



EBook Gratis

APRENDIZAJE nancy

Free unaffiliated eBook created from
Stack Overflow contributors.

#nancy

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con Nancy	2
Observaciones.....	2
Acerca de:.....	2
Recursos adicionales:.....	2
Versiones.....	2
Examples.....	2
Crear una aplicación Nancy autoalimentada simple.....	2
Configure Nancyfx con Dotnet core v1.1, Kestrel y Visual Studio Code en sistemas * nix.....	3
Prerequiste pasos:.....	3
Crear un proyecto NancyFx auto alojado:.....	3
Capítulo 2: Estado de soporte de .net core	7
Introducción.....	7
Examples.....	7
Estado.....	7
Núcleo de Nancy	7
Nancy Boostrappers	7
Nancy ver los motores	7
Autenticación Nancy	7
Nancy FluentValidation	7
Cambios de enrutamiento.....	7
Creditos	9

Acerca de

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

It is an unofficial and free nancy 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 nancy.

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 Nancy

Observaciones

Acerca de:

Nancy es un marco ligero para la construcción de servicios basados en HTTP en .Net basado en el marco Sinatra que existe para Ruby. Está diseñado para permitir el manejo de varios tipos diferentes de solicitudes HTTP y proporciona una forma sencilla de devolver una respuesta con una pequeña cantidad de código.

Recursos adicionales:

- Sitio web principal: <http://nancyfx.org>
- Documentación: <https://github.com/NancyFx/Nancy/wiki>
- Código fuente: <https://github.com/NancyFx/Nancy>
- Paquete Nuget: <https://www.nuget.org/packages/Nancy/>

Versiones

Versión	Fecha de lanzamiento
Nancy 1.4.3	2015-12-21
Nancy 1.4.2	2015-11-23
Nancy 1.4.1	2015-11-05
Nancy 1.4.0	2015-10-29
Nancy 1.3.0	2015-09-25
Nancy 1.2.0	2015-04-17
Nancy 1.1.0	2015-02-18
Nancy 1.0.0	2015-01-23

Examples

Crear una aplicación Nancy autoalimentada simple

1. Use Nuget para instalar los paquetes Nancy y Nancy.Hosting.Self en el proyecto.
2. Cree una instancia de un nuevo objeto NancyHost y pase la URL correspondiente

```
using( var host = new NancyHost( hostConfiguration, new Uri( "http://localhost:1234" ) ) )
{
    host.Start();
    Console.WriteLine( "Running on http://localhost:1234" );
    Console.ReadLine();
}
```

Coloque este código en su proyecto en el momento en que desee comenzar a escuchar el tráfico http.

3. Agregue una clase a su proyecto que herede de NancyModule y agregue un método de construcción.

```
public class FooModule : NancyModule
{
    public FooModule()
    {
    }
}
```

4. Definir rutas en el constructor:

```
...
public FooModule()
{
    Get["Bar"] = parameters => {
        return "You have reached the /bar route";
    }
}
```

Configure Nancyfx con Dotnet core v1.1, Kestrel y Visual Studio Code en sistemas * nix

Prerequiste pasos:

1. Consigue dotnet core para tu plataforma:

[Dotnet Core](#)

2. Siga las instrucciones y asegúrese de que el núcleo dotnet esté funcionando
3. Obtenga Visual Studio Code para su plataforma:

[Código VS](#)

4. Inicie Visual Studio Code (código VS) e instale la extensión C # y luego vuelva a cargar

Crear un proyecto NancyFx auto alojado:

1. Configurar un proyecto con una estructura de directorio de proyecto correcta.

Abre la Terminal de Bash y escribe:

```
mkdir nancydotnetcore
cd nancydotnetcore
mkdir src
mkdir test
touch global.json
```

2. Abra global.json e ingrese el siguiente código:

```
{
  "projects":["src", "test"]
}
```

3. En la terminal de Bash:

```
cd src
mkdir NancyProject1
dotnet new
```

Abra la carpeta NancyProject1 en el código VS

Aparecerá una advertencia: "Faltan los activos necesarios para compilar y depurar en 'nancyproject1'".

Haga clic en "si"

También verá: hay dependencias no resueltas de 'project.json'. Ejecute el comando de restauración para continuar.

Haga clic en "Cerrar" llegaremos a esto pronto.

4. Agregue las dependencias, abra "project.json" y sobrescríbalo con lo siguiente:

```
{
  "version": "1.0.0-*",
  "buildOptions": {
    "debugType": "portable",
    "emitEntryPoint": true
  },

  "frameworks": {
    "netcoreapp1.1": {
      "dependencies": {
        "Microsoft.AspNetCore.Hosting": "1.1.0",
        "Microsoft.AspNetCore.Server.Kestrel": "1.1.0",
        "Microsoft.AspNetCore.Owin": "1.1.0",
        "Nancy": "2.0.0-barneyrubble",
        "Microsoft.NETCore.App": {
          "type": "platform",
          "version": "1.1.0"
        }
      }
    }
  }
}
```

```
}
```

Código VS le pedirá a restaurar, haga clic en "Restaurar"

5. Crear carpeta "Módulos" en proyecto VSCode

En la carpeta Módulos, agregue un archivo llamado "IndexModule.cs", luego copie y guarde lo siguiente:

```
namespace NancyProject1
{
    using Nancy;
    public class IndexModule : NancyModule
    {
        public IndexModule()
        {
            Get("/", _ => "Hello dotnet core world!");
        }
    }
}
```

6. En el directorio raíz del proyecto, cree un archivo llamado "Startup.cs" y copie y pegue lo siguiente:

```
namespace NancyProject1
{
    using Microsoft.AspNetCore.Builder;
    using Nancy.Owin;

    public class Startup
    {
        public void Configure(IApplicationBuilder app)
        {
            app.UseOwin(x => x.UseNancy());
        }
    }
}
```

7. Abra el archivo "Program.cs" y sobrescriba el contenido con lo siguiente y guarde:

```
namespace NancyProject1
{
    using System.IO;
    using Microsoft.AspNetCore.Builder;
    using Microsoft.AspNetCore.Hosting;

    public class Program
    {
        public static void Main(string[] args)
        {
            var host = new WebHostBuilder()
                .UseContentRoot(Directory.GetCurrentDirectory())
                .UseKestrel()
                .UseStartup<Startup>()
                .Build();
        }
    }
}
```

```
        host.Run();  
    }  
}
```

8. ¡Hecho! Ahora vamos a ejecutar esto y ver la salida.

Haga clic en el símbolo de depuración en el Código VS y haga clic en el botón Ejecutar. Se debe compilar e iniciar el proyecto.

Abra el navegador @ <http://localhost:5000>

9. Dése una palmada en la espalda y disfrute!

Lea Empezando con Nancy en línea: <https://riptutorial.com/es/nancy/topic/6049/empezando-con-nancy>

Capítulo 2: Estado de soporte de .net core

Introducción

Este tema muestra el estado más reciente de la compatibilidad de NancyFx con .Net Core. El apoyo de la comunidad es muy bienvenido para actualizar esto.

Examples

Estado

Actualmente existen numerosas incompatibilidades con el núcleo .Net. Esto mostrará el estado del estado del paquete NancyFx. Esto se agrupa por la funcionalidad. Actualmente el soporte es para Nancy 2. * que aún no ha lanzado una versión estable.

Núcleo de Nancy

Nancy: 2.0.0-clinteastwood

Nancy Bootstrappers

Nancy.Bootstrappers.Ninject - No soportado

Nancy ver los motores

Nancy.ViewEngines.Razor - Problema no admitido # [2521](#)

Autenticación Nancy

Nancy.Authentication.Stateless: 2.0.0-clinteastwood

Nancy FluentValidation

Nota: Debe usar al menos el paquete "" FluentValidation.AspNetCore: 6.4.0-beta9

Nancy.Validation.FluentValidation: 2.0.0-clinteastwood

Cambios de enrutamiento

Ya no es compatible:

```
Get ["/"] = parameters => {  
    return View["index"];  
};
```

Cambiado a:

```
Get("/", parameters => {  
    return View["index"];  
});
```

Lea Estado de soporte de .net core en línea: <https://riptutorial.com/es/nancy/topic/8591/estado-de-soporte-de--net-core>

Creditos

S. No	Capítulos	Contributors
1	Empezando con Nancy	4444 , Allistaire Clark , Community , Fritz Seitz , Komeisa
2	Estado de soporte de .net core	Fritz Seitz