Walkthrough: Instrument an ASP.NET Core app for monitoring in Application Insights

**Sign in and Setting up Application Insight**

**=============================**

**1) Login to Azure**

***az login***

**2) Run the commands below to create the resource group and Application Insights instance. Replace <myLocation> with a region near you.**

***# Create the resource group***

**$myLocation="southeastasia"**

**$myGroup="*az204-appins-rg*"**

***az group create --name $myGroup --location $myLocation***

***# Create App Insights instance***

***az monitor app-insights component create --app az204appinsights --resource-group $myGroup --location $myLocation***

**Note: You may be prompted to install the application-insights extension. If you are be sure to choose to install it.**

**Create an ASP.NET Core app**

**=============================**

**1) Launch Visual Studio Code in the context of the new folder.**

***Code .***

**2) In a terminal on your local machine create a new folder for the project, and change in to the new folder.**

**3) Run the command below in the terminal to create a new ASP.Net Core web app**

***dotnet new webapp***

**4) Open a terminal in Visual Studio Code by selecting Terminal > New Terminal in the menu. Use the command below to add the Microsoft.ApplicationInsights.AspNetCore package to the application.**

***dotnet add package Microsoft.ApplicationInsights.AspNetCore***

**Enable server-side telemetry in the app**

**1) Add services.AddApplicationInsightsTelemetry(); to the ConfigureServices() method in your Startup class, as in this example:**

***// This method gets called by the runtime. Use this method to add services to the container.***

***public void ConfigureServices(IServiceCollection services) {***

 ***// The following line enables Application Insights telemetry collection.***

 ***services.AddApplicationInsightsTelemetry();***

 ***// This code adds other services for your application.***

 ***services.AddRazorPages();***

***}***

**2) Set up the instrumentation key in the app. The following code sample shows how to specify an instrumentation key in the *appsettings.json* file. Replace <INSTRUMENTATION\_KEY> in the code with the key shown in your Application Insights resource in the portal.**

{

  "ApplicationInsights": {

    "InstrumentationKey": "<INSTRUMENTATION\_KEY>"

    },

  "Logging": {

    "LogLevel": {

      "Default": "Information",

      "Microsoft": "Warning",

      "Microsoft.Hosting.Lifetime": "Information"

    }

  },

  "AllowedHosts": "\*"

}

**3) Run the dotnet build command to check for errors. If the build is successful then run the app with the dotnet run command.**

***dotnet build***

***dotnet run***

**4) Open a new browser window and navigate to dotnet run**

**to view your web app and leave it running.**

**5) Switch back to the Azure Portal browser window and select Live Metrics located in the Investigate group. Set the browser window for the app side-by-side with the portal showing the Live Metrics Stream.**

**6) In the app window select the different menu options at the top. Notice the information showing in the Incoming Requests area in the Live Metrics Stream in the portal as you navigate around the web app.**

**7) Return to Visual Studio Code and type Ctrl-C to close the application when you're finished.**

**Clean up resources**

**=============================**

***az group delete --name $myGroup --no-wait***