TestCheckWordLength\_ShouldReturnBoolean(word: "name") 1 ms
- Extensions.PalindromeCheckerTest.Test\_IsWordPalindrome\_ShouldReturnTrue 61 ms
- LoanApplication.Tests.Unit.GCDTest.Test\_GetGcd\_ShouldReturnTrue(firstNumber: 8, secondNumber: 12, gcd: 4) 5 ms

Not Run Tests (24)

Summary

Last Test Run Passed (Total Run Time 0:00:02.6315874)

2 Tests Passed

```
[Theory,
InlineData(new LoanDTO { LoanType = LoanType.CarLoan, JobType = JobType.Professional, LocationType = LocationType.Location1 })]
public void Test_CalculateLoan_ShouldReturnCorrectRate(Loan loan)
{
    Loan loan = carLoanCalculator.CalculateLoan(loanDTO);
    Assert.NotNull(loan);
    Assert.Equal(8, loan.InterestRate);
}
```

An attribute argument must be a constant expression, typeof expression or array creation expression of an attribute parameter type

An attribute argument must be a constant expression, typeof expression or array creation expression of an attribute parameter type

## Extensions.PalindromeCheckerTest.Test\_IsWordPalindrome\_ShouldReturnTrue [Copy All](#)

Source: [PalindromeCheckerTest.cs](#) line 14

- Test Passed - Extensions.PalindromeCheckerTest.Test\_IsWordPalindrome\_ShouldReturnTrue(word: "civic")  
Elapsed time: 0:00:00.059
- Test Passed - Extensions.PalindromeCheckerTest.Test\_IsWordPalindrome\_ShouldReturnTrue(word: "dad")  
Elapsed time: 0:00:00.001
- Test Passed - Extensions.PalindromeCheckerTest.Test\_IsWordPalindrome\_ShouldReturnTrue(word: "omo")  
Elapsed time: 0:00:00.001

## Extensions.PalindromeCheckerTest.Test\_IsWordPalindrome\_ShouldReturnTrue

[Copy All](#)

Source: [PalindromeCheckerTest.cs](#) line 14

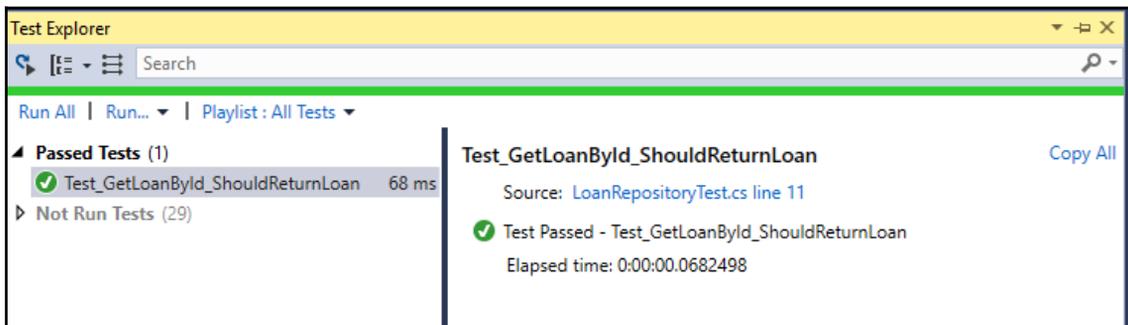
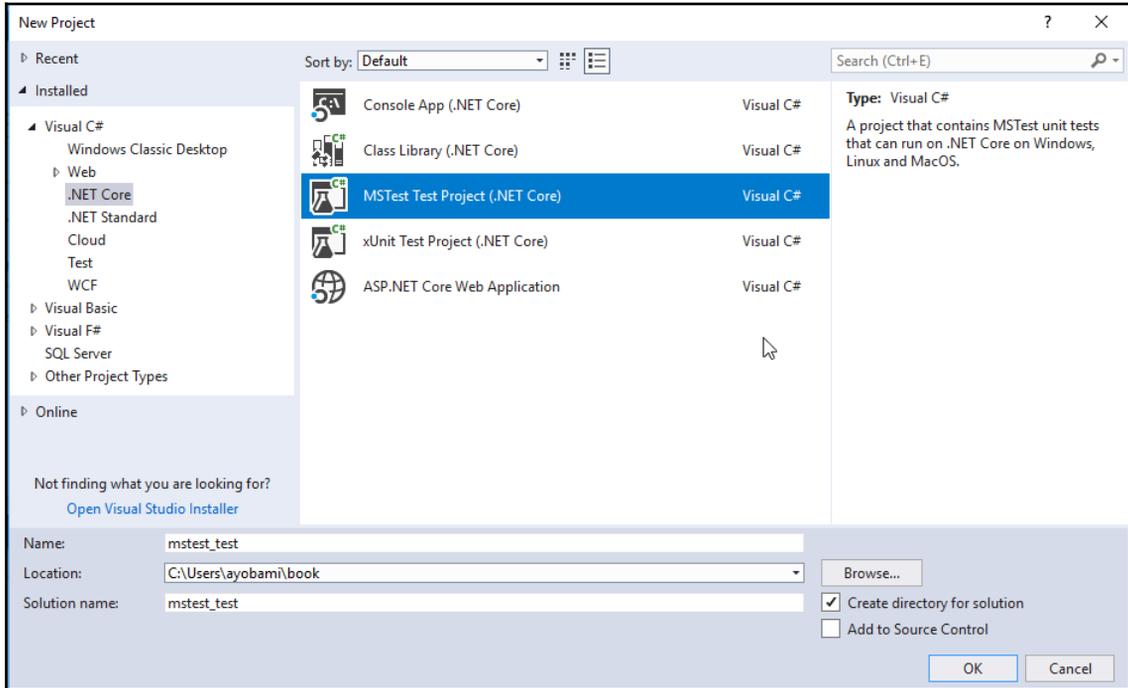
- ✓ Test Passed - Extensions.PalindromeCheckerTest.Test\_IsWordPalindrome\_ShouldReturnTrue(word: "civic")  
Elapsed time: 0:00:00.059
- ✓ Test Passed - Extensions.PalindromeCheckerTest.Test\_IsWordPalindrome\_ShouldReturnTrue(word: "dad")  
Elapsed time: 0:00:00.001
- ✓ Test Passed - Extensions.PalindromeCheckerTest.Test\_IsWordPalindrome\_ShouldReturnTrue(word: "omo")  
Elapsed time: 0:00:00.001

The screenshot shows the Test Explorer window with a search bar and a red header bar. Below the header, there are controls for 'Run All', 'Run...', and 'Playlist: All Tests'. The test results are organized into two columns: 'Failed Tests (2)' and 'Passed Tests (1)'. The 'Failed Tests' column lists two failed tests with their respective durations. The 'Passed Tests' column lists one passed test with its duration. A 'Summary' panel on the right provides an overview of the test run, indicating that the last test run failed, with a total run time of 0:00:02.2890969. The summary shows 2 tests failed and 1 test passed.

Test Name	Duration	Status
LoanApplication.Tests.Unit.GCDTest.Test_GetGcd_ShouldReturnTrue(firstNumber: 36, secondNumber: 20, gcd: 5)	20 ms	Failed
LoanApplication.Tests.Unit.GCDTest.Test_GetGcd_ShouldReturnTrue(firstNumber: 36, secondNumber: 28, gcd: 2)	1 ms	Failed
LoanApplication.Tests.Unit.GCDTest.Test_GetGcd_ShouldReturnTrue(firstNumber: 8, secondNumber: 12, gcd: 4)	5 ms	Passed

**Summary**  
Last Test Run Failed (Total Run Time 0:00:02.2890969)  
2 Tests Failed  
1 Test Passed

# Chapter 5: Data-Driven Unit Tests



```
Administrator: Command Prompt
C:\Users\...LoanApplication\MsTest>dotnet test
Build started, please wait...
Build completed.

Test run for C:\Users\...LoanApplication\MsTest\bin\Debug\netcoreapp2.0\MsTest.dll (.NETCoreApp,Version=v2.0)
Microsoft (R) Test Execution Command Line Tool Version 15.3.0-preview-20170628-02
Copyright (c) Microsoft Corporation. All rights reserved.

Starting test execution, please wait...

Total tests: 1. Passed: 1. Failed: 0. Skipped: 0.
Test Run Successful.
Test execution time: 1.2508 Seconds
```

Extensions and Updates

Sort by: Relevance

nunit

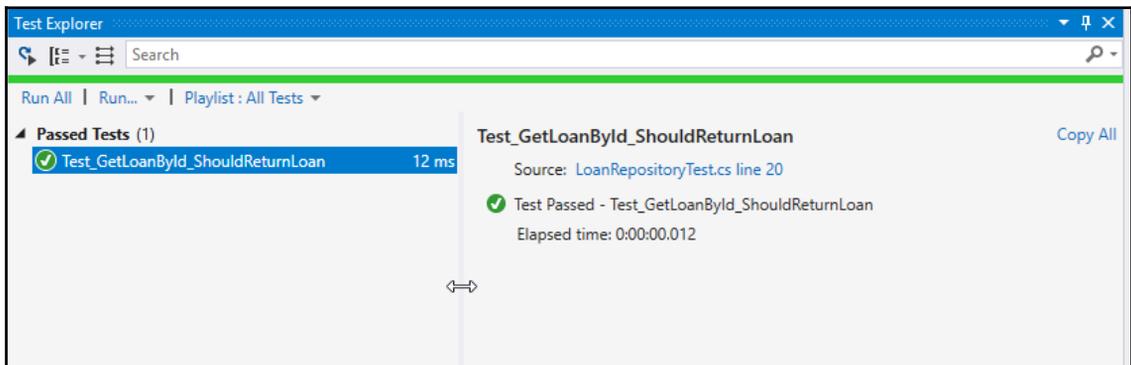
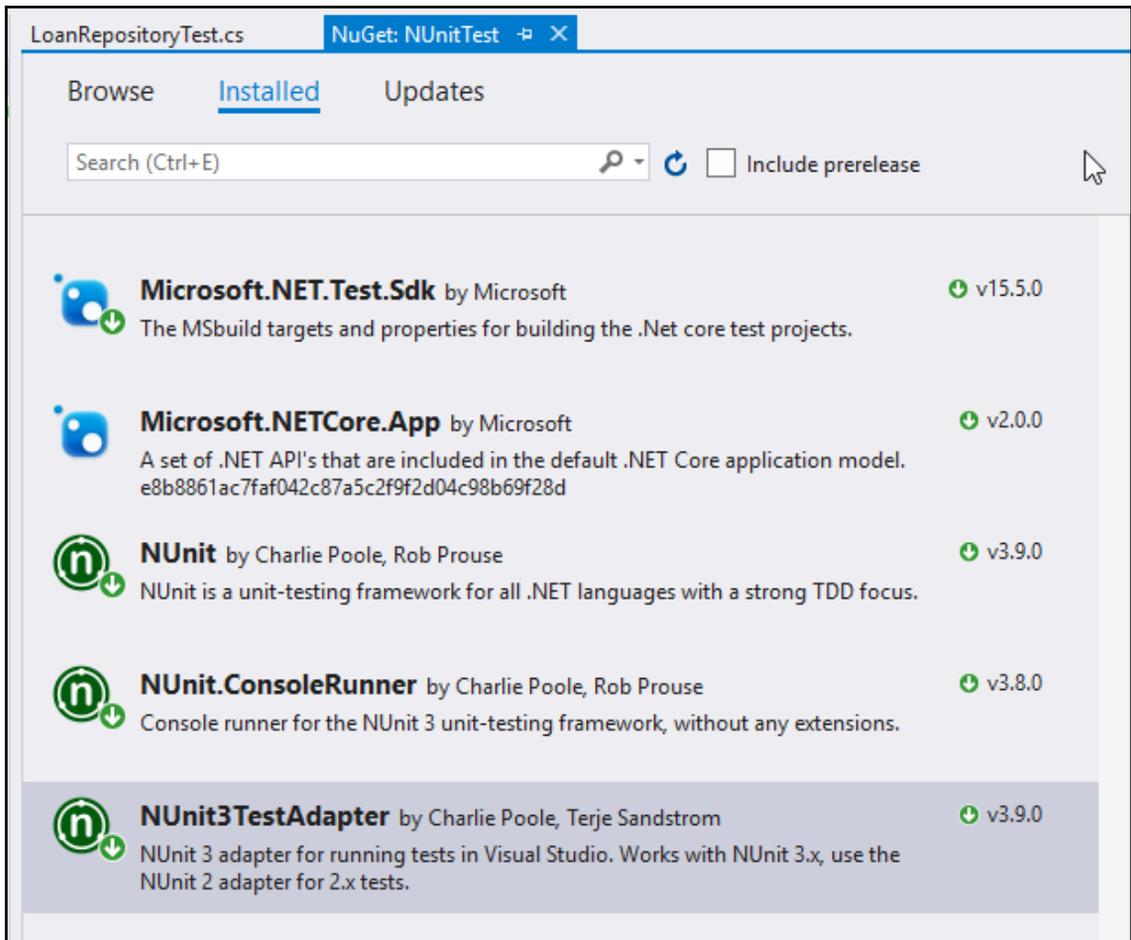
- NUnit 3 Test Adapter**  
NUnit 3 adapter for running tests in Visual Studio.  
Works with NUnit 3.x. [Download](#)
- NUnit 2 Test Adapter**  
NUnit 2 adapter for running tests in Visual Studio 2012 and newer.  
Works with NUnit 2.x, for 3.x tests use the NUnit 3 adapter.
- NUnit VS Templates**  
Provides Visual Studio project and item templates for NUnit 3 along with code snippets.
- Test Generator NUnit extension**  
Test Generator, NUnit extensions for Visual Studio.  
Creates Unit tests and Intellitests with both NUnit 2.6.4 and NUnit 3...
- NUnit Migrator**  
Helps transform NUnit v2 to NUnit v3. It consists of a set of Roslyn analyzers and code fixers.
- Unit Test Boilerplate Generator**  
Generates a unit test boilerplate from a given C# class, setting up mocks for all dependencies. Right click an item in Solution Explorer...
- SwitchToNUnit3**

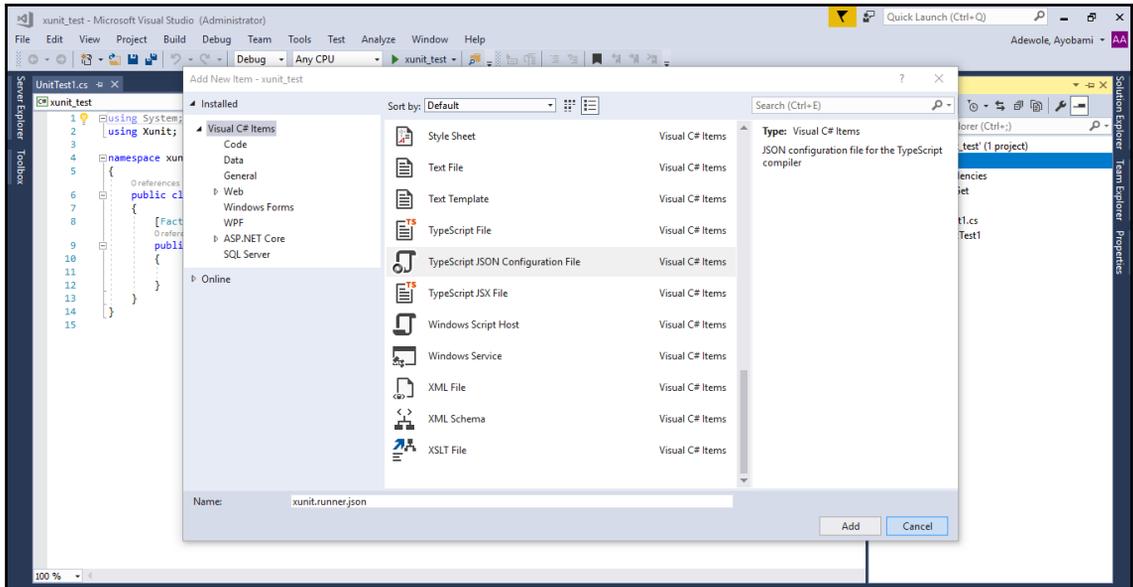
Created by: NUnit Developers  
Version: 3.9.0.0  
Downloads: 533220  
Rating: ★★★★★ (11 Votes)  
[More Information](#)  
[Report Extension to Microsoft](#)

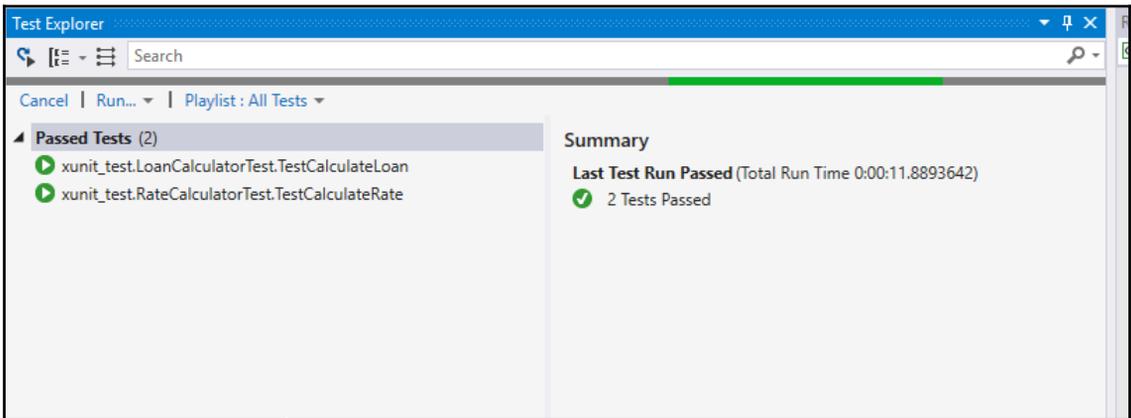
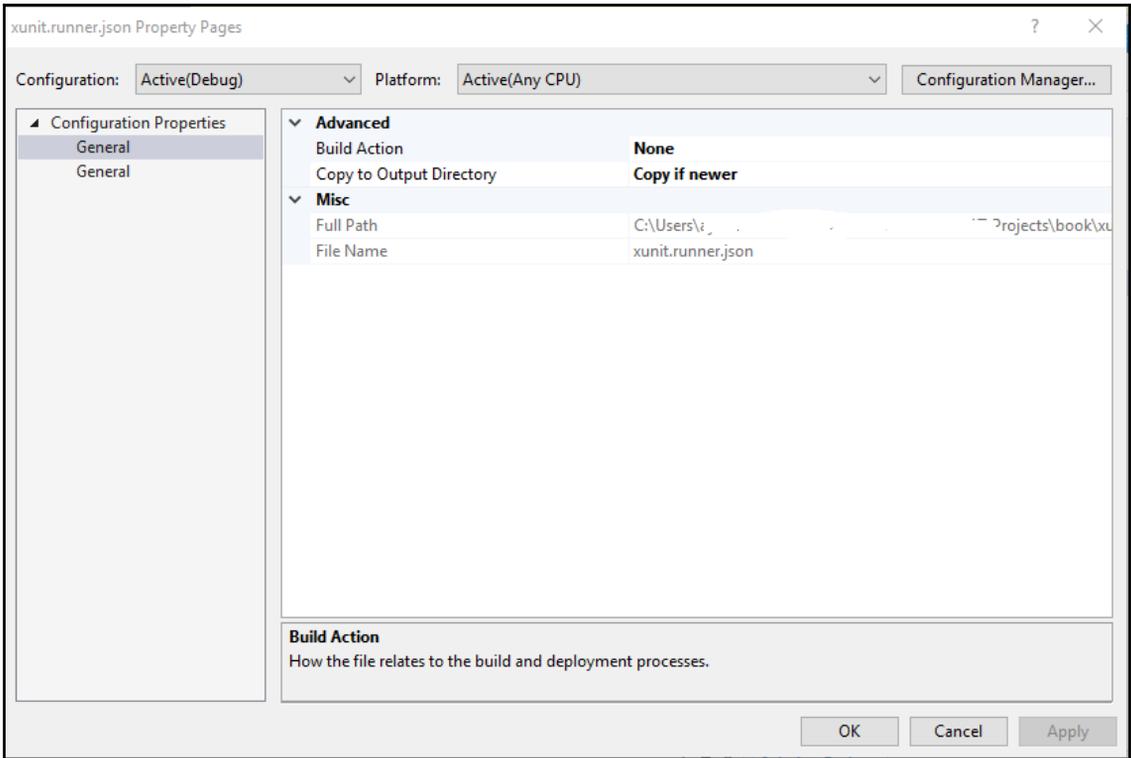
Scheduled For Install: None  
Scheduled For Update: None  
Scheduled For Uninstall: None

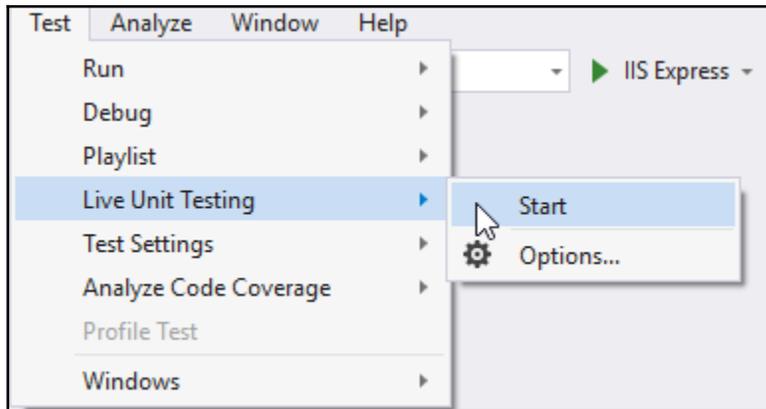
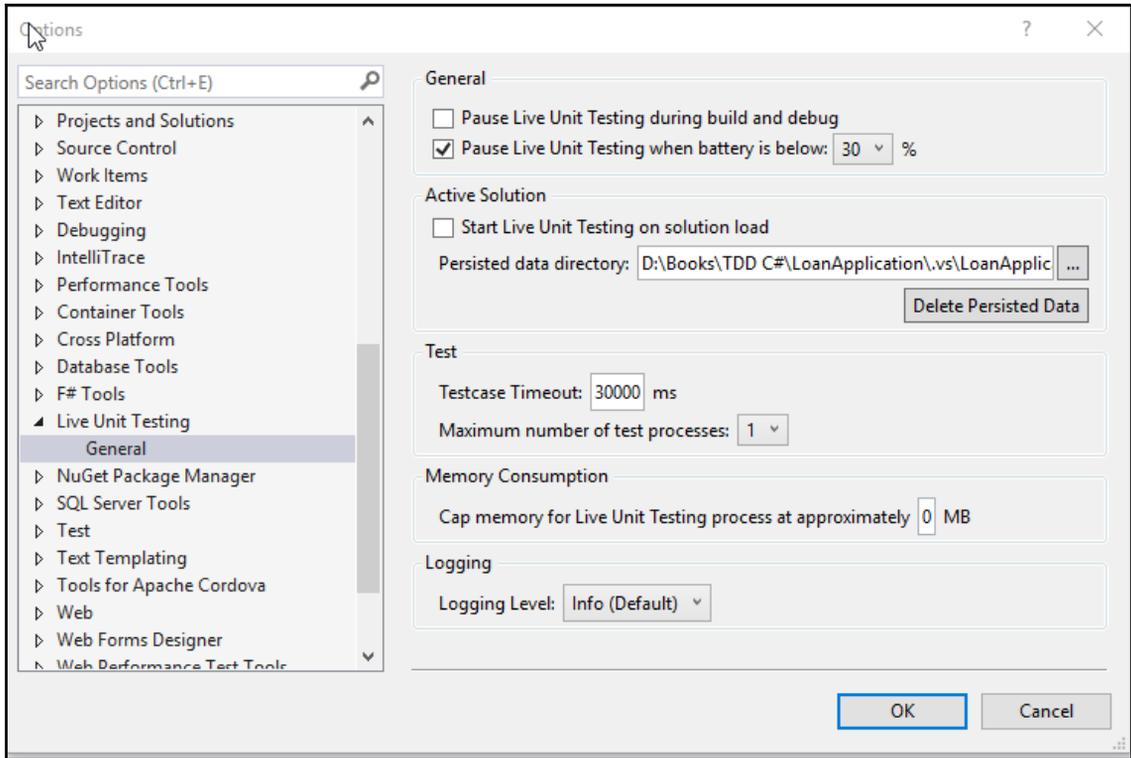
Change your Extensions and Updates settings

Close









LoanApplication - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU LoanApplication IIS Express

NuGet: LoanApplication.Tests.Unit HomeController.cs HomeControllerTests.cs

LoanApplication LoanApplication.Controllers.HomeController loanRepository

```
10 namespace LoanApplication.Controllers
11 {
12     3 references
13     public class HomeController : Controller
14     {
15         private ILoanRepository loanRepository;
16
17         1 reference | 0 exceptions
18         public HomeController(ILoanRepository loanRepository)
19         {
20             this.loanRepository = loanRepository;
21         }
22
23         1 reference | 0 requests | 0 exceptions
24         public IActionResult Index()
25         {
26             return View();
27         }
28
29         1 reference | 0 requests | 0 exceptions
30         public IActionResult About()
31         {
32             ViewData["Message"] = "Your application description page.";
33             return View();
34         }
35
36         0 references | 0 requests | 0 exceptions
37         public IActionResult Contact()
38         {
39             ViewData["Message"] = "Your contact page.";
40             return View();
41         }
42
43         0 references | 0 requests | 0 exceptions
44         public IActionResult Error()
45     }
46 }
```

LoanApplication.Tests.Unit LoanApplication.Tests.Unit.Controller.HomeControllerTest TestIndex()

```
82 }
83 }
84 }
85 0 references
86 public class IndexMethod : HomeControllerTest
87 {
88     [Fact]
89     0 references | 0 exceptions
90     public void TestIndex()
91     {
92         var result = homeController.Index();
93         var viewResult = Assert.IsType<ViewResult>(result);
94         var loanTypes = Assert.IsAssignableFrom<IEnumerable<LoanType>>(viewResult.ViewData["LoanTypes"]);
95         Assert.Equal(3, loanTypes.Count());
96     }
97
98     0 references
99     public class AboutMethod : HomeControllerTest
100 {
101     [Fact]
102     0 references | 0 exceptions
103     public void TestAbout()
104     {
105         var result = homeController.About();
106         var viewResult = Assert.IsType<ViewResult>(result);
107     }
108 }
```

Run  
Debug  
Append To: TestIndex  
Create New Session

The screenshot shows a test runner window titled "TestIndex". The interface includes a toolbar with various icons for running, pausing, and refreshing tests. Below the toolbar, a summary bar shows 2 tests passed (green checkmark), 1 failed (red X), 0 skipped (red circle), 0 rerun (red arrow), and 1 pending (blue question mark). The main area is divided into two panes. The left pane, titled "Type to search", contains a tree view of test results:

- LoanApplication.Tests.Unit (2 tests) Success
  - LoanApplication.Tests.Unit.Controller (2 tests) Success
    - HomeControllerTest+AboutMethod (1 test) Success
    - HomeControllerTest+IndexMethod (1 test)
      - TestIndex

The right pane displays the details for the selected test, "HomeControllerTest+AboutMethod passed", with a large empty area below the header. At the bottom of the right pane, there is an "Output" label.

# Chapter 6: Mocking Dependencies

## Test\_GetOlderCarLoanDefaulters\_ShouldReturnList failed

```
Moq.MockException :
Expected invocation on the mock at least 2 times, but was 1 times: x => x.GetCarLoanDefaulters
  (It.IsInRange<Int32>(1, 12, Range.Inclusive))

Configured setups:
x => x.GetCarLoanDefaulters(It.IsInRange<Int32>(1, 12, Range.Inclusive))

Performed invocations:
ILoanRepository.GetCarLoanDefaulters(12)
  at Moq.Mock.ThrowVerifyException(MethodCall expected, IEnumerable`1 setups, IEnumerable`1
    actualCalls, Expression expression, Times times, Int32 callCount)
  at Moq.Mock.VerifyCalls(Mock targetMock, MethodCall expected, Expression expression, Times
    times)
  at Moq.Mock.Verify[T,TResult](Mock`1 mock, Expression`1 expression, Times times, String
    failMessage)
  at Moq.Mock`1.Verify[TResult](Expression`1 expression, Times times)
```

## Test\_GetOlderCarLoanDefaulters\_ShouldReturnList failed

```
Moq.MockException :
Expected invocation on the mock exactly 2 times, but was 1 times: x => x.GetCarLoanDefaulters
  (It.IsInRange<Int32>(1, 12, Range.Inclusive))

Configured setups:
x => x.GetCarLoanDefaulters(It.IsInRange<Int32>(1, 12, Range.Inclusive))

Performed invocations:
ILoanRepository.GetCarLoanDefaulters(12)
  at Moq.Mock.ThrowVerifyException(MethodCall expected, IEnumerable`1 setups, IEnumerable`1
    actualCalls, Expression expression, Times times, Int32 callCount)
  at Moq.Mock.VerifyCalls(Mock targetMock, MethodCall expected, Expression expression, Times
    times)
  at Moq.Mock.Verify[T,TResult](Mock`1 mock, Expression`1 expression, Times times, String
    failMessage)
  at Moq.Mock`1.Verify[TResult](Expression`1 expression, Times times)
```

```
7 using Moq;
8 using Xunit;
9
10 namespace LoanApplication.Tests.Unit
11 {
12     public class LoanServiceTest
13     {
14         private Mock<ILoanRepository> loanRepository;
15         private LoanService loanService;
16         public void Test()
17         {
18             loanRepository.Setup(x => x.GetCarLoanDefaulters(It.IsAny<int>(), Range.Inclusive))
19                 .Returns(() => new List<Person>
20                 {
21                     new Person { FirstName = "Donald", LastName = "Duke", Age = 30},
22                     new Person { FirstName = "Ayobami", LastName = "Adewole", Age = 20}
23                 });
24             loanRepository.Setup(x => x.GetCarLoanDefaulters(It.IsAny<int>(), Range.Inclusive))
25                 .Returns(() => people);
26             loanService = new LoanService(loanRepository.Object);
27         }
28     }
29 }
```

1 reference

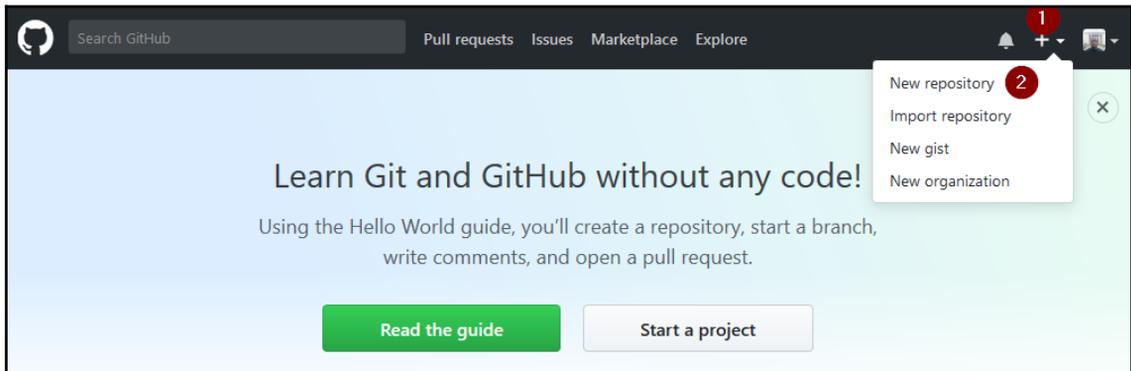
0 reference

'LoanService' is inaccessible due to its protection level

Cannot access internal class 'LoanService' here

Show potential fixes (Alt+Enter or Ctrl+.)

# Chapter 7: Continuous Integration and Project Hosting



# Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 ayobami ▾

Repository name

LoanApplication ✓

Great repository names are short and memorable. Need inspiration? How about **vigilant-invention**.

Description (optional)

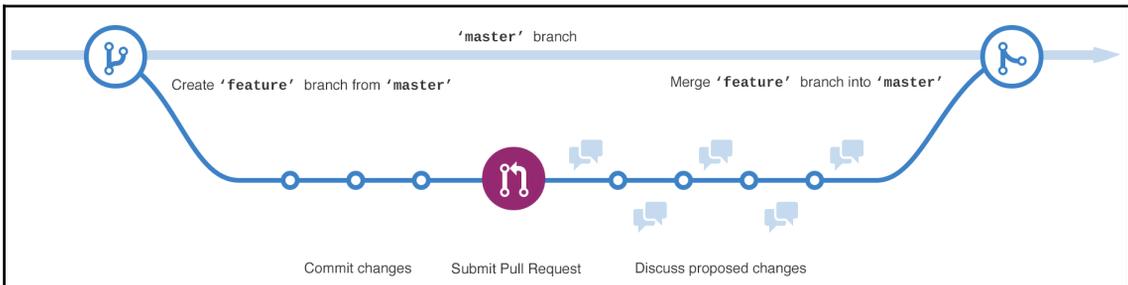
TDD in C# 7 Book

-  **Public**  
Anyone can see this repository. You choose who can commit.
-  **Private**  
You choose who can see and commit to this repository.

- Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

**Create repository**



TDD in C# 7 Book Edit

Add topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Switch branches/tags  
Loan Rate Fix  
Initial commit 2 days ago  
Latest commit a36ae08 2 days ago

Branches Tags  
Create branch: Loan Rate Fix from 'master'

# LoanApplication

TDD in C# 7 Book

Code Issues 0 Pull requests 0 Projects 0 Wiki

## TDD in C# 7 Book

Add topics

1 commit 2 branches

Branch: Loan-Rate-Fix **New pull request**

This branch is even with master.

ayobami Initial commit

README.md Initial commit

README.md

ayobami Updated the readme file Verified 7e558a9

Showing 1 changed file with 3 additions and 0 deletions. Unified Split

3 README.md View

```
@@ -1,2 +1,5 @@
1 # LoanApplication
2 TDD in C# 7 Book
3 +
4 +
5 +Made Changes to this file, in order to make a pull request
```

- Options
- Collaborators
- Branches
- Webhooks 2**
- Integrations & services
- Deploy keys

## Webhooks

Add webhook 3

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).



## Webhooks / Add webhook

**Note:** [GitHub Services](#) are being deprecated. Please contact your integrator for more information on how to migrate or replace a service to [webhooks](#) or [GitHub Apps](#).

We'll send a `POST` request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (`JSON`, `x-www-form-urlencoded`, etc). More information can be found in [our developer documentation](#).

Payload URL \*

Content type

Secret

Which events would you like to trigger this webhook?

- Just the push event.
- Send me everything.
- Let me select individual events.

Active

We will deliver event details when this hook is triggered.

Add webhook

- Options
- Collaborators
- Branches
- Webhooks
- Integrations & services
- Deploy keys

## Webhooks

[Add webhook](#)

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

<a href="http://localhost:54113/API/webhook">http://localhost:54113/API/webhook</a> <i>(create, issues, and push)</i>	<a href="#">Edit</a> <a href="#">Delete</a>
---	---

Last delivery was not successful. Couldn't connect to server.

```

C:\ngrok\ngrok.exe - ngrok http -host-header="localhost:54113" 54113
ngrok by @inconshreveable

Session Status      online
Session Expires    7 hours, 55 minutes
Version             2.2.8
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding          http://d73c1ef5.ngrok.io -> localhost:54113
Forwarding          https://d73c1ef5.ngrok.io -> localhost:54113

Connections
  ttl    opn    rt1    rt5    p50    p90
   3      0    0.01  0.01  120.71 122.43

HTTP Requests
-----
POST /api/webhook      200 OK
POST /api/webhook      200 OK
GET  /favicon.ico      200 OK
GET  /api/webhook      200 OK
  
```

## Recent Deliveries



 217bf560-1f40-11e8-8d7d-20e964da8f46

2018-03-04 01:08:13



Request

Response **200**



Redeliver



Completed in 0 seconds.

### Headers

```
Request URL: http://d73c1ef5.ngrok.io/api/webhook
Request method: POST
content-type: application/json
Expect:
User-Agent: GitHub-Hookshot/7a71d82
X-GitHub-Delivery: 217bf560-1f40-11e8-8d7d-20e964da8f46
X-GitHub-Event: ping
```



## TeamCity First Start

Please review the settings below before proceeding with the first TeamCity start.

TeamCity server stores server configuration settings, project definitions, build results and caches on disk in a **Data Directory**.<sup>?</sup>

Data Directory location on the TeamCity server machine:

If you already worked with TeamCity on this machine you can specify existing Data Directory.

Proceed

TeamCity 2017.2.2 (build 50909)



## Database connection setup

TeamCity server stores builds history and users-related data in an SQL database. ⓘ

Select the database type\*:  ▾

The internal database suits evaluation purposes only and is not intended for production. We strongly recommend using an external database in a production environment. ⓘ

You can start with the internal database and then migrate the data to an external one after successful evaluation.

Proceed

TeamCity 2017.2.2 (build 50909)



Projects | ▾

Changes

Agents 1

Build Queue 0

## Getting started with TeamCity

There are no projects in TeamCity. To start running builds, create projects and build configurations first.

[+ Create project](#)

You may also want to:

- [configure email and Jabber settings](#) to enable notifications,
- [manage licenses](#), and
- [add more users to TeamCity](#).



Projects | ▾

Changes

Agents 1

Build Queue 0

Ayobami Adewole | ▾

Administration / <Root project>

## Create Project

From a repository URL

From GitHub.com

From Bitbucket Cloud  
[Set up connection](#)

From Visual Studio  
Team Services  
[Set up connection](#)

Manually

## Register a new OAuth application

---

### Application name

Something users will recognize and trust

### Homepage URL

The full URL to your application homepage

### Application description

This is displayed to all users of your application

### Authorization callback URL

Your application's callback URL. Read our [OAuth documentation](#) for more information.

---

### Add Connection ✕

Connection type:

Please register TeamCity on GitHub using the following parameters:

Homepage URL: <http://localhost:8060>

Callback URL: <http://localhost:8060/oauth/github/accessToken.html>

And once registered, enter the **client id** and the **client secret** in the form below.

---

Display name: \*   
Provide some name to distinguish this connection from others.

---

Client ID: \*

---

Client secret: \*

Parent project: \* <Root project>

---

Choose a repository: \*

Your repositories (5)

- 7WCM0033 (<https://github.com/ayobami/7WCM0033>)
- ChildrenLearning (<https://github.com/ayobami/ChildrenLearning>)
- Currency-Converter (<https://github.com/ayobami/Currency-Converter>)
- LoanApplication (<https://github.com/ayobami/LoanApplication>)**
- MachineLearning.Net (<https://github.com/ayobami/MachineLearning.Net>)

VCS repository connection has been verified. Please review project and build configuration names and click **Proceed** to create

Project name: \*

Build configuration name: \*

VCS Repository: (Git) <https://github.com/ayobami/LoanApplication>

Proceed

Cancel

## New Build Step

Runner type:    
 Provides .NET CLI toolchain support for .NET projects

Step name:    
 Optional, specify to distinguish this build step from other steps.

Command:

Projects:    
 Specify paths to projects and solutions. **Wildcards** are supported.

Build step settings updated.

## Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with

+ Add build step

✎ Auto-detect build steps

Build Step	Parameters	Description
1. Run Unit Tests	.NET CLI (dotnet) test *	Execute: If all previous steps finished successfully

LoanApplication / Build

## #1 (06 Mar 18 13:54)

[Overview](#) [Changes](#) [Build Log](#) [Parameters](#) [Artifacts](#)

Status:  Running

Progress: **3s passed** 

Thread dump: [View thread dump](#)

# Chapter 8: Creating Continuous Integration Build Processes

```
Administrator: Windows PowerShell
-----
NUnit Console Runner 3.4.0
Copyright (C) 2016 Charlie Poole

Runtime Environment
  OS Version: Microsoft Windows NT 10.0.15063.0
  CLR Version: 4.0.30319.42000

Test Files
  C:\Users\ [redacted] \Downloads\example-master\src\Example.Tests\bin\Release\Example.Tests.dll

Run Settings
  WorkDirectory: C:\Users\ [redacted] \Downloads\example-master
  ImageRuntimeVersion: 4.0.30319
  ImageTargetFrameworkName: .NETFramework,Version=v4.5
  ImageRequiresX86: False
  ImageRequiresDefaultAppDomainAssemblyResolver: False
  NumberOfTestWorkers: 4

Test Run Summary
  Overall result: Passed
  Test Count: 2, Passed: 2, Failed: 0, Inconclusive: 0, Skipped: 0
  Start time: 2018-03-15 13:14:22Z
  End time: 2018-03-15 13:14:22Z
  Duration: 0.113 seconds

-----
Default
-----

Task                Duration
-----
Clean                00:00:00.0181013
Restore-NuGet-Packages 00:00:31.8734676
Build                00:00:03.6141848
Run-Unit-Tests      00:00:02.9735688
-----
Total:                00:00:38.4865294
PS C:\Users\ [redacted] \Downloads\example-master>
```

```
Windows PowerShell
PS C:\Users\ayobami\Downloads\example-master> .\build.ps1
.\build.ps1 : File C:\Users\ayobami\Downloads\example-master\build.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ ~~~~~
+ .\build.ps1
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\ayobami\Downloads\example-master>
```

```

Windows PowerShell
PS C:\Users\ayobami\Downloads\example-master> Get-ExecutionPolicy -List

Scope ExecutionPolicy
-----
MachinePolicy      Undefined
UserPolicy         Undefined
Process            Undefined
CurrentUser        Undefined
LocalMachine       Undefined

PS C:\Users\ayobami\Downloads\example-master>

```

```

Windows PowerShell
PS C:\Users\ayobami\Downloads\example-master> Get-ExecutionPolicy -List

Scope ExecutionPolicy
-----
MachinePolicy      Undefined
UserPolicy         Undefined
Process            Undefined
CurrentUser        Undefined
LocalMachine       Undefined

PS C:\Users\ayobami\Downloads\example-master> Set-ExecutionPolicy RemoteSigned -Scope Process

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): Y
PS C:\Users\ayobami\Downloads\example-master>

```

Configuration option	Description	Default value	Environment variable name	Config file content	Direct argument
Tools path	This option is used to tell Cake which folder contains the tools to be used by Cake when restoring.	./tools	CAKE_PATHS_TOOLS	[Paths]Tools=./tools	cake.exe --paths_tools=./tools
Addins path	This config option is used to specify the Addins folder to be used by cake while restoring Add-ins.	./tools/Addins	CAKE_PATHS_ADDINS	[Paths]Addins=./tools/Addins	cake.exe --paths_addins=./tools/Addins
NuGet download URL	Using this option, you can specify where Cake should download Nuget packages from when you are using the addin and tool preprocessor.	https://packages.nuget.org/api/v2	CAKE_NUGET_SOURCE	[Nuget]Source=http://someurl/nuget/	cake.exe --nuget_source=http://someurl/nuget/
Modules path	This is used for configuring the Modules folder for use by Cake when loading custom modules.	./tools/Modules	CAKE_PATHS_MODULES	[Paths]Modules=./tools/Modules	cake.exe --paths_modules=./tools/Modules
Skip verification	This option is used to avoid runtime errors that can occur during script execution. When exceptions occur during script execution, Cake will abort the execution of the script.	false	CAKE_SETTINGS_SKIPVERIFICATION	[Settings]SkipVerification=true	cake.exe --settings_skipverification=true
Add-in NuGet dependencies	A configuration option that became available since Cake Version 0.22.0 to install and reference NuGet package dependencies.	false	CAKE_NUGET_LOADDEPENDENCIES	[Nuget]LoadDependencies=true	cake.exe --nuget_loaddependencies=true

```
Preparing to run build script...
Running build script...
=====
Clean
=====
Restore-NuGet-Packages
=====
Restoring NuGet Packages

Welcome to .NET Core!
-----
Learn more about .NET Core @ https://aka.ms/dotnet-docs. Use dotnet --help to see available commands or go to https://aka.ms/dotnet-cli-docs.

Telemetry
-----
The .NET Core tools collect usage data in order to improve your experience. The data is anonymous and does not include command-line arguments. The data is collected by Microsoft and shared with the community.
You can opt out of telemetry by setting a DOTNET_CLI_TELEMETRY_OPTOUT environment variable to 1 using your favorite shell.
```

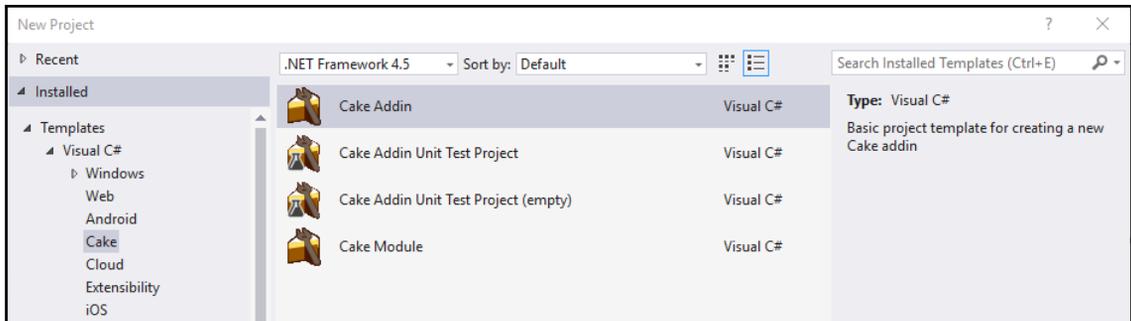
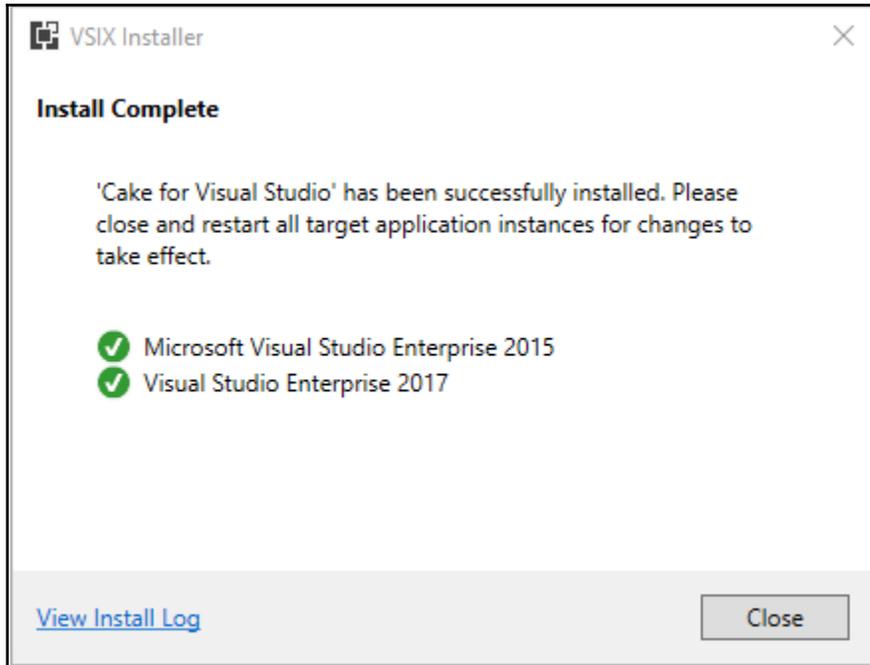
```
0 Error(s)

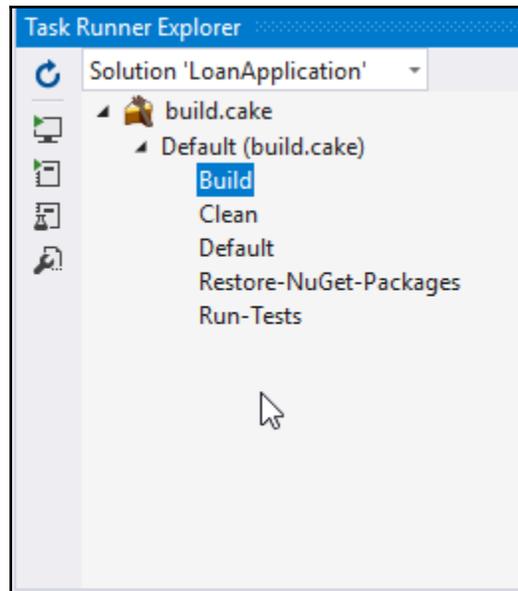
Time Elapsed 00:01:07.79

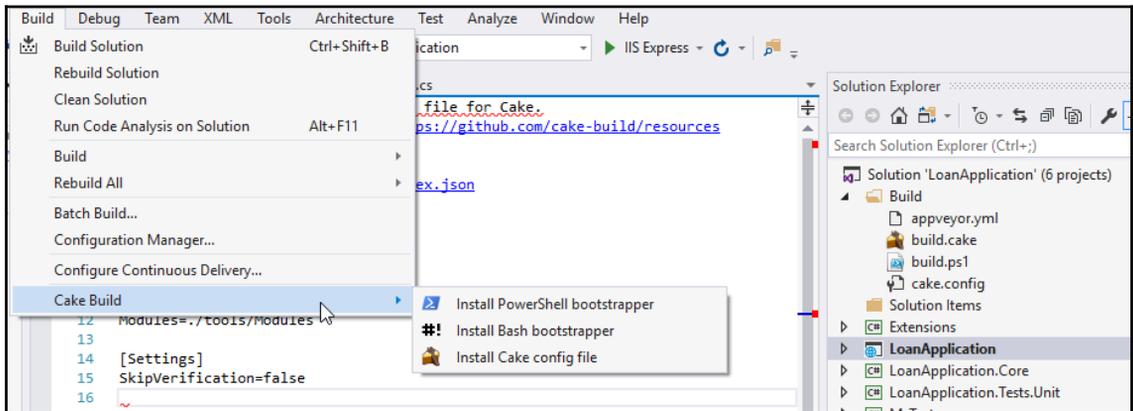
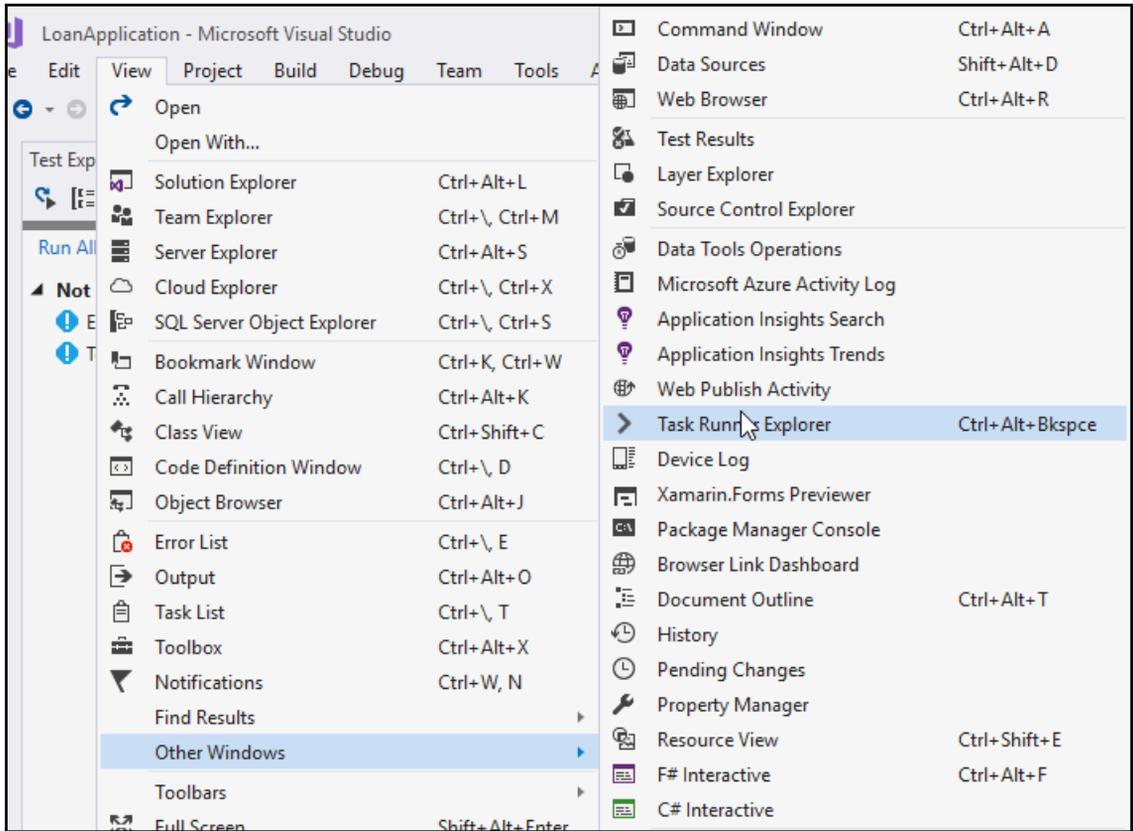
=====
Run-Tests
=====

Default
=====

Task                Duration
-----
Clean                00:00:00.0248073
Restore-NuGet-Packages 00:00:29.1532166
Build                00:01:17.0493209
Run-Tests            00:00:00.0128710
-----
Total:                00:01:46.2433470
```







Task Runner Explorer

Solution 'LoanApplication' ▾

- build.cake
  - Default (build.cake)
    - Build
    - Clean**
    - Default
    - Restore-NuGet-Packages
    - Run-Tests

Bindings Clean x

```
C:\Users\... \Documents\Visual Studio 2017\Projects\book\LoanApplication> C:\Users\... \Documents\Visual Studio 2017\Projects\book\LoanApplication\tools\Cake\Cake.exe build.cake -Target="Clean"
=====
Clean
=====
Task                Duration
-----
Clean                00:00:00.0392250
Total:               00:00:00.0392250
Process terminated with code 0.
```

build.cake LoanRepositoryTest.cs

```
1 //Arguments
2 var target = Argument("target", "Default");
3 var configuration = Argument("configuration", "Release");
4 var solution = "./LoanApplication.sln";
5
6 // Define directories.
7 var buildDir = Directory("./LoanApplication.Core/bin") + Directory(configuration);
8
9 //Tasks
10 Task("Clean")
11     .Does(() =>
12     {
13         CleanDirectory(buildDir);
14     });
15
16 Task("Restore-NuGet-Packages")
17     .IsDependentOn("Clean")
18     .Does(() =>
19     {
20         Information("Restoring NuGet Packages");
21         DotNetCoreRestore();
22     });
23
24 Task("Build")
25     .IsDependentOn("Restore-NuGet-Packages")
26     .Does(() =>
27     {
28         Information("Build Solution");
29         DotNetCoreBuild(solution,
30             new DotNetCoreBuildSettings()

```



### New Build Step

Runner type:  PowerShell runner

Step name:  Optional, specify to distinguish this build step from other steps.

Script:

Script file: \*  Path to the PowerShell script, relative to the checkout directory

Script execution mode:  Specify PowerShell script execution mode. By default, PowerShell may not allow -ExecutionPolicy Bypass argument.

Script arguments: [Expand](#)

[Show advanced options](#)

- appveyor.yml
- build.cake
- build.ps1
- cake.config
- LoanApplication.sln
- README.md

### Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. [?](#)

Build Step	Parameters	Description	
1. Cake Build	PowerShell PowerShell <Any Bitness> File: build.ps1	Execute: If all previous steps finished successfully	<a href="#">Edit</a> <input type="button" value="⋮"/>

## Host my projects at:

ayobami

.visualstudio.com

Manage code using:

-  Git
-  Team Foundation Version Control

We will host your projects in **Central US** location.

You can share work with other |  users.

 [Change details](#)

[Continue](#)

To keep our lawyers happy:

By continuing, you agree to the [Terms of Service](#), [Privacy Statement](#), and [Code of Conduct](#).



## Create new project

Projects contain your source code, work items, automated builds and more.

Project name \*

LoanApplication



Description

LoanApplication



Version control

Git



Work item process

Agile



Create

Cancel

# Get started with your new project!

## ^ Clone to your computer

**HTTPS** SSH   OR

[Generate Git credentials](#)

 [Clone in Visual Studio](#) | 

 Having problems authenticating in Git? Be sure to get the latest version of [Git for Windows](#) or our plugins for [IntelliJ](#), [Eclipse](#), [Android Studio](#) or [Windows command line](#).

## ∨ or push an existing repository from command line

## ^ or import a repository

[Import](#)

## ∨ or initialize with a README or gitignore

## Import a Git repository



Source type

Git



Clone URL \*

https://github.com/ayobami/LoanApplication.git



Requires authorization

Username

ayobami

Username

Password / PAT \*

●●●●●●●●

Import

Close

# Import successful!



Congratulations! Your <https://github.com/ayobami/LoanApplication.git> repository has been successfully imported.

If you are not automatically redirected to your repository page [Click here to navigate to code view.](#)

The screenshot shows the GitHub interface for the repository 'LoanApplication'. The top navigation bar includes 'Files', 'Commits', 'Pushes', 'Branches', 'Tags', and 'Pull Requests'. The breadcrumb path is 'LoanApplication / Type to find a file or folder...'. A 'Set up build' button is visible in the top right.

The file explorer on the left shows the following structure:

- LoanApplication
  - Extensions
  - LoanApplication
  - LoanApplication.Core
  - LoanApplication.Tests.Unit
  - packages
  - tools
  - appveyor.yml
  - build.cake
  - build.ps1
  - cake.config
  - LoanApplication.sln
  - README.md

The main content area displays a table of files with columns for Name, Last change, and Commits. A tooltip is shown over the commit hash 'b3b965a3' in the 'Commits' column for the file 'LoanApplication.Core'.

Name ↑	Last change	Commits
Extensions	3/26/2018	0b8627f8 Extension project Ayobami Adewole
LoanApplication	3/26/2018	90198950 LoanApplicaion upload Ayobami Adewole
LoanApplication.Core	3/26/2018	b3b965a3 Test project upload Ayobami Adewole
LoanApplication.Tests.Unit	3/26/2018	it project upload Ayobami Adewole
packages	3/26/2018	9912b2fd Cake upload Ayobami Adewole
tools	3/26/2018	9912b2fd Cake upload Ayobami Adewole
appveyor.yml	3/25/2018	e49c3534 committed Cake files Ayobami Adewole
build.cake	3/25/2018	e49c3534 committed Cake files Ayobami Adewole
build.ps1	3/25/2018	e49c3534 committed Cake files Ayobami Adewole
cake.config	3/25/2018	e49c3534 committed Cake files Ayobami Adewole
LoanApplication.sln	3/25/2018	e49c3534 committed Cake files Ayobami Adewole



# Cake

**Cake Build** | 📄 1,268 installs | ★★★★★ (8) | Free

Tools for building with Cake.

Get it free



Cake



Account



Done

Select a Visual Studio Team Services account

ayobami

Install

For Team Foundation Server

Download

Visual Studio | Marketplace



Cake

Account Done

You are all set!

[Proceed to account](#) [Go to Marketplace](#)

Additionally:  
[Learn more about Cake](#)

Select your repository

Tell us where your sources are.  
You can customize how to get these sources from the repository later.

Select a source

- VSTS Git
- GitHub
- GitHub Enterprise
- Subversion
- Bitbucket® Cloud
- External Git

Team project

Repository

Default branch for manual and scheduled builds

[Continue](#)



Choose a template

Choose a template that builds your kind of app.  
Don't worry if it's not an exact match; you can add and customize the tasks later.

### Select a template

Or start with an [Empty process](#)

Search

---

**Featured**

-  **.NET Desktop**  
 Build and run tests for .NET Desktop or Windows Classic Desktop solutions. This template requires that Visual Studio be installed on the build agent.
-  **ASP.NET**  
 Build ASP.NET web applications
-  **ASP.NET Core**  
 Build ASP.NET Core web applications targeting .NET Core Apply
-  **ASP.NET Core (.NET Framework)**  
 Build ASP.NET Core web applications targeting the full .NET Framework
-  **Azure Web App**  
 Build, package, test and deploy your Azure Web App.
-  **Universal Windows Platform**



Choose a template

Choose a template that builds your kind of app.  
Don't worry if it's not an exact match; you can add and customize the tasks later.

### Select a template

Or start with an [Empty process](#)

Empty ✕

---

**Others**

 **Empty**  
 Start with a definition that has no steps.
 Apply

Tasks Variables Triggers Options Retention History

Process  
Build process

Get sources  
LoanApplication master

Phase 1  
Run on agent

### LoanApplication-ASP.NET Core-CI

Save & queue Discard Summary Queue

Add tasks Refresh

Cake ✕

 **Cake**  
 Build with Cake
 Add

LoanApplication-CI

Tasks Variables Triggers Options Retention History Save & queue Discard Summary Queue

Process  
Build process

Get sources  
LoanApplication master

Phase 1  
Run on agent

Cake  
Cake

Cake  
Link settings Remove

Version 0,\*

Display name \*  
Cake

Cake Script \*  
build.cake

Target \*  
Default

Verbosity \*  
Verbose

Cake Arguments

Advanced

Control Options

Output Variables

## Save build definition and queue

Save comment

Configured Cake Script

Agent queue

Hosted VS2017

Branch

master

Commit

Configured Cake Script

Variables

Demands

BuildConfiguration	release
BuildPlatform	any cpu
system.debug	false

+ Add

Save & queue

Cancel

Build #20180503.1 has been queued.

Tasks Variables Triggers Options Retention History Save & queue Discard Summary Queue

**Process**  
Build process

**Get sources**  
LoanApplication master

**Phase 1**  
Run on agent

**Cake**  
Cake

**Publish Artifact**  
Publish Build Artifacts

Name \*  
LoanApplication-ASPNET Core-CI

Agent queue \* | Manage  
Hosted VS2017

Parameters  
This definition doesn't have any process parameters yet. You can create and use them to promote the most important settings in your process.  
[Learn more](#)

Builds Releases Library Task Groups Deployment Groups

Build 20180503.1

Phase 1

Job

- Initialize Agent
- Initialize Job
- Get Sources
- Cake
- Publish Artifact
- Post Job Cleanup

LoanApplication-ASPNET Core-CI / Build 20180503.1 / Phase 1 / Job

Edit build definition Cancel Queue new build... Download all logs as zip Release

**Build Started**

Job  
Running for 10 seconds (Hosted Agent)

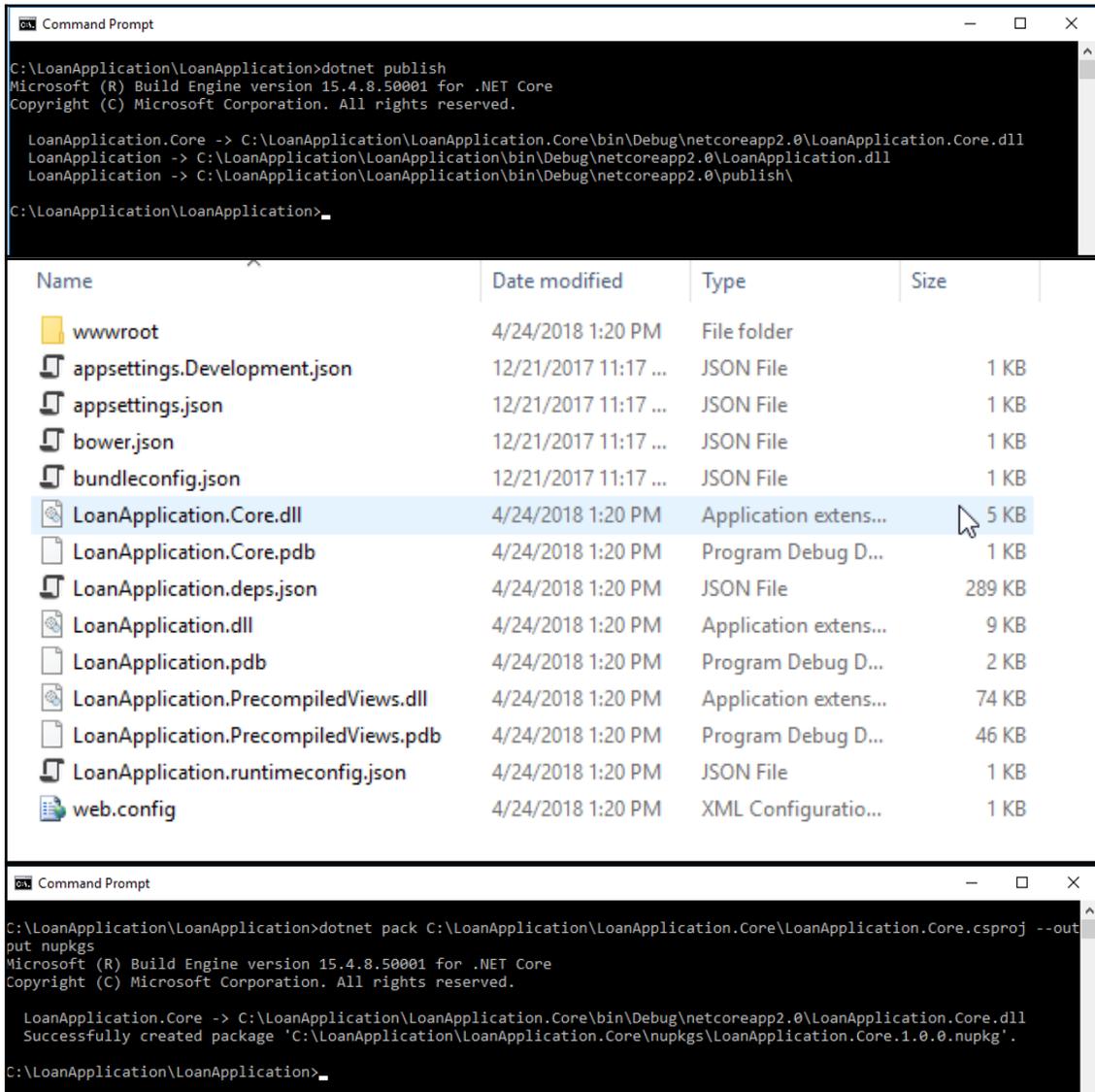
Console Timeline Code coverage\* Tests

```

From https://ayoband.visualstudio.com/_git/LoanApplication
* [new branch] Loan-Rate-Fix -> origin/Loan-Rate-Fix
* [new branch] master -> origin/master
git checkout --progress --force 9912b2fddef2bffc4d0dfc1e8d53ee7c8549dd4
Note: checking out '9912b2fddef2bffc4d0dfc1e8d53ee7c8549dd4'.
You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.
If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:
  git checkout -b <new-branch-name>
HEAD is now at 9912b2f... Cake upload
.....
Finishing: Get Sources
.....
Starting: Cake
.....
=====
Task      : Cake
Description : Build with Cake
Version    : 0.3.0
Author    : Patrik Svensson
Help      : [More Information about Cake](http://cakebuild.net)
=====

```

# Chapter 9: Testing and Packaging the Application



```
C:\LoanApplication\LoanApplication>dotnet publish
Microsoft (R) Build Engine version 15.4.8.50001 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

LoanApplication.Core -> C:\LoanApplication\LoanApplication.Core\bin\Debug\netcoreapp2.0\LoanApplication.Core.dll
LoanApplication -> C:\LoanApplication\LoanApplication\bin\Debug\netcoreapp2.0\LoanApplication.dll
LoanApplication -> C:\LoanApplication\LoanApplication\bin\Debug\netcoreapp2.0\publish\

C:\LoanApplication\LoanApplication>
```

Name	Date modified	Type	Size
wwwroot	4/24/2018 1:20 PM	File folder	
appsettings.Development.json	12/21/2017 11:17 ...	JSON File	1 KB
appsettings.json	12/21/2017 11:17 ...	JSON File	1 KB
bower.json	12/21/2017 11:17 ...	JSON File	1 KB
bundleconfig.json	12/21/2017 11:17 ...	JSON File	1 KB
LoanApplication.Core.dll	4/24/2018 1:20 PM	Application extens...	5 KB
LoanApplication.Core.pdb	4/24/2018 1:20 PM	Program Debug D...	1 KB
LoanApplication.deps.json	4/24/2018 1:20 PM	JSON File	289 KB
LoanApplication.dll	4/24/2018 1:20 PM	Application extens...	9 KB
LoanApplication.pdb	4/24/2018 1:20 PM	Program Debug D...	2 KB
LoanApplication.PrecompiledViews.dll	4/24/2018 1:20 PM	Application extens...	74 KB
LoanApplication.PrecompiledViews.pdb	4/24/2018 1:20 PM	Program Debug D...	46 KB
LoanApplication.runtimeconfig.json	4/24/2018 1:20 PM	JSON File	1 KB
web.config	4/24/2018 1:20 PM	XML Configuratio...	1 KB

```
C:\LoanApplication\LoanApplication>dotnet pack C:\LoanApplication\LoanApplication.Core\LoanApplication.Core.csproj --out
put nupkgs
Microsoft (R) Build Engine version 15.4.8.50001 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

LoanApplication.Core -> C:\LoanApplication\LoanApplication.Core\bin\Debug\netcoreapp2.0\LoanApplication.Core.dll
Successfully created package 'C:\LoanApplication\LoanApplication.Core\nupkgs\LoanApplication.Core.1.0.0.nupkg'.

C:\LoanApplication\LoanApplication>
```

```
Command Prompt
C:\LoanApplication\LoanApplication.Core\nupkgs> dotnet nuget push LoanApplication.Core.1.0.0.nupkg -k oy2ako4kjecbdumfc3
tag5rybiril7wubavkpyhm6ho5li -s https://www.nuget.org/
info : Pushing LoanApplication.Core.1.0.0.nupkg to 'https://www.nuget.org/'...
info : PUT https://www.nuget.org/api/v2/package/
info : Created https://www.nuget.org/api/v2/package/ 4882ms
info : Your package was pushed.

C:\LoanApplication\LoanApplication.Core\nupkgs>
```

🔍 > Packages

✓ Published Packages 1 package / 0 downloads

Package ID	Owners	Downloads	Latest Version	
 LoanApplication.Core	ayobami	0	1.0.0	  