# Hosting an API Server

# Project Set Up

## Setup Steps for Visual Studio

From within VS (using point and click):

* Open the ConnectingToOtherSystems solution.
* Add console project called HostingAnAPIServerDemo.csproj to the solution
* Use NuGet to add a reference to the following package:
  + Newtonsoft.json

## Setup Steps for Visual Studio Code

From Command Prompt:

* Create console project and add to solution by typing:

dotnet new console --output hosting\_an\_API\_server\_demo

dotnet sln add hosting\_an\_API\_server\_demo

* Install the following packages by typing:

dotnet add hosting\_an\_API\_server\_demo package newtonsoft.json

## Now do the following regardless of project type:

* Replace the Program.cs file’s content with the following:

using System.Net;

using Newtonsoft.Json;

namespace HostingAnAPIServerDemo

{

public class Program

{

static void Main(string[] args)

{

// Simple listener

using (HttpListener listener = new HttpListener())

{

listener.Prefixes.Add("http://localhost:8000/");

listener.Start();

while (true)

{

HttpListenerContext context = listener.GetContext(); // Wait for request

// Extracting information from the request URL

System.Text.StringBuilder sb = new System.Text.StringBuilder();

sb.AppendLine($"Host: {context.Request.Url.Host}");

sb.AppendLine($"Document: {context.Request.Url.AbsolutePath}");

foreach (string key in context.Request.QueryString.AllKeys)

{

sb.AppendLine($"- {key} = {context.Request.QueryString[key]}");

}

string responseText = sb.ToString();

// Extracting the payload body

string bodyText = String.Empty;

if (context.Request.HasEntityBody)

{

using (StreamReader reader = new StreamReader(

context.Request.InputStream,

context.Request.ContentEncoding

))

{

bodyText = reader.ReadToEnd();

}

}

dynamic filmDetails = JsonConvert.DeserializeObject(bodyText);

if (filmDetails != null

&& filmDetails.title != null

&& filmDetails.duration != null

&& filmDetails.certificate != null)

responseText += $"Film is called {filmDetails["title"]} " +

$"it runs for {filmDetails["duration"]} minutes " +

$"and is rated as a {filmDetails["certificate"]}";

else

responseText += "There were NO film details passed in a JSON format"

+ "\n Format should be {\"title\":\"XXX\", \"duration\":\"nnn\", "

+ "\"certificate\":\"XX\"}";

byte[] responseBytes = System.Text.Encoding.UTF8.GetBytes(responseText);

context.Response.StatusCode = 200;

context.Response.StatusDescription = "OK";

context.Response.OutputStream.Write(

responseBytes, 0, responseBytes.Length

);

context.Response.OutputStream.Close();

}

listener.Stop();

}

}

}

}

* Review the code and try to work out what it’s doing.
* Build and run the code and confirm it behaves as you expected.