100 XP

Exercise - Install the Application Insights SDK

10 minutes

Sandbox activated! Time remaining: **3 hr 16 min**

You have used 2 of 10 sandboxes for today. More sandboxes will be available tomorrow.

Business stakeholders have given you permission to add the SDK to your video sharing app, with the expectation that the app will begin generating more interesting and useful telemetry.

In this unit, we'll create a web app and add the Application Insights SDK to it. We'll also create an App Service instance for deploying our web app, and configure it to use Application Insights.

Create the web app source code

To create a web app, we'll use a quickstart template that's included with the ASP.NET Core command-line tools.

Run the following command in the Azure Cloud Shell terminal window on the right:

Bash Copy

```
dotnet new mvc -o videowebapp
```

Add the Application Insights SDK

To reference the Application Insights SDK within the app, install the appropriate NuGet packages:

1. In the Cloud Shell, make sure you're in the video app's directory by running the following command:

   Bash Copy

   ```
   cd videowebapp
   ```

2. To add the Application Insights SDK to the app, run this command:

```
Bash
dotnet add package Microsoft.ApplicationInsights.AspNetCore
```

Initialize the Application Insights SDK

To initialize the Application Insights SDK, you need to call the `UseApplicationInsights` method in the *Program.cs* file. Take the following steps:

1. Open the source code for the app in the Cloud Shell editor by running this command:

```
Bash
code .
```

2. Open *Program.cs* by using the code editor's file navigator.

3. Locate the following line of code:

```
C#
public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
WebHost.CreateDefaultBuilder(args)
    .UseStartup<Startup>();
```

4. Add a call to the `UseApplicationInsights` method. The call should look like this:

```
C#
public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
WebHost.CreateDefaultBuilder(args)
    .UseStartup<Startup>()
    .UseApplicationInsights();
```

Important

Be sure to save files when you're done editing them. You can do this by using the ellipsis menu (...) or by using a keyboard shortcut (Ctrl+S on Windows and Linux, Cmd+S on macOS).

Configure a deployment environment

We've created our app and initialized the SDK in our code. Now we need a deployment environment for it. We need a place to host the web app, and we also need an Application Insights resource. We'll follow the best practice mentioned in the last unit and configure the instrumentation key by using an environment variable in the deployment environment.

We'll host our app in Azure App Service. Instead of manually creating an Application Insights resource and configuring its instrumentation key with an application setting, we'll enable Application Insights runtime instrumentation on the app. This will create the Application Insights resource for us and automatically add the `APPINSIGHTS_INSTRUMENTATIONKEY` application setting with the correct value.

1. Go to the Azure portal.
2. On the Azure portal menu or from the **Home** page, select **Create a resource**.
3. Then select **Web** > **Web App**.
4. In the **Web App** window, enter these settings:

Setting	Value
App name	Choose a unique name. Make a note of it. You'll need it later on.
Subscription	**Concierge Subscription**
Resource Group	Select **Use existing** and then select **learn-5313c38b-841c-406a-9fc0-1dcc595fbb96**
OS	**Windows**
Publish	**Code**
App Service plan/Location	Click **Change size** to open the Spec Picker wizard. On the **Dev / Test** tab, select **F1** then select **Apply**.
Application Insights	See the steps that follow.

5. Select **Application Insights**.
6. In the Application Insights window, select **Enable**. An alert will appear indicating that your app will be connected to an automatically created Application Insights resource with the same name as the app.

7. In the **Location** list, select the location closest to you.
8. Scroll to the bottom of the window and select **Apply**.
9. In the Web App window, select **Create**.

Creating your App Service web app will take a minute or two. The portal will notify you when it's finished.

Confirm the Application Insights configuration

After the web app is created, we can see how it's been configured to use Application Insights.

1. In the Azure portal, go to the App Service web app you created. You can use the **All Resources** view to do this.
2. Select **Application Settings** in the navigation menu of the Web App window. Scroll down to where the application settings are listed and select the `APPINSIGHTS_INSTRUMENTATIONKEY` setting to see its value. When your app runs in App Service, this value will be available as an environment variable, and the Application Insights SDK will use it as configuration.
3. In the navigation menu, select **Application Insights**. At the top of the Application Insights window, select **View Application Insights data** to go to the Application Insights resource linked to the web app.
4. The instrumentation key for the Application Insights resource will be shown near the top of the window. Note that it's the same as the one shown in the app's settings.

We've set up our app and deployment environment, but we're not going to deploy the app quite yet. First, we're going to add some custom event telemetry.

Next unit: Instrument the application

Continue >

Azure Cloud Shell