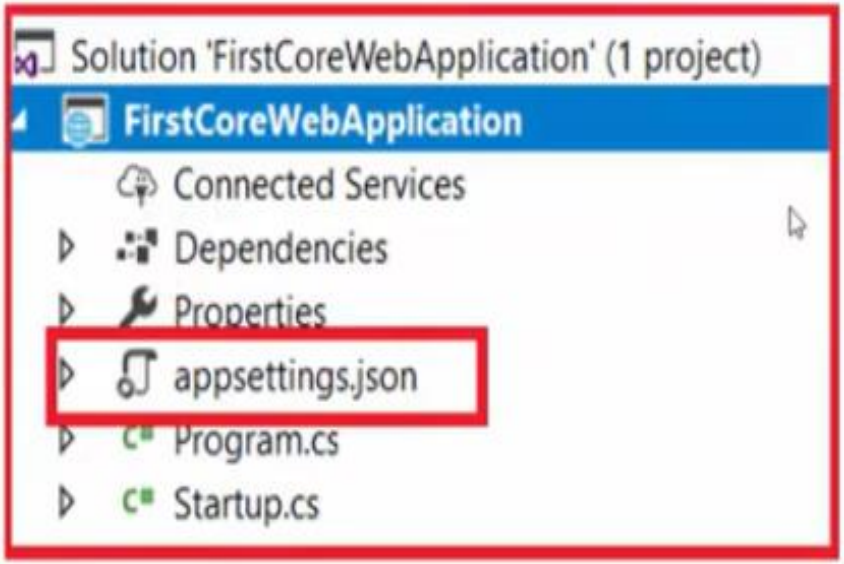
appsettings.json file in ASP.NET CORE

What are the different Configuration Sources available in the ASP.NET Core application?

- If you have worked with the previous versions of the ASP.NET application, then you make know the importance of the web.config file.
- In previous versions of ASP.NET application, we generally used to store the application configuration settings such as database connection strings, any application scope global variables and many more within the web.config file.
- But in ASP.NET Core, the application configuration settings can come from different configurations sources such as
  - **Files** (appsettings.json, appsettings.{Environment}.json, where the {Environment} is the nothing but the applications current hosting environments such as Development, Staging or Production)
  - **User secrets**
  - **Environment variables**
  - **Command-line arguments**

What is ASP.NET Core appsettings.json File?

- When we create an asp.net core web application with an Empty project template, then the visual studio automatically creates the **appsettings.json** file for us as shown in the below image.

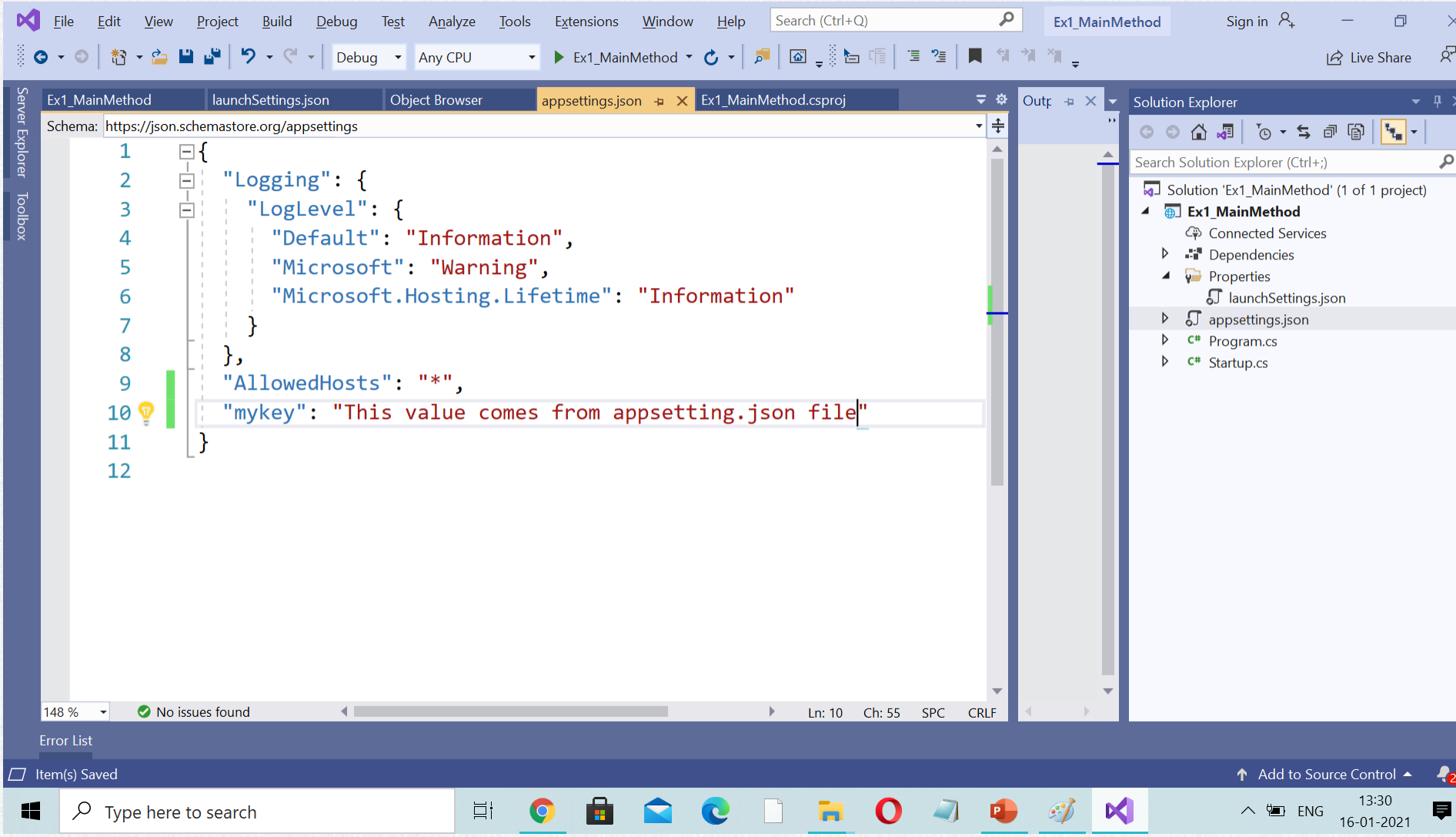


How to access the configuration information in the ASP.NET Core application?

- To access the configuration information within the Startup class, you need to use the **IConfiguration** service which is provided by the ASP.NET Core Framework.
- So what you need to do is just inject the **IConfiguration** service through the constructor of the Startup class. To do so modify the Startup class which is present in the **Startup.cs** file

How to access the value from appSetting.json file:-

1)declare a mykey variable inside that and access in startup.cs file.



- First add a package inside startup.cs file
- using Microsoft.Extensions.Configuration;

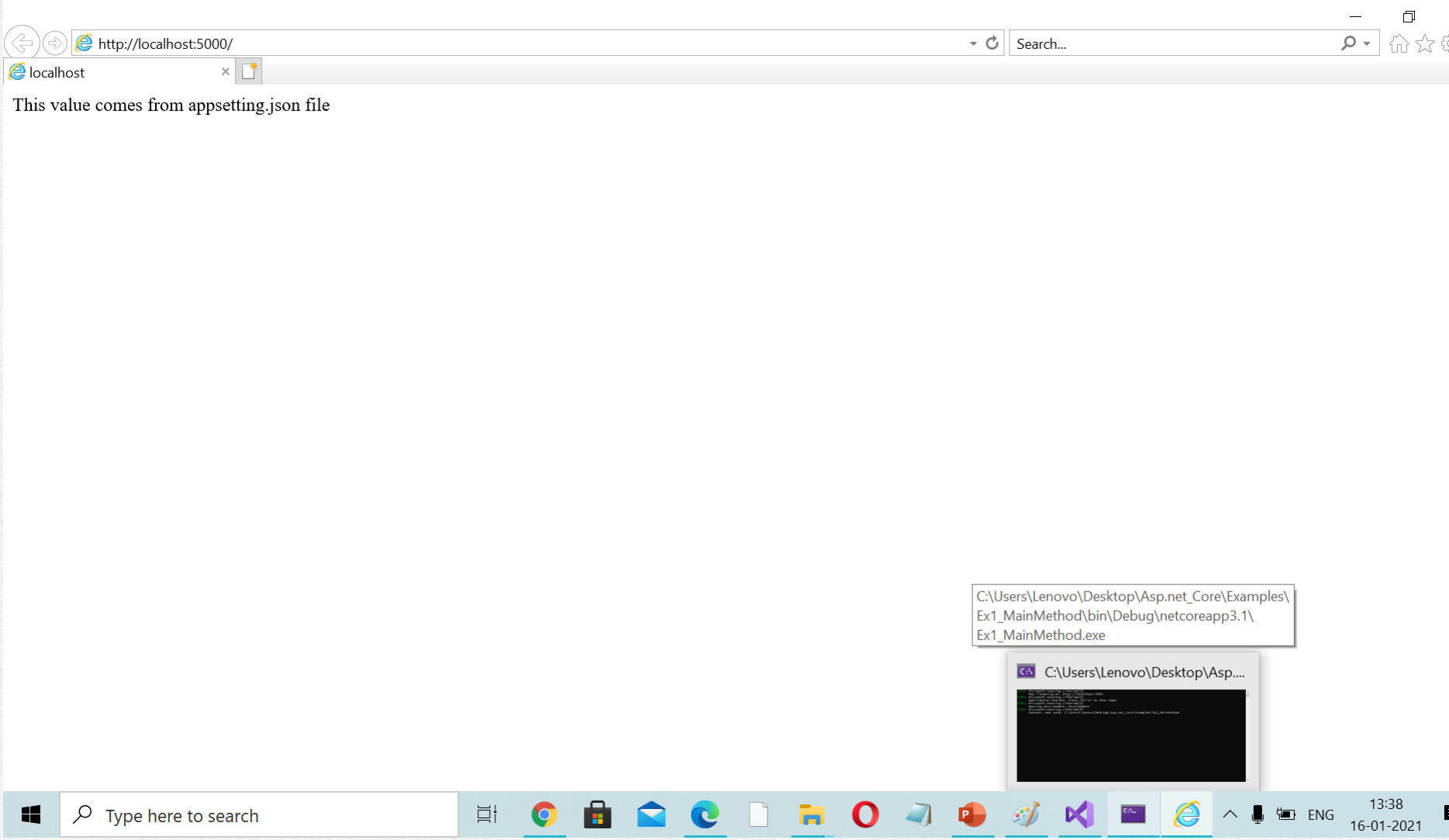
Startup.cs × | launchSettings.json | appsettings.json ×

mywebapp > Startup.cs

D:\mywebapp\appsettings.json

```
12  {
13      public class Startup
14      {
15          // This method gets called by the runtime. Use this method to add services to the container.
16          // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940
17
18          private IConfiguration _config;
19          public Startup(IConfiguration conifg)
20          {
21              _config=config;
22          }
23
24          public void ConfigureServices(IServiceCollection services)
25          {
26          }
27
28          // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
29          public void Configure(IApplicationBuilder app, IHostingEnvironment env)
30          {
31              if (env.IsDevelopment())
32              {
33                  app.UseDeveloperExceptionPage();
34              }
35
36              app.Run(async (context) =>
37              {
38                  await context.Response.WriteAsync(_config["mykey"]);
39              });
40          }
```

Output:-

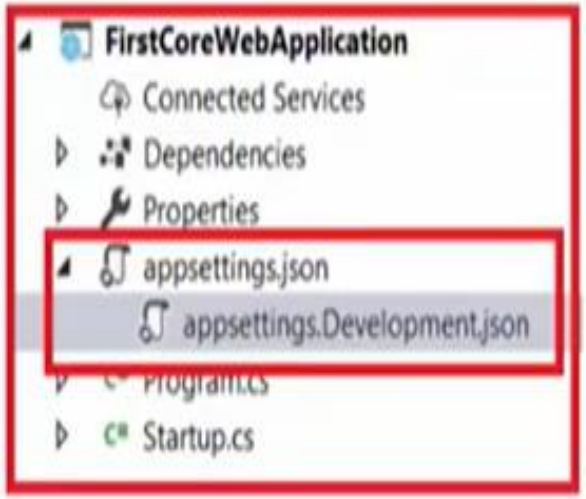

http://localhost:5000/
Search...
localhost
This value comes from appsetting.json file
C:\Users\Lenovo\Desktop\Asp.net_Core\Examples\Ex1_MainMethod\bin\Debug\netcoreapp3.1\Ex1_MainMethod.exe
C:\Users\Lenovo\Desktop\Asp....
Type here to search
ENG
13:38
16-01-2021

Dependency Injection

- In previous versions of ASP.NET applications, the Dependency Injection design pattern was optional. But if you want to configure it in your application, then you need to use some of the frameworks like Ninject, StructureMap, IUnity container, etc.
- But in ASP.NET Core application Dependency Injection is an integral part and the framework provides the inbuilt support for dependency injection. The Dependency Injection Design Pattern allows us to develop loosely coupled systems that are extensible and also easy to testable.

What is the Configuration Execution Order in ASP.NET Core Application?

- Before understanding the execution order, let's have a look at the appsettings.Development.json file. You can find this file within the appsettings.json file as shown below



Same key tag is define in appSettings.Development.json file this file data access first in stratup.cs file

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Ex1_MainMethod Sign in

Debug Any CPU Ex1_MainMethod Live Share

appsettings.Development.json | Ex1_MainMethod | launchSettings.json | appsettings.json | Ex1_MainMethod.csproj | Startup.cs | Object Browser

Schema: <No Schema Selected>

```
1  {
2    "Logging": {
3      "LogLevel": {
4        "Default": "Information",
5        "Microsoft": "Warning",
6        "Microsoft.Hosting.Lifetime": "Information"
7      }
8    },
9    "mykey": "This value comes from appsetting.Development.json file"
10 }
11
```

Output

```
\Ex1_MainMethod\bin\Debug\netcoreapp3.1\Ex1_MainMethod.dll
==========
Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped
==========
```

163 % No issues found Ln: 11 Ch: 1 SPC LF

Error List

Build succeeded Add to Source Control

Type here to search ENG 14:51 16-01-2021

What is the Default Orders of reading the configuration sources?

- The default orders in which the various configuration sources are read for the same key are as follows
  - appsettings.json,
  - appsettings.{Environment}.json here we use appsettings.development.json
  - User secrets
  - Environment variables
  - Command-line arguments

How to Pass Config value from Command Line in ASP.NET Core Application?

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 1: powershell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\> cd mywebapp
PS D:\mywebapp> dotnet run mykey="This is command Line argument"
Hosting environment: Development
PS D:\mywebapp> dotnet run mykey="This is command Line argument"
Hosting environment: Development
Content root path: D:\mywebapp
Now listening on: https://localhost:5001
Now listening on: http://localhost:5000
Application started. Press Ctrl+C to shut down.
info: Microsoft.AspNetCore.Hosting.Internal.WebHost[1]
      Request starting HTTP/1.1 GET http://localhost:5000/
info: Microsoft.AspNetCore.Hosting.Internal.WebHost[2]
      Request finished in 33.3536ms 200
info: Microsoft.AspNetCore.Hosting.Internal.WebHost[1]
      Request starting HTTP/1.1 GET http://localhost:5000/favicon.ico
info: Microsoft.AspNetCore.Hosting.Internal.WebHost[2]
      Request finished in 0.3086ms 200
Application is shutting down...
PS D:\mywebapp>
```

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Ex1_MainMethod Sign in

Debug Any CPU Ex1_MainMethod Live Share

Object Browser | appsettings.Development.json | Ex1_MainMethod | launchSettings.json | appsettings.json | Ex1_MainMethod.csproj | Startup.cs | Output

Schema: https://json.schemastore.org/launchsettings

```
13          "launchBrowser": true,
14          "environmentVariables": {
15            "ASPNETCORE_ENVIRONMENT": "Development"
16          }
17        },
18        "Ex1_MainMethod": {
19          "commandName": "Project",
20          "launchBrowser": true,
21          "applicationUrl": "http://localhost:5000",
22          "environmentVariables": {
23            "ASPNETCORE_ENVIRONMENT": "Development",
24            "mykey": "This value comes from launchSettings.json file"
25          },
26
27        }
28      }
29    }
30
```

```
.
The program
'[0x50FC]
iexplore.e
xe' has
exited
with code
0 (0x0).
The program
'[0x5510]
Ex1_MainMe
thod.exe'
has exited
with code
-1
(0xffffff
f).
```

163 % 0 1 Ln: 25 Ch: 9 SPC CRLF

Error List

Ready Add to Source Control

Type here to search ENG 15:03 16-01-2021

http://localhost:5000/

Search...

localhost

This value comes from launchSettings.json file

Type here to search

ENG

15:03
16-01-2021

Thanks