

 eBook Gratuit

# APPRENEZ

---

## nancy

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#nancy

# Table des matières

À propos.....	1
<b>Chapitre 1: Commencer avec nancy.....</b>	<b>2</b>
Remarques.....	2
Sur:.....	2
Ressources additionnelles:.....	2
Versions.....	2
Exemples.....	2
Créer une application Nancy auto-hébergée simple.....	2
Installez Nancyfx avec Dotnet Core v1.1, Kestrel et Visual Studio Code sur les systèmes * .....	3
Étapes prérequis:.....	3
Créer un projet NancyFx auto-hébergé:.....	3
<b>Chapitre 2: Statut du support de base .net.....</b>	<b>7</b>
Introduction.....	7
Exemples.....	7
Statut.....	7
<b>Noyau de Nancy.....</b>	<b>7</b>
<b>Nancy Boostrappers.....</b>	<b>7</b>
<b>Nancy View Engines.....</b>	<b>7</b>
<b>Authentification Nancy.....</b>	<b>7</b>
<b>Nancy FluentValidation.....</b>	<b>7</b>
Changements de routage.....	7
<b>Crédits.....</b>	<b>9</b>

---

# À propos

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](mailto:info@zzzprojects.com)

---

# Chapitre 1: Commencer avec nancy

## Remarques

### Sur:

Nancy est un framework léger pour la création de services basés sur HTTP dans .Net basé sur le framework Sinatra existant pour Ruby. Il est conçu pour permettre le traitement de plusieurs types de requêtes HTTP et fournit un moyen simple de retourner une réponse avec une petite quantité de code.

## Ressources additionnelles:

- Site principal: <http://nancyfx.org>
- Documentation: <https://github.com/NancyFx/Nancy/wiki>
- Code source: <https://github.com/NancyFx/Nancy>
- Paquet Nuget: <https://www.nuget.org/packages/Nancy/>

## Versions

Version	Date de sortie
<a href="#">Nancy 1.4.3</a>	2015-12-21
<a href="#">Nancy 1.4.2</a>	2015-11-23
<a href="#">Nancy 1.4.1</a>	2015-11-05
<a href="#">Nancy 1.4.0</a>	2015-10-29
<a href="#">Nancy 1.3.0</a>	2015-09-25
<a href="#">Nancy 1.2.0</a>	2015-04-17
<a href="#">Nancy 1.1.0</a>	2015-02-18
<a href="#">Nancy 1.0.0</a>	2015-01-23

## Exemples

### Créer une application Nancy auto-hébergée simple

1. Utilisez Nuget pour installer les packages Nancy et Nancy.Hosting.Self dans le projet.
2. Instancier un nouvel objet NancyHost et passer l'URL appropriée

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

*Placez ce code dans votre projet au moment où vous souhaitez commencer à écouter le trafic HTTP.*

3. Ajoutez une classe à votre projet qui hérite de NancyModule et ajoutez une méthode constructeur.

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

4. Définir des routes dans le constructeur:

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

## Installez Nancyfx avec Dotnet Core v1.1, Kestrel et Visual Studio Code sur les systèmes \* nix

### Étapes prérequis:

1. Obtenez le noyau de dotnet pour votre plate-forme:

[Noyau Dotnet](#)

2. Suivez les instructions et assurez-vous que le cœur de dotnet fonctionne
3. Obtenez le code Visual Studio pour votre plate-forme:

[Code VS](#)

4. Lancez le code Visual Studio (code VS) et installez l'extension C # puis rechargez

### Créer un projet NancyFx auto-hébergé:

1. Configurez un projet avec une structure de répertoire de projet correcte.

Open Bash Terminal et tapez:

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

2. Ouvrez global.json et entrez le code suivant:

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

3. Dans le terminal Bash:

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

Ouvrir le dossier NancyProject1 dans le code VS

Vous recevrez un avertissement: "Les éléments requis pour la génération et le débogage sont absents de 'nancyproject1'."

Cliquez sur "Oui"

Vous verrez également: Il existe des dépendances non résolues de 'project.json'. Veuillez exécuter la commande de restauration pour continuer.

Cliquez sur "Fermer", nous y reviendrons bientôt.

4. Ajoutez les dépendances, ouvrez "project.json" et remplacez-le par ce qui suit:

```
{
  "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"
        }
      }
    }
  }
}
```

```
}
```

Le code VS vous demandera de restaurer cliquez sur "Restaurer"

## 5. Créer un dossier "Modules" dans le projet VSCode

Dans le dossier Modules, ajoutez un fichier nommé "IndexModule.cs", puis copiez et enregistrez les éléments suivants:

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

## 6. Dans le répertoire racine du projet, créez un fichier appelé "Startup.cs" et copiez et collez les éléments suivants:

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

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

## 7. Ouvrez le fichier "Program.cs" et écrasez le contenu avec les éléments suivants et enregistrez:

```
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. Terminé! Maintenant, allons à ceci et voir la sortie.

Cliquez sur le symbole de débogage dans le code VS et cliquez sur le bouton Exécuter. Il devrait compiler et démarrer le projet.

Ouvrez le navigateur @ <http://localhost:5000>

9. Pat vous-même sur le dos et profitez-en!

Lire Commencer avec nancy en ligne: <https://riptutorial.com/fr/nancy/topic/6049/commencer-avec-nancy>



---

# Chapitre 2: Statut du support de base .net

## Introduction

Cette rubrique présente le dernier état de la compatibilité NancyFx avec .Net Core. Le support communautaire est le bienvenu pour mettre à jour cela.

## Exemples

### Statut

Actuellement, il existe de nombreuses incompatibilités avec le noyau .Net. Cela indiquera l'état du package NancyFx. Ceci est regroupé par fonctionnalité. Le support actuel est pour Nancy 2. \* qui n'a pas encore publié de version stable.

---

## Noyau de Nancy

Nancy: 2.0.0-clinteastwood

---

## Nancy Bootstrappers

Nancy.Bootstrappers.Ninject - Non pris en charge

---

## Nancy View Engines

Nancy.ViewEngines.Razor - Problème non pris en charge [# 2521](#)

---

## Authentification Nancy

Nancy.Authentication.Stateless: 2.0.0-clinteastwood

---

## Nancy FluentValidation

Remarque: Doit utiliser au moins le package "" FluentValidation.AspNetCore: 6.4.0-beta9

Nancy.Validation.FluentValidation: 2.0.0-clinteastwood

### Changements de routage

N'est plus pris en charge:

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

Changé en:

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

Lire Statut du support de base .net en ligne: <https://riptutorial.com/fr/nancy/topic/8591/statut-du-support-de-base--net>

# Crédits

S. No	Chapitres	Contributeurs
1	Commencer avec nancy	<a href="#">4444</a> , <a href="#">Allistaire Clark</a> , <a href="#">Community</a> , <a href="#">Fritz Seitz</a> , <a href="#">Komeisa</a>
2	Statut du support de base .net	<a href="#">Fritz Seitz</a>