

Activate Azure with Application Insights

LAB

October 2, 2020

Version 1.01

Revision and Signoff Sheet

Change Record

Date	Author	Version	Change Reference
29 Sep 2020	Mubi Ali	1.0	First Version

Reviewers

Date	Reviewer	Version	Change Reference
2 Oct 2020	Prakash Patel	1.01	First Release

Contents

1	Introduction	4
1.1	Objectives.....	4
1.2	Prerequisites	4
1.3	Estimated Time to Complete	4
1.4	Scenario.....	4
2	Exercise 1 – Creating Application Insight.....	5
3	Exercise 2 – Add App Insights to .Net Core Project	7
4	Exercise 3 – Getting Telemetry.....	14
4.1	Telemetry in Visual Studio	14
4.2	Telemetry in Azure Portal.....	14
5	Exercise 4 – Adding Custom Events	16

1 Introduction

In this lab you will integrate Azure Application Insights into application.

1.1 Objectives

After completing this lab, you will be able to:

- Deploy Azure Application Insights into new ASP.NET Core application

1.2 Prerequisites

- Microsoft Azure Subscription
- Internet Connectivity
- Visual Studio 2019
- Access to Azure Portal
- Hands-on experience developing ASP.NET Core applications
- Basic understanding of Azure and its application development model

1.3 Estimated Time to Complete

60 Minutes

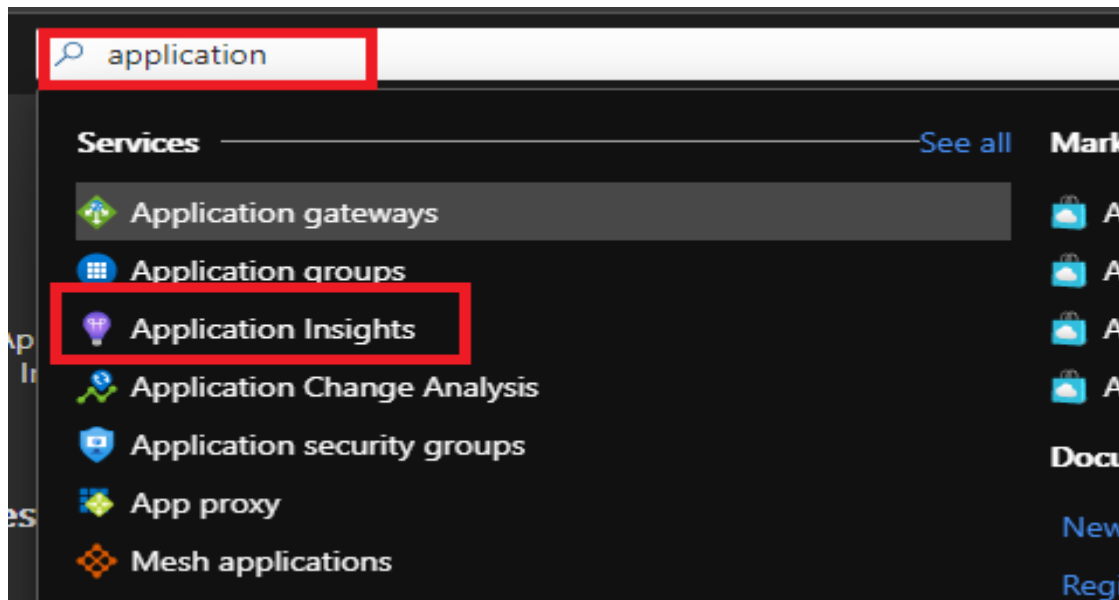
1.4 Scenario

To demonstrate the capabilities of the Azure Application Insights, You will be deploying a number of features that will touch on Azure application monitoring capabilities. This will include creating a new ASP.NET Core application and integrate it with Azure Application Insights.

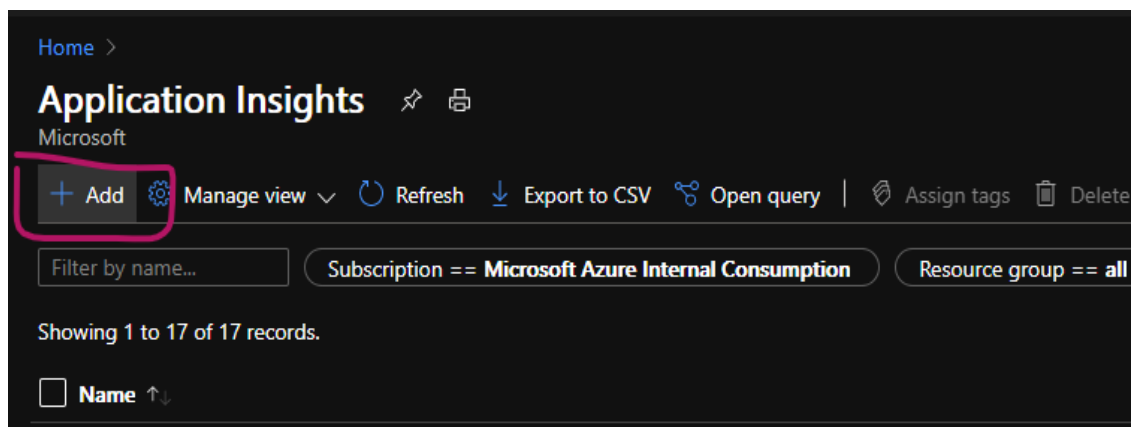
2 Exercise 1 – Creating Application Insight

In this exercise we will create App Insight instance.

1. Go to Azure Portal <https://portal.azure.com>
2. Search for Application Insights on the main search bar in Azure Portal



3. Click + Add




4. Provide the details
 - Select subscription you want to use for this resource
 - Provide name of resource group, you can select existing one or create new by clicking at create new
 - Provide name of Application Insights Resource e.g ApplInsights-Lab
 - Select Regions, select nearest e.g. North Europe (Dublin)

- Select Classic, unless you want to retain data for longer period in that case select workspace-based

Home > Application Insights >

Application Insights



Monitor web app performance and usage

Create an Application Insights resource to monitor your live web application. With Application Insights, you have full observability into your application across all components and dependencies of your complex distributed architecture. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js and Java EE, hosted on-premises, hybrid, or any public cloud. [Learn More](#)

PROJECT DETAILS

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Microsoft Azure Internal Consumption

Resource Group * ⓘ

(New) appinsights-lab

Create new

INSTANCE DETAILS

Name * ⓘ

Region * ⓘ

(US) West US 2

Resource Mode * ⓘ

ClassicWorkspace-based

WORKSPACE DETAILS

Subscription * ⓘ

Microsoft Azure Internal Consumption

Log Analytics Workspace * ⓘ


DefaultWorkspace-3184ff26-27cb-4e97-8a37-4cdefff251e7-WEU [we...]

Review + create

« Previous

Next : Tags >

- Instrument key will be displayed in the Overview tab, this is key which is needed for SDK integration.



The screenshot shows the Azure portal interface for a resource group named 'mynew-rg'. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Investigate, and Application map. The main content area is titled 'Essentials' and displays the following information:

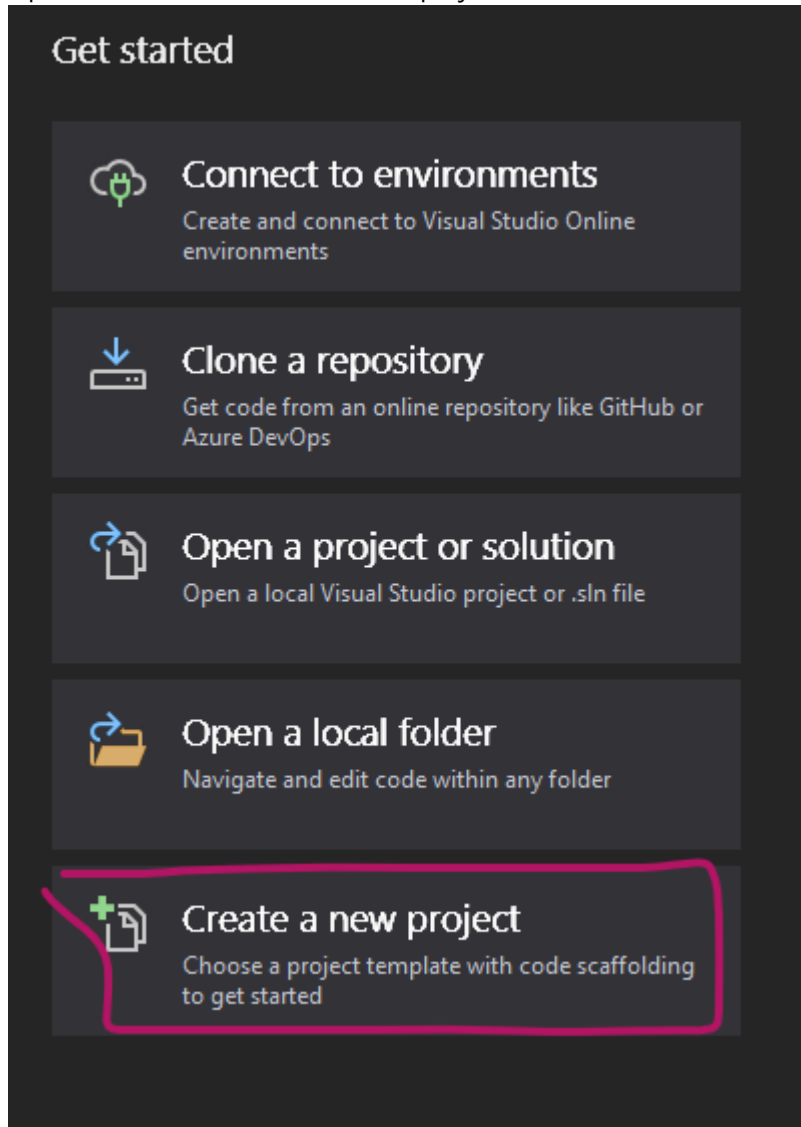
- Resource group (change): mynew-rg
- Location: North Europe
- Subscription (change): Microsoft Azure Internal Consumption
- Subscription ID: 3184ff26-27cb-4e97-8a37-4cdefff251e7
- Tags (change): Click here to add tags

Below this information, there is a section for 'Show data for last:' with a time range selector. The '1 hour' option is selected. The 'Instrumentation Key' is highlighted with a red box, showing a value starting with 'd-'. The 'Connection String' is also visible, starting with 'InstrumentationKey='.

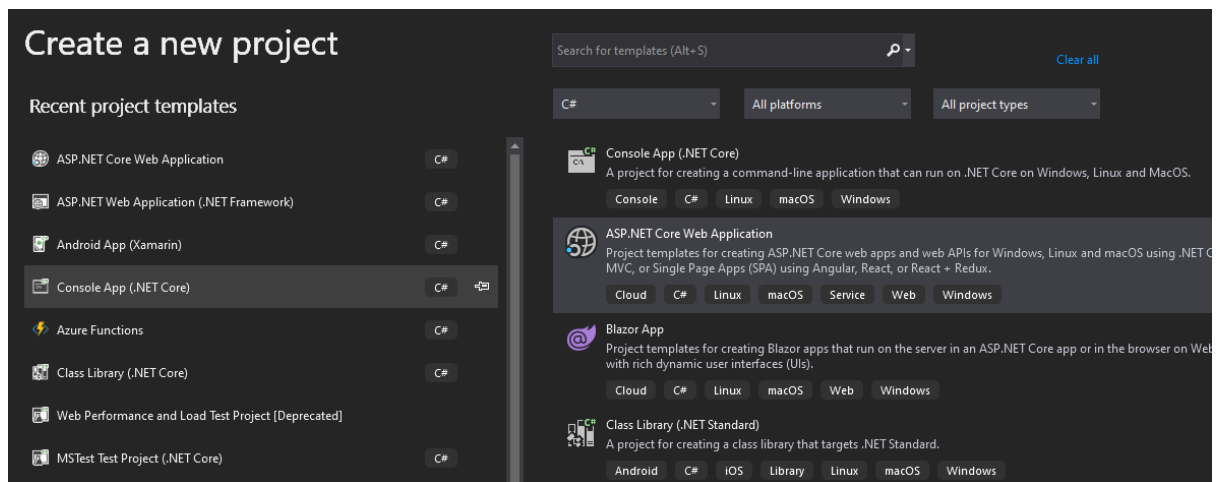
3 Exercise 2 – Add App Insights to .Net Core Project

In this exercise we will create a new ASP.NET Core Project and link it with App Insight created earlier.

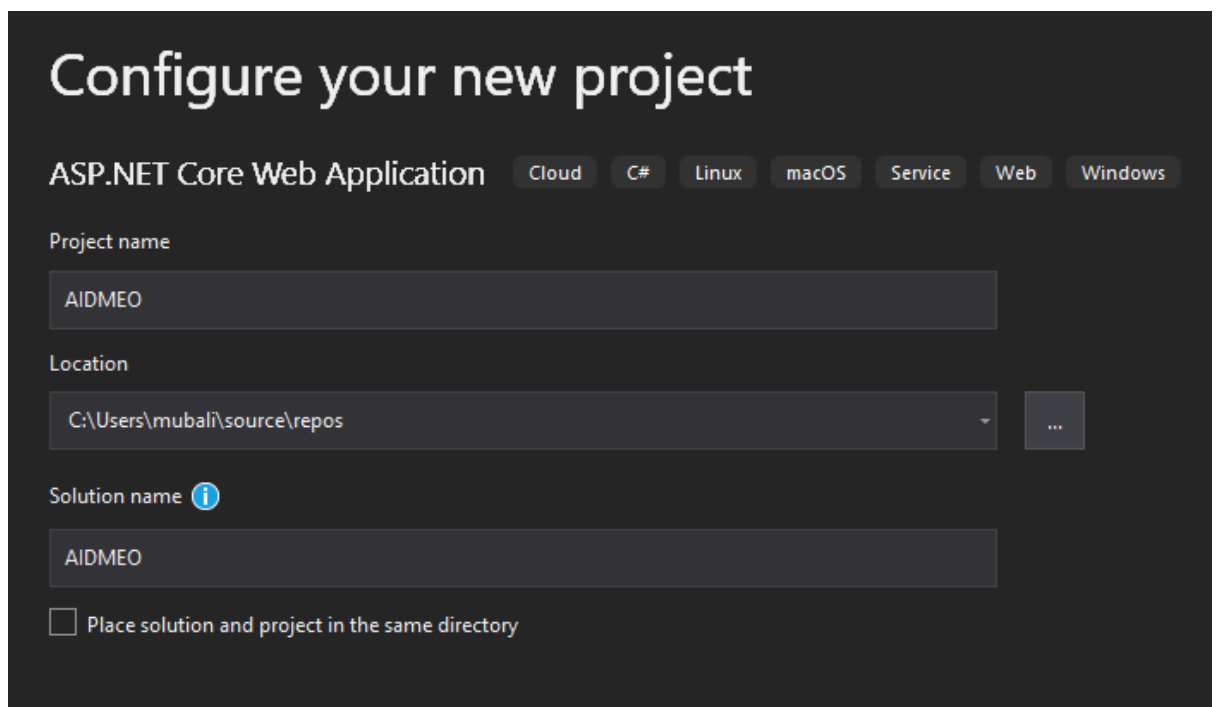
1. Open visual studio and create new project



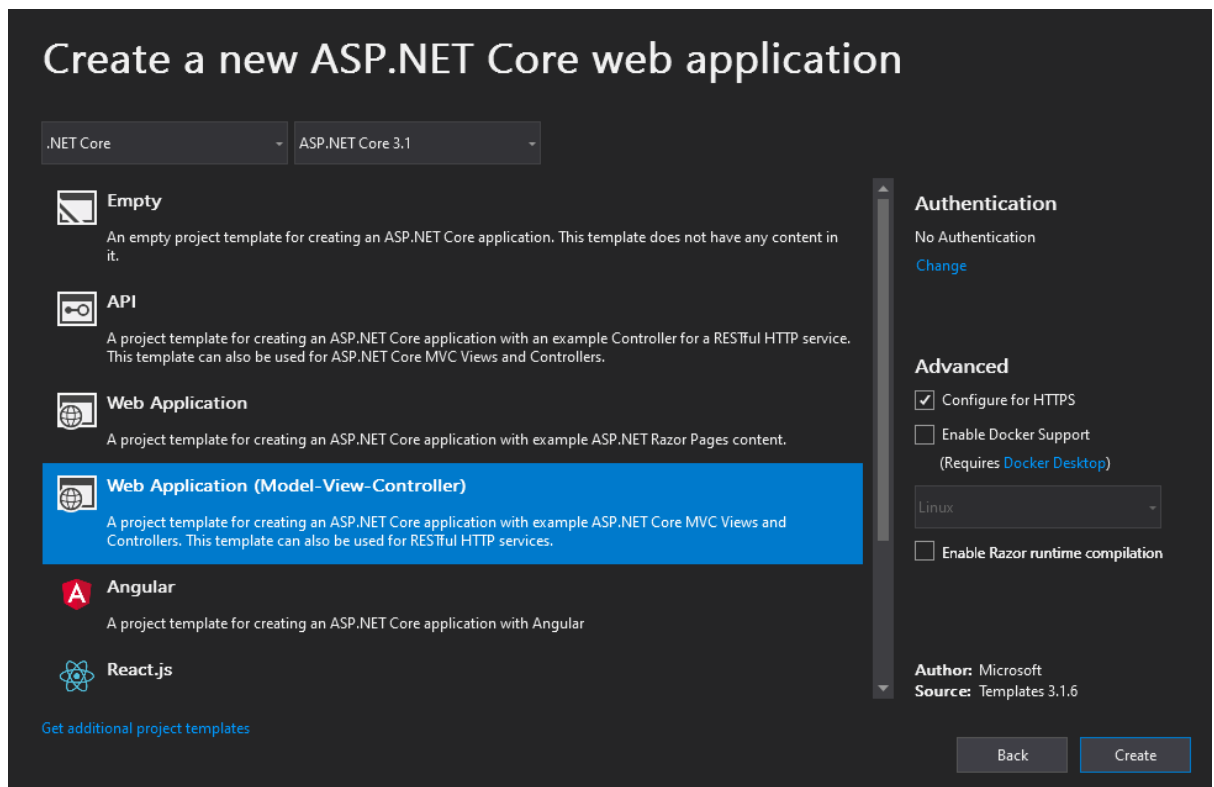
2. Select ASP.NET Core Web Application template



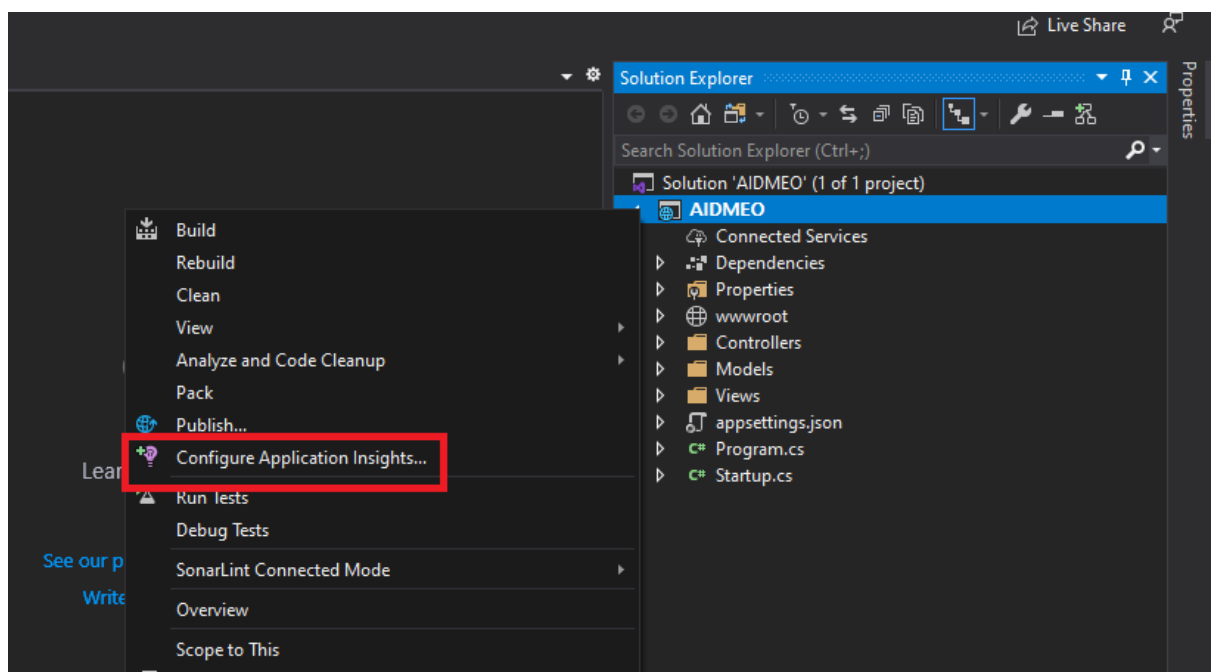
3. Configure new project, provide name and location



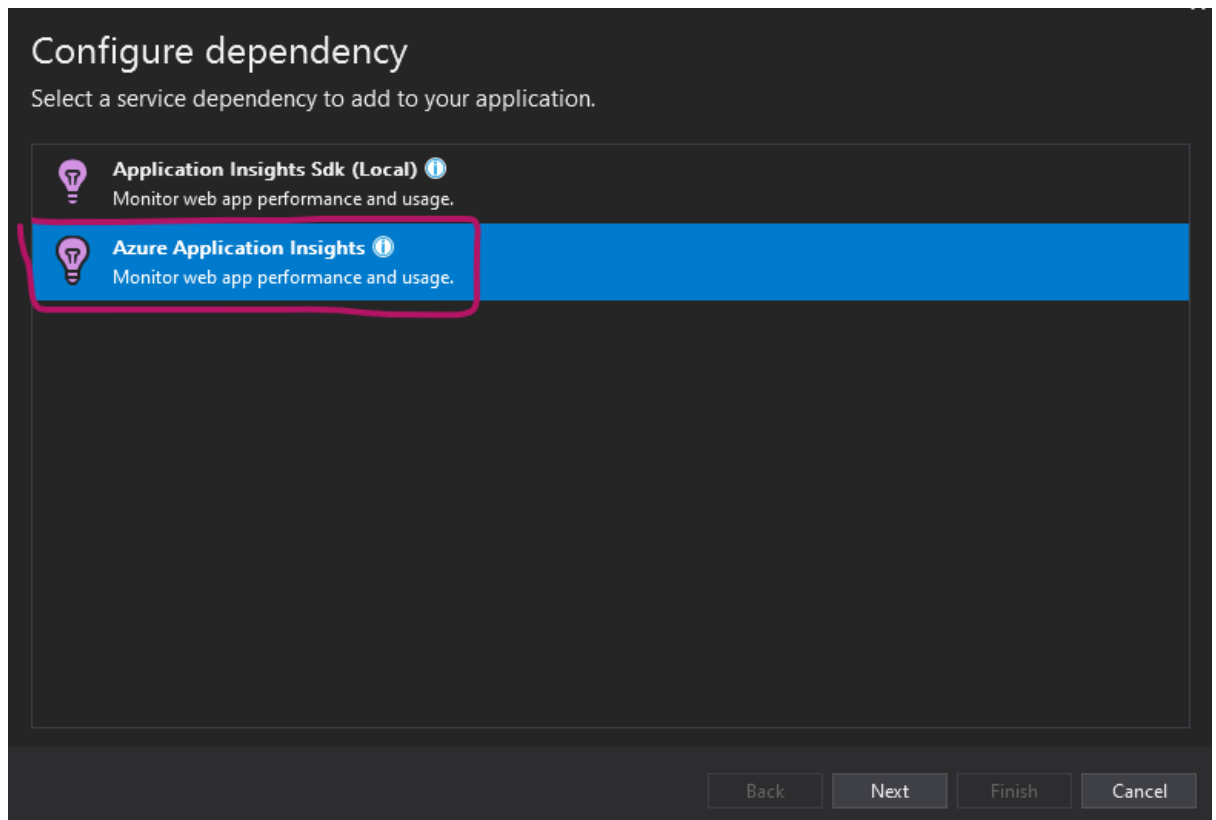
4. Select Web Application (Model-View-Controller)



5. After few seconds your project will be created, right click on your project file and click Configure Application Insights....



6. Select Azure Application Insights. SDK option just add code and NuGet packages and let's you debug locally, where as second option will transmit data to Azure App Insights.



7. Select the App Insight that you have created for Lab, if you have not created one, create a new App Insights...

Configure Azure Application Insights

Select a service dependency to add to your application.

Subscription

Microsoft [redacted]@microsoft.com

⚠ Reenter your credentials

Name	Resource group	Location
[redacted]	ApplicationInsights	North Europe
[redacted]	ApplicationInsights	North Europe
[redacted]	ApplicationInsights	North Europe
[redacted]	ApplicationInsights	North Europe
myappinsight	mynew-rg	North Europe
[redacted]	[redacted]	North Europe
[redacted]	[redacted]	North Europe
[redacted]	[redacted]	North Europe
[redacted]	[redacted]	North Europe
[redacted]	[redacted]	West Europe
[redacted]	[redacted]	West Europe

+ Create a new Application Insights component

Refresh

Back

Next

Finish

Cancel

8. Once you link App Insights, click Next

×

Configure Azure Application Insights


Provide connection string name and specify how to save it

Instrumentation key name

APPINSIGHTS_INSTRUMENTATIONKEY

Connection string value

👁️📄

 Tip: avoid pasting application secrets directly into your code.

Save connection string value in [Learn more](#)

☐ Local user secrets file

☒ None

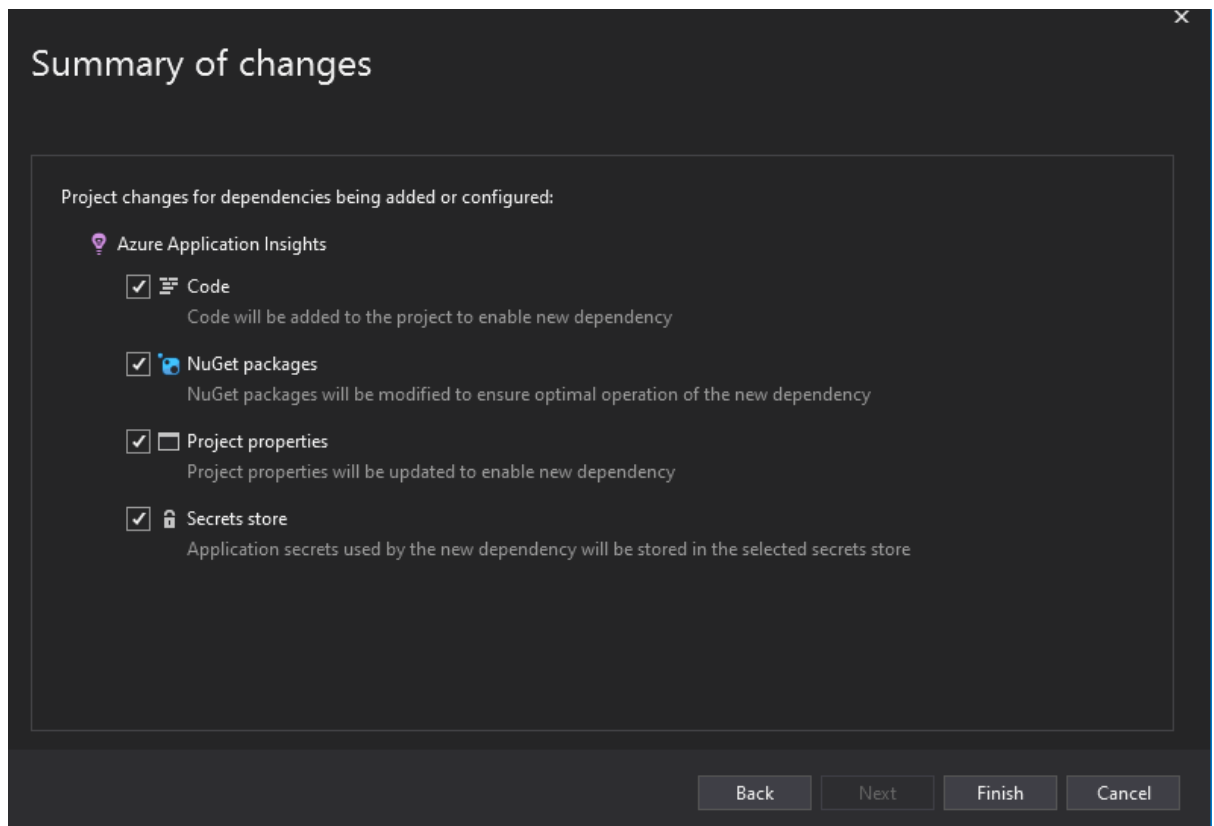
Back

Next

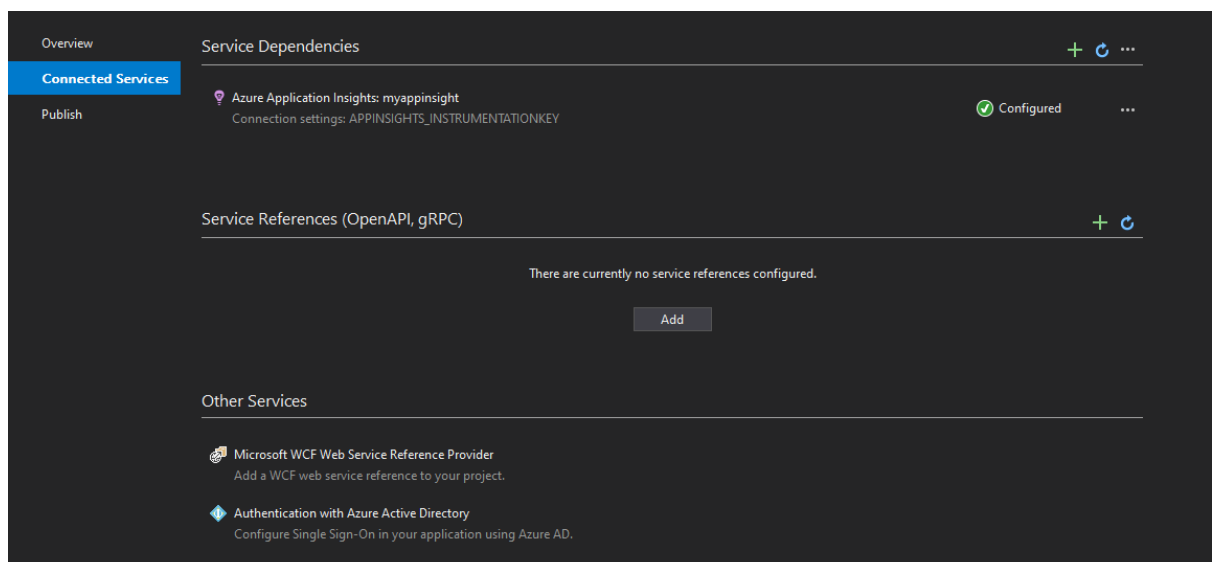
Finish

Cancel

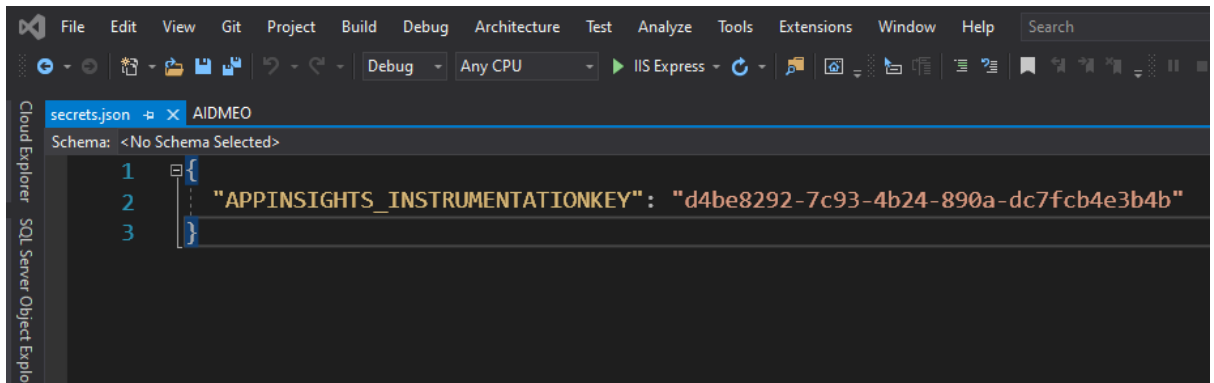
- Click Finish and App Insights will be added to your project.



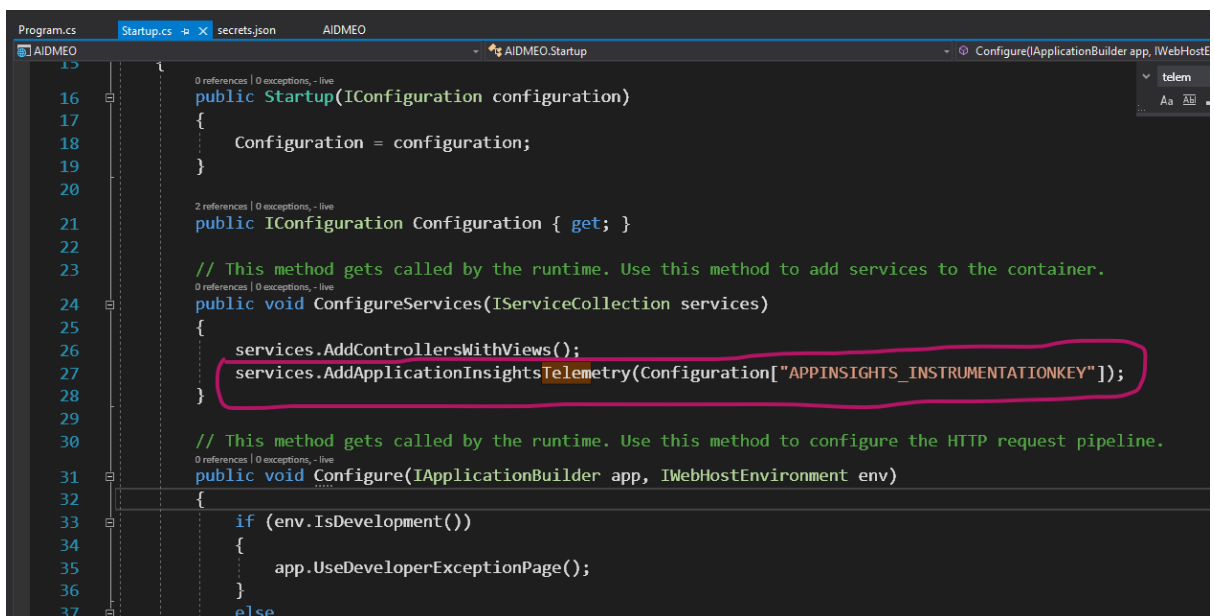
- Confirm that application insights is added to the project and configured properly.



- Right click on your project and click "Manage User Secrets", you should be able to see App Insights Key.



11. Application Insight has been added to your project

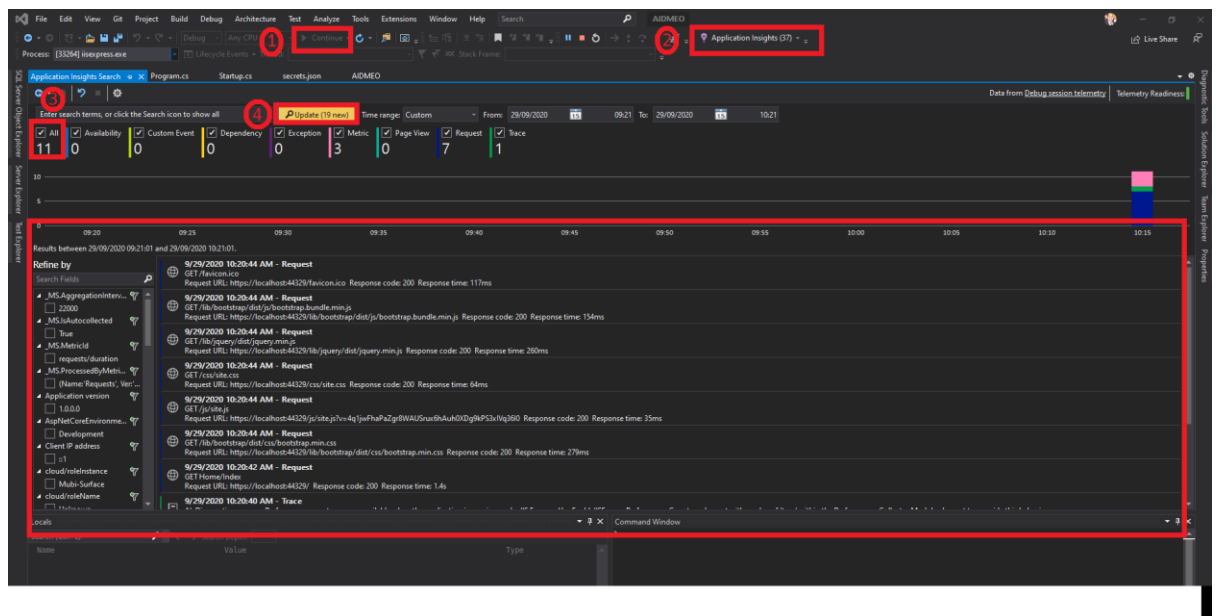


4 Exercise 3 – Getting Telemetry

4.1 Telemetry in Visual Studio

Viewing Telemetry locally in Visual Studio

- Click Run web app
- Click on App Insights (right top corner as shown in image)
- Tick [All]
- Click Update
- Browse web page, you will see telemetry events being loaded...

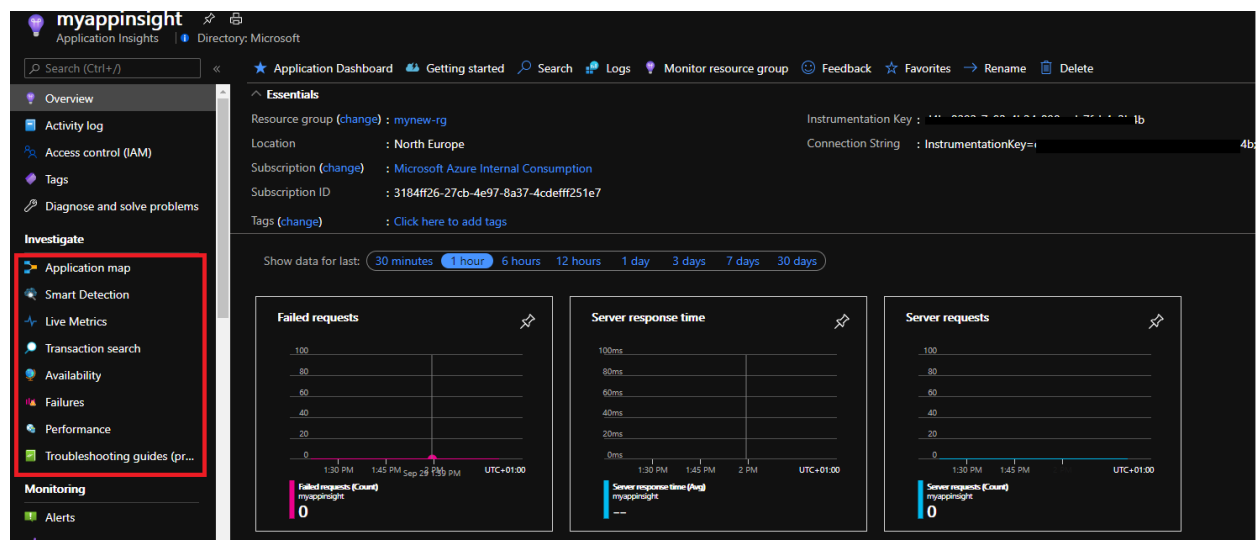


4.2 Telemetry in Azure Portal

Go to Azure portal, open App Insights you created, to view Telemetry.

See telemetry data on following tabs.

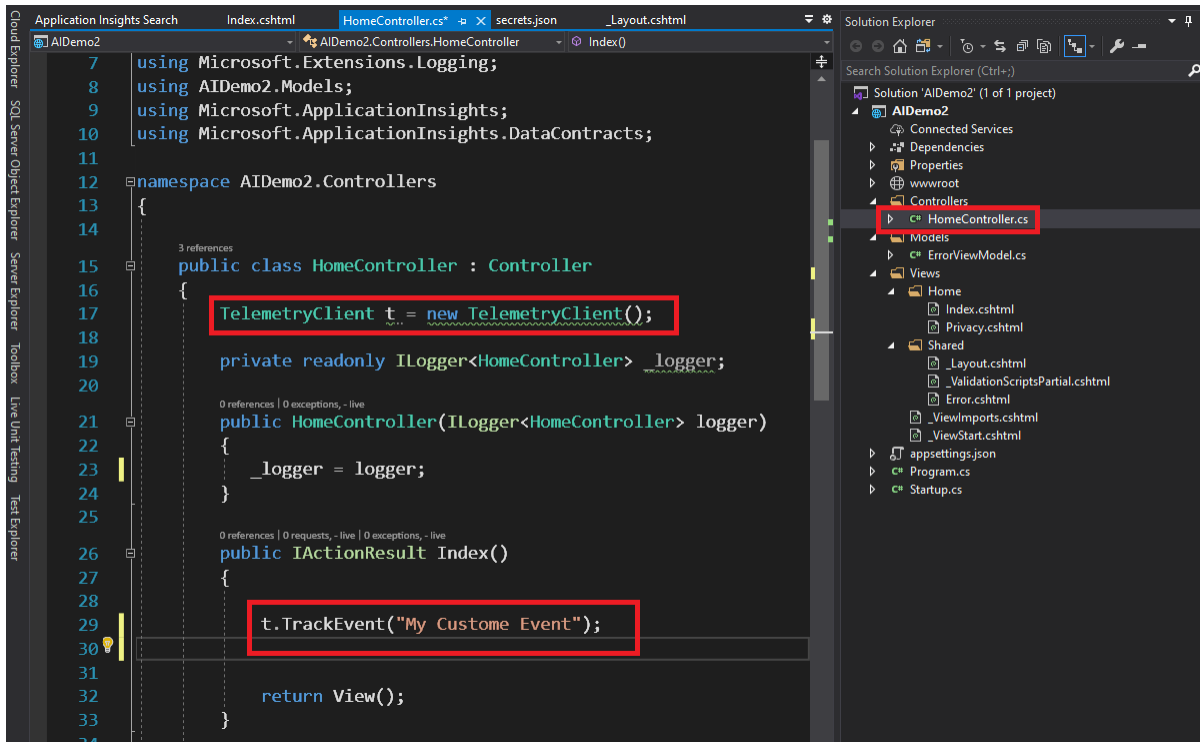
- Application Map
- Smart Detection
- Live Metrics
- Transaction Search
- Availability
- Failure
- Performance



5 Exercise 4 – Adding Custom Events

Adding custom events

1. Go to Controllers/HomeController.cs
2. In the main class declare
`TelemetryClient t = new TelemetryClient();`
3. In Index method add
`t.TrackEvent("My Custome Event");`



```
7 using Microsoft.Extensions.Logging;
8 using AIDemo2.Models;
9 using Microsoft.ApplicationInsights;
10 using Microsoft.ApplicationInsights.DataContracts;
11
12 namespace AIDemo2.Controllers
13 {
14     3 references
15     public class HomeController : Controller
16     {
17         TelemetryClient t = new TelemetryClient();
18
19         private readonly ILogger<HomeController> _logger;
20
21         0 references | 0 exceptions, - live
22         public HomeController(ILogger<HomeController> logger)
23         {
24             _logger = logger;
25         }
26
27         0 references | 0 requests, - live | 0 exceptions, - live
28         public IActionResult Index()
29         {
30             t.TrackEvent("My Custome Event");
31
32             return View();
33         }
34     }
```

4. Navigate to App Insights screen and view your custom event

The screenshot shows the Visual Studio Application Insights interface. At the top, the 'Application Insights (37)' tab is selected. Below the search bar, a table of metrics is displayed, with 'Custom Event' highlighted. The 'Custom Event' row shows a value of 2. Below the metrics table, a list of events is shown, with the first event, '9/29/2020 3:53:15 PM - Custom Event', highlighted. To the right of the event list, the 'Custom Event Details' pane is open, showing the properties of the selected event. The 'Event name' property is highlighted and shows 'My Custom Event'.

Application Insights (37)

Enter search terms, or click the Search icon to show all

Time range: Custom From: 29/09/2020 14:53 To: 29/09/2020 15:53

12 0 2 0 0 2 0 7 1

Results between 29/09/2020 14:53:15 and 29/09/2020 15:53:15

Refine by

Search Fields

9/29/2020 3:53:15 PM - Custom Event
My Custom Event
Operation name: GET Home/Index Operation ID: b3f16223e7c3b4853af1ef5d3c7d10

9/29/2020 3:52:15 PM - Request
GET Home/Index
Request URL: https://localhost:44300/ Response code: 200 Response time: 34ms

9/29/2020 3:52:28 PM - Request
GET /js/site.js
Request URL: https://localhost:44300/js/site.js?v=4q1yofhPaZgrBWAU/Sua9huu0IDg9kP33Vq360 Response code: 304 Response time: 0ms

9/29/2020 3:52:28 PM - Request
GET /js/site.js
Request URL: https://localhost:44300/js/site.js?v=4q1yofhPaZgrBWAU/Sua9huu0IDg9kP33Vq360 Response code: 304 Response time: 1ms

9/29/2020 3:52:28 PM - Request
GET /css/site.css
Request URL: https://localhost:44300/css/site.css Response code: 304 Response time: 17ms

9/29/2020 3:52:28 PM - Request
GET /lib/bootstrap/dist/js/bootstrap.bundle.min.js
Request URL: https://localhost:44300/lib/bootstrap/dist/js/bootstrap.bundle.min.js Response code: 304 Response time: 17ms

9/29/2020 3:52:28 PM - Request
GET /lib/bootstrap/dist/css/bootstrap.min.css
Request URL: https://localhost:44300/lib/bootstrap/dist/css/bootstrap.min.css Response code: 304 Response time: 18ms

9/29/2020 3:52:28 PM - Custom Event

Custom Event Details

Properties

Custom ID: Development

AspNetCoreEnvironment: Development

DeveloperMode: true

Standard ID: 1.0.0.0

Application version: 1.0.0.0

Client IP address: z1

Event name: My Custom Event

Event time: 29/09/2020 14:53:15

Node name: Multi-Surface

Operation ID: b3f16223e7c3b4853af1ef5d3c7d10

Operation name: GET Home/Index

Operation parent ID: eaf933993244b4b

Role instance: Multi-Surface

SDK version: dotnet:2.13.1-12554

5. Try some other custom events...

- Exception
- Page View
- Metric
- TrackTrace