Introduction to the .NET Core SDK

Preparation

The workshop can be performed using either the full Visual Studio IDE in Windows, or Visual Studio Code in Windows or macOS (or Linux if you're feeling lucky).

When using VSCode, Yeoman is used to scaffold new web projects. To install Yeoman and the ASP.NET-generator, first install Node, then run the following:

```
npm install -g yo generator-aspnet
```

Install the .NET Core SDK (Skip if you have already installed the SDK)

1. Go to https://dot.net and follow the instructions to download and install the .NET Core SDK for your OS

Create and run your first application

1. Open a command prompt
2. Make a new directory to put your application in and change to it

   ```
   mkdir MyNewApp
   cd MyNewApp
   ```

3. Create a new application by typing `dotnet new`
4. Restore the application's dependencies by typing `dotnet restore`
5. Run the application by typing `dotnet run`
6. Open the `Program.cs` file and change the greeting message
7. Run the application again using `dotnet run` and note the message about the application being re-built

Run the project output directly

1. `dotnet run` checks the project source every time to determine if a re-build is necessary and as such is intended for active development scenarios.
2. Run the project output directly by typing `dotnet ./bin/Debug/netcoreapp1.0/MyNewApp.dll`
3. Change the greeting in `Program.cs` again and run the application output directly once more, note that the greeting doesn't change as you didn't re-build the project.
4. Build the project by typing `dotnet build`

5. Run the project output directly again and see the greeting has now changed

Explore the project files

1. Open the `project.json` file in Visual Studio (or VS Code) and explore its contents and try using the IntelliSense to change some project configuration values
2. Look at the files and directories created when the project is built

Make it a web application

1. Add a reference to the ASP.NET Core web server "Kestrel" by adding it to the `"dependencies"` section of the `project.json` file:

   ```
   "dependencies": {
       "Microsoft.AspNetCore.Server.Kestrel": "1.0.0"
   },
   ```

2. At the command prompt, restore the new dependency by typing `dotnet restore`

3. Open the `Program.cs` file and add the following `using` statements to the top of the `Program.cs` file:

   ```
   using Microsoft.AspNetCore.Builder;
   using Microsoft.AspNetCore.Http;
   using Microsoft.AspNetCore.Hosting;
   ```

4. Change the `Main` method to:

   ```
   public static void Main(string[] args)
   {
       var host = new WebHostBuilder()
           .UseKestrel()
           .Configure(app => app.Run(context
               => context.Response.WriteAsync("Hello World!")))
           .Build();

       host.Run();
   }
   ```

5. At the command prompt, run the application using `dotnet run`

6. Open a web browser and browse to http://localhost:5000/ to see the greeting