



EBook Gratis

APRENDIZAJE

asp.net-core-mvc

Free unaffiliated eBook created from
Stack Overflow contributors.

**#asp.net-
core-mvc**

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con asp.net-core-mvc.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Instalando Visual Studio.....	2
Creando una aplicación ASP.NET Core MVC.....	2
Añadir MVC Middleware.....	5
Fundamentos de inyección de dependencia.....	6
Gestión de por vida.....	7
Versiones.....	7
Capítulo 2: Cambiar ubicación de vista predeterminada.....	9
Introducción.....	9
Examples.....	9
Crear un expansor de ubicación de vista.....	9
Registrar el expansor de ubicación de vista.....	9
Capítulo 3: Instale e instale .Net Core MVC con código de Visual Studio y arranque rápido	10
Introducción.....	10
Observaciones.....	10
Examples.....	10
Paso 1 - Instalación de código de estudio visual.....	10
Paso 2 - Configuración de .Net core y C #.....	13
Paso 3 - Crear una plantilla MVC básica.....	19
Paso 4 - Ejecutar y depurar la aplicación.....	22
Creditos.....	25

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [asp-net-core-mvc](#)

It is an unofficial and free asp.net-core-mvc ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official asp.net-core-mvc.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con asp.net-core-mvc

Observaciones

Esta sección proporciona una descripción general de qué es asp.net-core-mvc, y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de asp.net-core-mvc, y vincular a los temas relacionados. Dado que la Documentación para asp.net-core-mvc es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

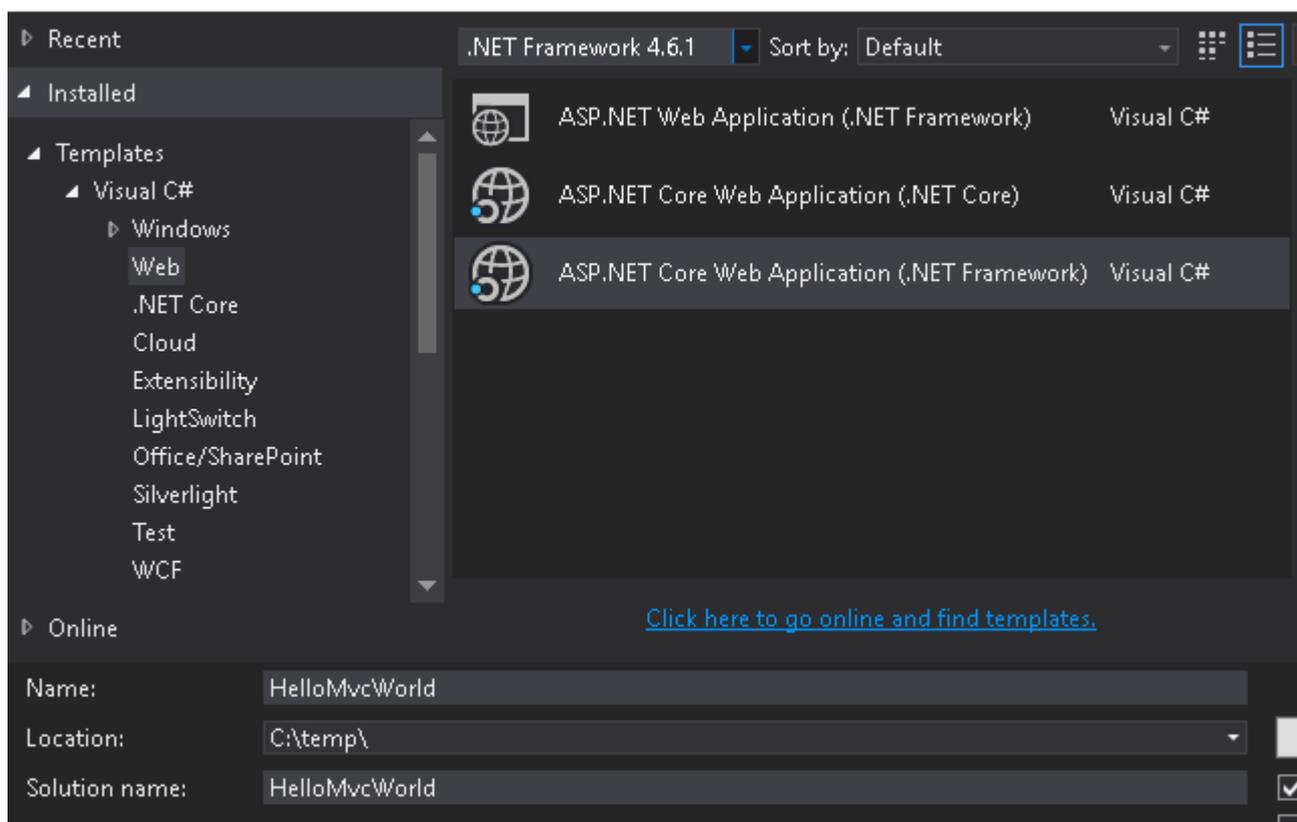
Instalando Visual Studio

Si no tiene instalado Visual Studio, puede [descargar la Edición de comunidad de Visual Studio gratis aquí](#) . Si ya lo tiene instalado, puede continuar con el siguiente paso.

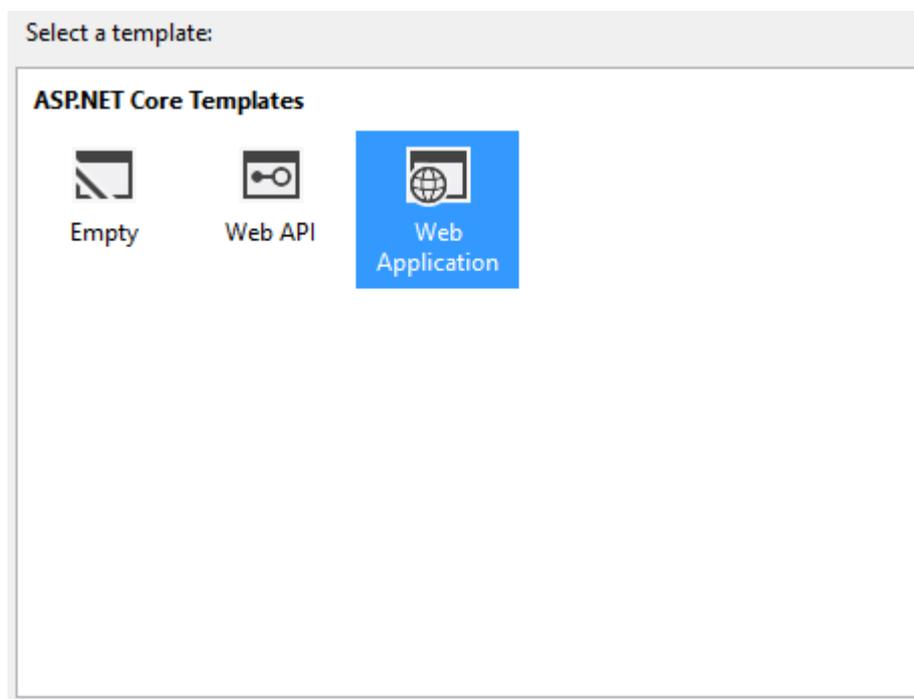
Creando una aplicación ASP.NET Core MVC.

1. **Abra Visual Studio.**
2. **Seleccione Archivo> Nuevo proyecto.**
3. **Seleccione Web en el idioma de su elección** dentro de la sección Plantillas a la izquierda.
4. **Elija un tipo de proyecto preferido** dentro del diálogo.
5. **Opcional: elija un .NET Framework al que le gustaría apuntar**
6. **Nombre su proyecto** e indique si desea crear una solución para el proyecto.
7. **Haga clic en Aceptar** para crear el proyecto.

New Project



Se le presentará otro cuadro de diálogo para seleccionar la plantilla que desea utilizar para el proyecto:

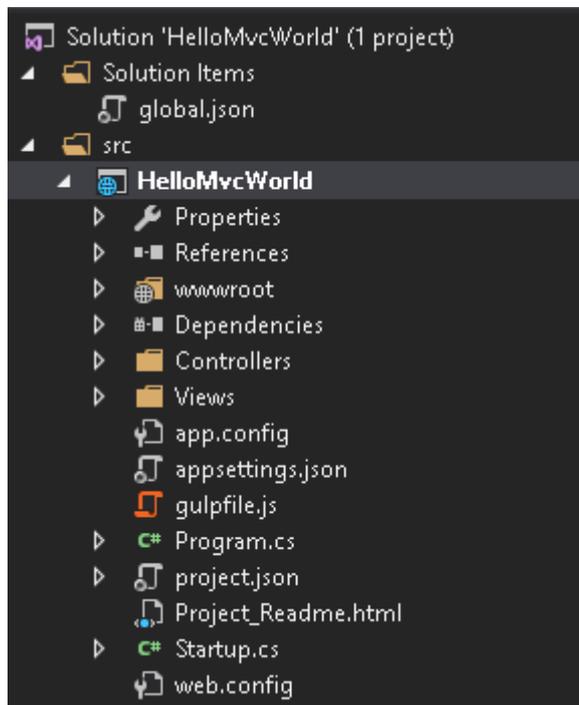


Cada una de las descripciones se explica por sí misma. Para este primer proyecto, **seleccione Aplicación Web**, que contendrá todas las configuraciones predeterminadas, la autenticación y algunos contenidos existentes.

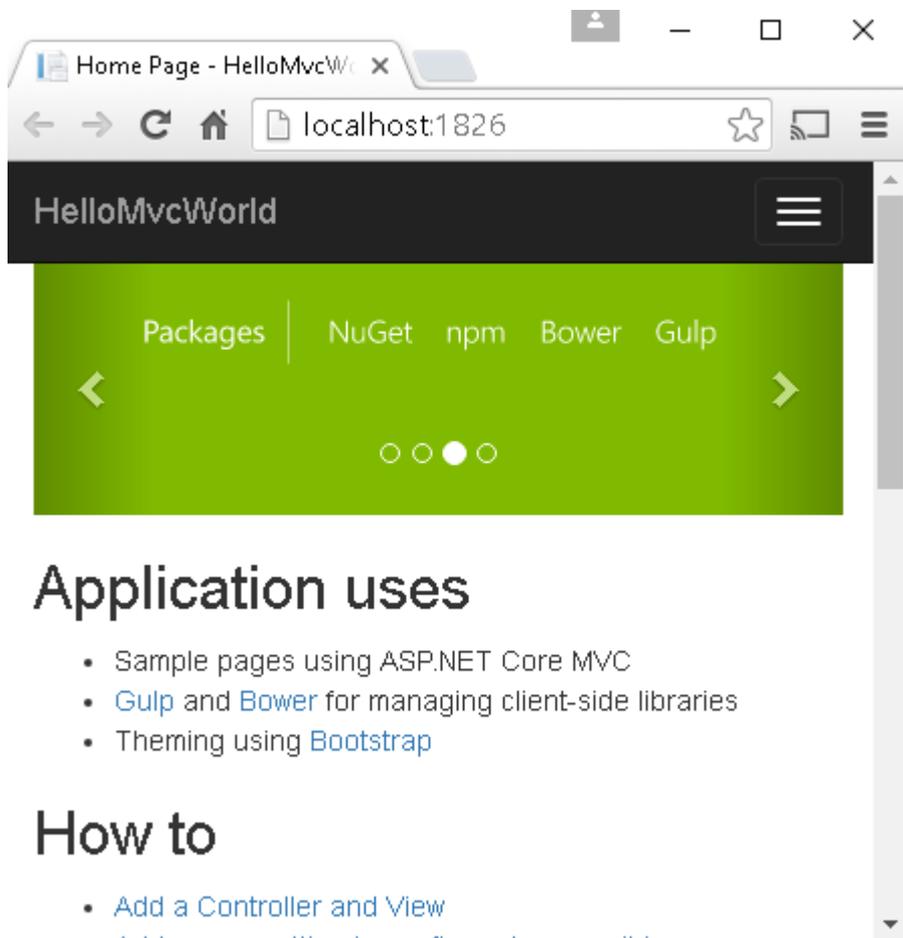
Dado que se trata de una aplicación de introducción y no requiere seguridad ni autenticación,

puede **cambiar la opción de autenticación a Sin autenticación** en el lado derecho del cuadro de diálogo y **hacer clic en Aceptar para crear el proyecto** .

Entonces debería ver el nuevo proyecto dentro del Explorador de soluciones:



Presione la tecla F5 para ejecutar la aplicación y comenzar una sesión de depuración, que iniciará la aplicación dentro de su navegador predeterminado:



Ahora puede ver que su proyecto está funcionando y funcionando localmente y está listo como un punto de partida para que construya su aplicación.

PS: Usado [Comenzando con el tema asp.net-core de la documentación de asp.net-core](#) .

Añadir MVC Middleware

Si creó un proyecto vacío, o aún no tiene mvc configurado en su aplicación, puede agregar dependencia:

```
"Microsoft.AspNetCore.Mvc": "1.0.1"
```

A su archivo `project.json` bajo "dependencies" .

Y registre el middleware MVC en su clase de inicio:

```
public void ConfigureServices(IServiceCollection services)
{
    ...
    services.AddMvc();
}
```

Tenga en cuenta que tenemos tanto `services.AddMvc()` como `services.AddMvcCore()` . Si está comenzando con el `asp.net core` , o quiere que sea como es, debe seguir con `services.AddMvc()` . Pero si desea una experiencia avanzada, puede comenzar con un pipeline MVC mínimo y agregar características para obtener un marco personalizado utilizando `services.AddMvcCore()` . Consulte [esta discusión](#) para obtener más información sobre `AddMvcCore`

```
public void ConfigureServices(IServiceCollection services)
{
    services
        .AddMvcCore()
        .AddAuthorization()
        .AddJsonFormatters(j => j.Formatting = Formatting.Indented);
}
```

Ahora puede decirle a su generador de aplicaciones que use el mvc:

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory
loggerFactory)
{
    ...
    app.UseMvc();
}
```

o con enrutamiento por defecto:

```
app.UseMvc(routes =>
{
    routes.MapRoute(
        name: "default",
        template: "{controller=Home}/{action=Index}/{id?}");
});
```

Fundamentos de inyección de dependencia

Casi cualquier controlador necesita algunas dependencias externas para funcionar. Aquí hay una forma de configurar un objeto de dependencia (o su fábrica) y pasarlo a un controlador. Si lo hace, ayudará a mantener una [separación de preocupaciones](#), mantenga el código claro y verificable.

Digamos que tenemos una interfaz y su implementación que necesita algunos valores de config en su constructor:

```
public interface ISomeDependency
{
    async Task<IEnumerable<string>> GetItemsAsync(string key);
}

public class SomeDependency : ISomeDependency
{
    public SomeDependency(string connectionString)
    {
        ...
    }
    ...
}
```

Se usa en alguna clase de controlador:

```
public class SomeController : Controller
{
    private readonly ISomeDependency dependency;

    public SomeController(ISomeDependency dependency)
    {
        ...
        this.dependency = dependency;
    }

    ...

    public async Task<IEnumerable<string>> Get(string key) =>
        await dependency.GetItemsAsync(key);
}
```

Uno puede inyectar esta dependencia en el constructor del controlador que llama a los `services.AddTransient Startup.ConfigureServices` método de `Startup.ConfigureServices` dentro del método `Startup.ConfigureServices`:

```
public class Startup
{
    public Startup(IHostingEnvironment env)
    {
        var builder = new ConfigurationBuilder()
            .SetBasePath(env.ContentRootPath)
            .AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)
            ...
        Configuration = builder.Build();
    }
}
```

```

}

public IConfigurationRoot Configuration { get; }

public void ConfigureServices(IServiceCollection services)
{
    ...
    services.AddTransient(serviceProvider =>
        new MyDependency(Configuration["Data:ConnectionString"]));
}

...
}

```

Aquí `Data:ConnectionString` es una ruta a una configuración en el archivo `appsettings.json` :

```

{
    ...
},
"Data": {
    "ConnectionString": "some connection string"
}
}

```

Gestión de por vida

Para administrar la vida útil del objeto inyectado, junto con `AddTransient` , existen otras dos opciones: `AddSingleton` y `AddScoped` . El último significa que el tiempo de vida del objeto está sujeto a una solicitud HTTP.

Versiones

[Hoja de ruta oficial @ Github](#)

Versión	Anuncios	Fecha de lanzamiento
RC1 *	1.0.0-rc1	2015-11-01
RC2 *	1.0.0-rc2	2016-05-16
1.0.0	1.0.0	2016-06-27
1.0.1	1.0.1	2016-09-13
1.0.1	1.0.1	2016-09-13
1.1	1.1.0	Q4 2016 / Q1 2017
1.2	1.2.0	Q1 2017 / Q2 2017

* Las referencias a los trimestres anuales (Q1, Q2, Q3, Q4) están basadas en el calendario

Lea Empezando con asp.net-core-mvc en línea: <https://riptutorial.com/es/asp-net-core-mvc/topic/2174/empezando-con-asp-net-core-mvc>

Capítulo 2: Cambiar ubicación de vista predeterminada

Introducción

En ASP.NET MVC, las vistas se colocan de forma predeterminada en la carpeta `Views`. A veces desea cambiar estas ubicaciones y almacenar las vistas en otro lugar.

Examples

Crear un expansor de ubicación de vista

Para poder cambiar la ubicación de la vista, debe implementar el `IViewLocationExpander`. El método `ExpandViewLocations` devuelve una `IEnumerable<string>` contiene las diferentes ubicaciones donde buscar, con

```
public class MyViewLocationExpander : IViewLocationExpander
{
    public IEnumerable<string> ExpandViewLocations(ViewLocationExpanderContext context,
        IEnumerable<string> viewLocations)
    {
        yield return "/CustomViewFolder/{1}/{0}.cshtml";
        yield return "/SharedFolder/{0}.cshtml";
    }

    public void PopulateValues(ViewLocationExpanderContext context)
    {
    }
}
```

Registrar el expansor de ubicación de vista

Ahora debe registrar el Expander para que lo utilice el motor de Razor View. Solo agregue esto en los servicios de `ConfigureServices` de su clase de `Startup`.

```
public void ConfigureServices(IServiceCollection services)
{
    services.Configure<RazorViewEngineOptions>(options => {
        options.ViewLocationExpanders.Add(new MyViewLocationExpander());
    });
}
```

Lea [Cambiar ubicación de vista predeterminada en línea](https://riptutorial.com/es/asp-net-core-mvc/topic/8669/cambiar-ubicacion-de-vista-predeterminada): <https://riptutorial.com/es/asp-net-core-mvc/topic/8669/cambiar-ubicacion-de-vista-predeterminada>

Capítulo 3: Instale e instale .Net Core MVC con código de Visual Studio y arranque rápido .net core mvc hello world.

Introducción

Este artículo da una idea acerca de la configuración e instalación del núcleo Asp.Net con código de estudio visual. También crea plantillas MVC básicas y depuración.

Pasos involucrados abajo ...

Paso 1 - instalando el código de Visual Studio.

Paso 2 - Configuración de .Net core y C #.

Paso 3 - Crear una plantilla MVC básica.

Paso 4 - Ejecutar y depurar la aplicación.

Observaciones

Este artículo está a punto de configurarse desde cero con el código abierto de Visual Studio y crear y depurar aplicaciones básicas .net core mvc.

- La ubicación del archivo utilizada anteriormente es un cambio según los usuarios, sin restricciones.
- Necesita internet para descargar configuraciones.

Examples

Paso 1 - Instalación de código de estudio visual

- Descargue el código de [Visual Studio](#) desde aquí. Seleccione su instalador de destino [mac | windows | linux].

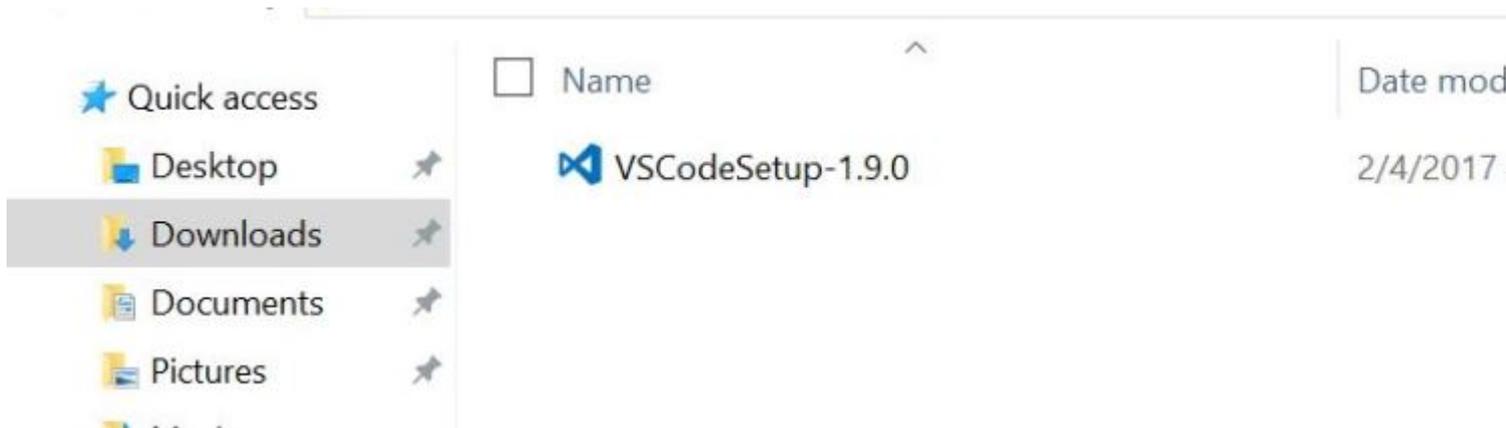
Code editing. Redefined.

Free. Open source. Runs everywhere.

Download for Windows		Stable Build	
		Stable	Insiders
macOS	Package	↓	↓
Windows	Installer	↓	↓
	.zip	↓	↓
Linux x64 32bit versions	.deb	↓	↓
	.rpm	↓	↓
	.tar.gz	↓	↓



- Ir al archivo descargado en su local.



- A continuación se detallan los pasos para la instalación.



Recycle Bin



Andi AR



VSCODE



Tamil

Setup - Visual Studio Cod



2. Presione [**ctrl + P**]
3. **Pega " ext install csharp "** y **pulsa .**

Una vez hecho los pasos anteriores, la extensión C # está disponible en VS Code.

EXTENSIONS

Search Extensions in Marketplace

C# 1.6.2 1M 4

C# C# for Visual Studio Code (powered by OmniSharp).
Microsoft

HomeControl

Details C#

C# for V

Maste

build pas

Welcome to

- Light
- Great
- Debu
- Supp

PROBLEMS

- Ahora configura .net core.

Descarga .net core sdk desde [aquí](#) . Elija Windows => CommandLine.

Windows

Visual Studio 2015

Install for

1

Install .NET Core SDK

To start creating .NET Core apps you need to install the .NET Core SDK for Windows.

[Download .NET Core 1.1 SDK](#)

.NET Core 1.1 is the latest version. For more information and additional downloads check the

Instale el SDK como a continuación.



Recycle Bin



Google
Chrome



Visual Studio
Code



Andi AR



Tamil



VSCODE

Microsoft .NET Core 1.1.0 - SD

Microsoft .NET

You just need a shell,
a text editor and 10
minutes of your time.

Ready? Set? Let's go!

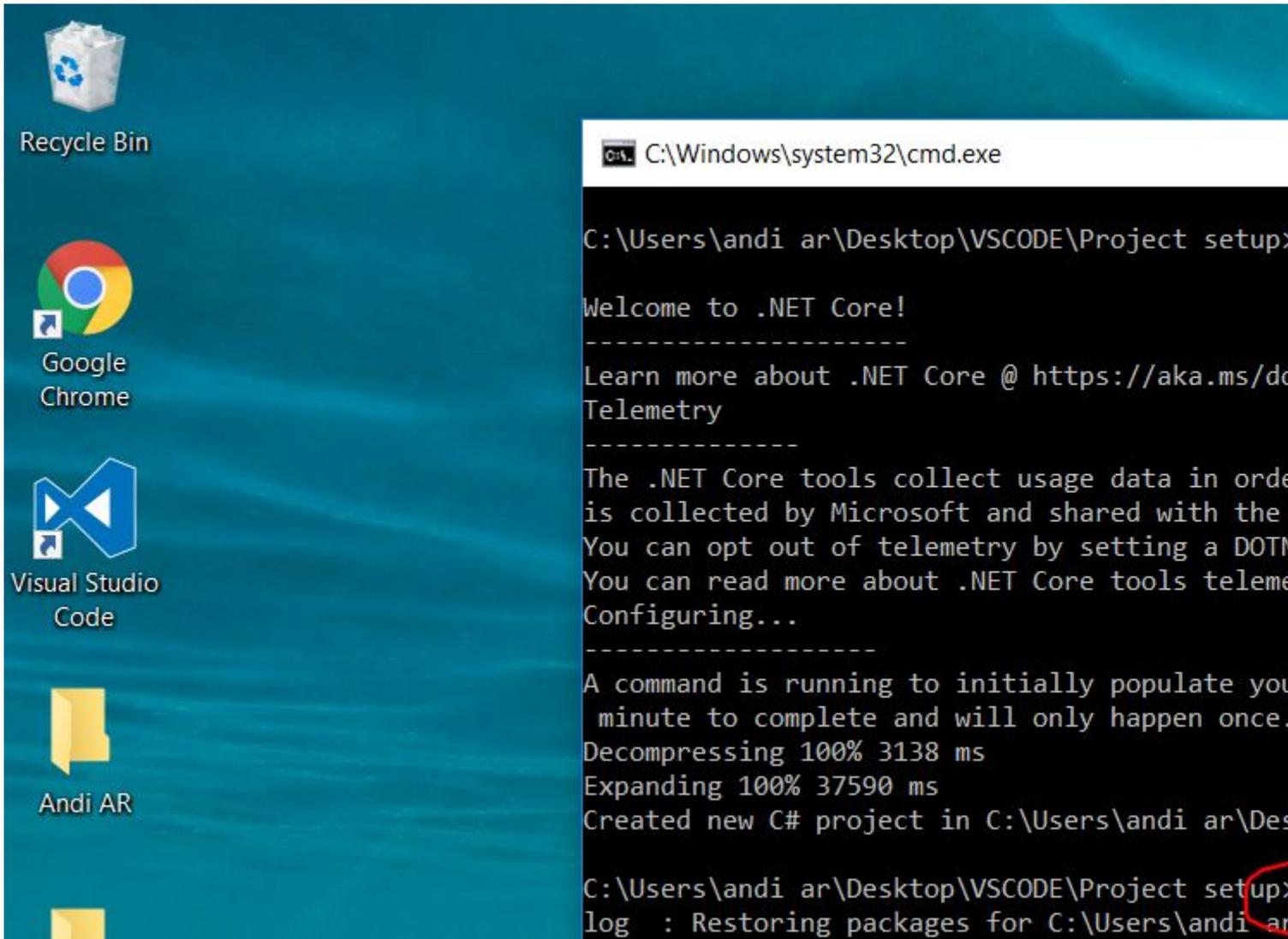
Microsoft®
.NET

MI

MI

The
Cor
and
abo
The

" y pulsa. Esto es para restaurar todos los paquetes del archivo project.json.





```
C:\Windows\system32\cmd.exe
log : Installing Microsoft.EntityFrameworkCore
log : Installing Microsoft.DotNet.InternalAbstractions
log : Installing SQLite.Native 3.12.2.
log : Installing Microsoft.EntityFrameworkCore
log : Restoring packages for tool 'Microsoft.EntityFrameworkCore.Tools'
log : Installing Microsoft.NETCore.DotNetHostResolver
log : Installing Microsoft.NETCore.DotNetHostResolver
log : Installing Microsoft.NETCore.Jit 1.0.2.
log : Installing System.Net.Requests 4.0.11.
log : Installing System.Linq.Parallel 4.0.1.
log : Installing System.Reflection.DispatchProxy
log : Installing System.Threading.Tasks.Dataflow
log : Installing Microsoft.VisualBasic 10.0.1.
log : Installing System.IO.MemoryMappedFiles
log : Installing System.IO.UnmanagedMemoryStream
log : Installing Microsoft.NETCore.App 1.0.0.
log : Installing System.Net.WebHeaderCollection
log : Installing Microsoft.NETCore.DotNetHostResolver
log : Installing Microsoft.NETCore.Runtime.CoreCLR
log : Restoring packages for tool 'Microsoft.EntityFrameworkCore.Tools'
log : Installing Microsoft.AspNetCore.Server.Kestrel
log : Restoring packages for tool 'Microsoft.EntityFrameworkCore.Tools'
log : Restoring packages for tool 'Microsoft.EntityFrameworkCore.Tools'
log : Installing Microsoft.Extensions.SecretManager
log : Restoring packages for tool 'Microsoft.EntityFrameworkCore.Tools'
log : Installing Microsoft.VisualStudio.Web.CodeGeneration
log : Installing Microsoft.VisualStudio.Web.CodeGeneration
log : Installing Microsoft.AspNetCore.Server.Kestrel
log : Installing Microsoft.VisualStudio.Web.CodeGeneration
log : Writing lock file to disk. Path: C:\Users\andi ar\Desktop\VSCODE\Project
log : C:\Users\andi ar\Desktop\VSCODE\Project
log : Restore completed in 515443ms.

C:\Users\andi ar\Desktop\VSCODE\Project setup
```

- Ahora inicie VScode y abra la carpeta del proyecto.

File Edit Selection View Go Help

- New File Ctrl+N
- New Window Ctrl+Shift+N
- Open File...
- Open Folder...
- Open Recent ▶
- Save Ctrl+S
- Save As... Ctrl+Shift+S
- Save All
- Auto Save
- Preferences ▶
- Revert File
- Close Editor Ctrl+F4
- Close Folder
- Close Window Ctrl+Shift+W
- Exit

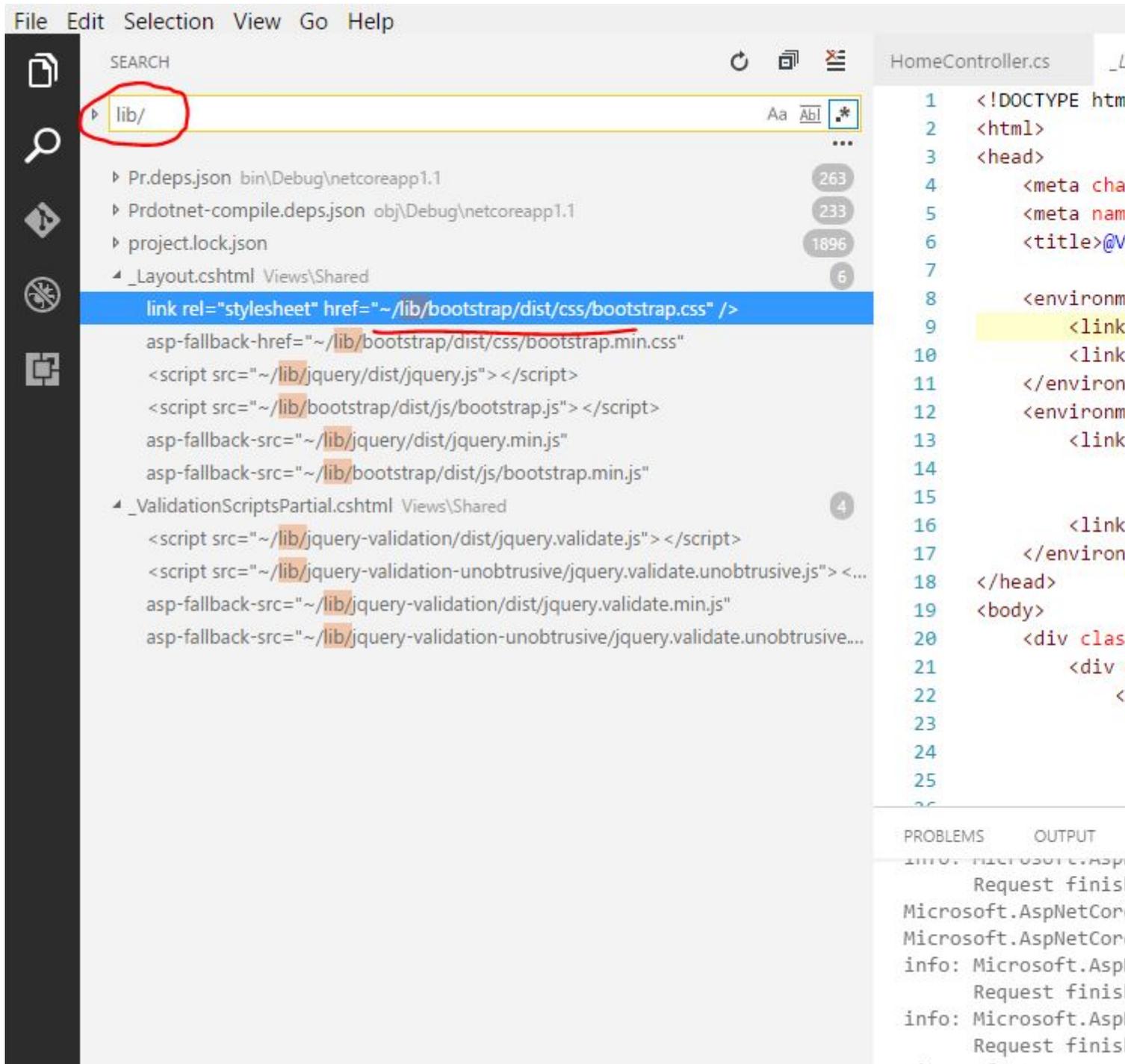
PROBLEMS OUTPUT

Updating C# dependencies
Platform: win32, x86

Downloading package

en el código VS e ingrese el nombre del archivo perdido y busque.]

Solucione esto agregando scripts con cdn o la ubicación exacta del archivo en el archivo de diseño.



Ahora refresca la página y mira.

ASP.NET Core



Learn how to bu

HTML1300: Navigation occurred.
localhost:5000

Creditos

S. No	Capítulos	Contributors
1	Empezando con asp.net-core-mvc	Community , Ole K , Rafael Marques , Set , stop-cran , tmg , Zach Becknell
2	Cambiar ubicación de vista predeterminada	glacasa
3	Instale e instale .Net Core MVC con código de Visual Studio y arranque rápido .net core mvc hello world.	Andi AR