

عنوان: راه اندازی وب سایت سریع و سبک با Nancy

نویسنده: سلمان عرب عامری

تاریخ: ۷:۴۰ ۱۳۹۱/۰۴/۱۱

آدرس: www.dotnettips.info

برچسب‌ها: ASP.Net, Nancy, Sinatra, HTTP

Nancy یک فریم ورک سبک برای ساخت سرویس‌های مبتنی بر HTTP بر روی .Net و Mono و در واقع پیاده سازی Sinatra در Ruby برای .net است. با استفاده از این کتابخانه شما می‌توانید به سادگی درخواست‌های مختلف کاربران را از طریق وب پاسخ دهید. از ویژگی‌های این کتابخانه امکان اجرای آن بدون نیاز به وجود وب سرور و به صورت Standalone می‌باشد. بهتر است وارد عمل شویم و ببینیم این سیستم چگونه عمل می‌کند. برای شروع ما خود Asp .net را به عنوان میزبان در نظر می‌گیریم.

- یک پروژه خالی Asp .net ایجاد کنید. (Asp .net Empty Web Application)

- وارد خط فرمان Package Manager شوید و عبارت زیر را وارد کنید:

```
Install-Package Nancy.Hosting.Aspnet
```

- پروژه‌های Nancy از تعدادی ماژول تشکیل شده اند که می‌توانیم برای هر ماژول آدرس وب جداگانه ای در نظر بگیریم. یک کلاس به پروژه اضافه کنید و نام آن را TestModule بگذارید. در متن کلاس عبارات زیر را تایپ کنید:

```
public class TestModule : NancyModule
{
    public TestModule()
    {
        Get["/" ] = x => "Welcome to my site!";
        Get["/Hello/" ] = x=> "Hello Nancy!";
        Get["/Bye/{name}"] = x=> "Good bye " + x.name;
    }
}
```

حال کلید F5 را زده و برنامه را اجرا کنید.

حالا در مرورگر خودتان عبارت `http://localhost:12345/Hello` را تایپ کنید. توجه کنید که به جای 12345 باید شماره پورتی که وب سرور دات نت اجراست تایپ کنید.

همانطور که متوجه شدید ما در خطوط بالا تعیین کرده ایم که برای درخواست‌های از نوع Get که مسیر ریشه سایت را درخواست می‌کنند عبارت `Welcome to my site!` ارسال شود. همچنین برای درخواست‌هایی که مسیر `Hello/` را درخواست می‌کنند عبارت `Hello Nancy` نمایان می‌شود.

نکته جالب برای درخواست‌های `Bye/` است. در اینجا ما یک پارامتر به نام `name` تعریف کرده ایم و گفته ایم که درخواست‌های `Bye` که در ادامه آنها عبارتی وجود دارد به صورت `Good bye` به همراه عبارت بازگردانده شوند. همین عملیات برای درخواست‌های `Put` و `Head` و سایر انواع درخواست وجود دارد.

برای اینکه درخواست‌ها از مسیر خاصی فراخوانی شوند کافی است در هنگام اعلان سازنده کلاس ماژول، مسیر را تعیین کنید.

```
public class TestModule : NancyModule
{
    public TestModule() : base("/test")
    {
        Get["/user"] = x=> "It is test for user";
    }
}
```

این درخواست‌ها از مسیری شبیه این مسیر فراخوانی خواهند شد:

```
http://localhost:12345/test/user
```

همانطور که برای پردازش درخواست از عبارات کوتاه استفاده کردیم می‌توانیم از توابع هم استفاده کنیم.

```
Get["/hello/{username}"] = x=> {  
    // some code  
    // ...  
    // ...  
    return "Hello, " + username;  
};
```

همچنین امکان بازگرداندن کدهای وضعیت به طور مستقیم وجود دارد.

```
Get["/user"] = x=> return 200;
```

و یا

```
Get["/user"] = x=> return HttpStatusCode.OK;
```

همانطور که گفته شد امکان میزبانی پروژه‌های Nancy از داخل برنامه‌های دات نتی هم وجود دارد. در مقاله بعدی به این موضوع خواهیم پرداخت.

نظرات خوانندگان

نویسنده:

بهمن

تاریخ:

۱۳۹۱/۰۴/۱۱ ۸:۴۴

سلام ممنون از اینکه این مطلب را ارسال کردید فقط یه سوالی داشتم اونم اینکه آیا در بیشتر سایتها که آدرس url آنها بدین شکل هست چه asp و چه php از این روش برای خلاصه کردن آدرس سایت استفاده میکنند ؟....

نویسنده:

وحید نصیری

تاریخ:

۱۳۹۱/۰۴/۱۱ ۱۰:۴۹

این تکنیک اصطلاحاً [Url Rewriting](#) نام دارد.

نویسنده:

سلیمان عرب عامری

تاریخ:

۱۳۹۱/۰۴/۱۱ ۱۲:۲۴

همانطور که جناب نصیری گفتند، به این روش که آدرسها را کاربر پسند می‌کنیم اصطلاحاً Url rewriting می‌گویند. این نوع آدرس دهی باعث خوانا شدن، و همچنین بهتر ایندکس شدن در موتورهای جستجو می‌شود. در دات نت پروژه MVC آدرسها را طبق این اصول پیاده می‌کند و کاربر می‌تواند قاعده آدرس دهی خود را به صورت یک Route تعریف کند. در Asp .net سنتی هم کتابخانه‌های Url Rewriting وجود دارند که برای شما این کار را انجام می‌دهند. ضمن اینکه خود شما هم با نوشتن HttpModule می‌توانید این کار را انجام دهید.