عنوان: **هدایت خودکار آدرس‌های یافت نشد در یک سایت ASP.NET MVC به جستجوی سایت**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۲ ۱۲:۱۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Dependency Injection

هر از چندگاهی یک چنین آدرس‌های یافت نشدی را در لاگ‌های سایت مشاهده می‌کنم:

```
http://www.dotnettips.info/jquery
http://www.dotnettips.info/mvc
http://www.dotnettips.info/برنامه
```

روش متداول مدیریت این نوع آدرس‌ها، هدایت خودکار به صفحه‌ی 404 است. اما شاید بهتر باشد بجای اینکار، کاربران به صورت خودکار به صفحه‌ی جستجوی سایت هدایت شوند. در ادامه مراحل اینکار را بررسی خواهیم کرد.

الف) ساختار کنترلر جستجوی سایت

فرض کنید جستجوی سایت در کنترلری به نام Search و توسط اکشن متد پیش فرضی با فرمت زیر مدیریت می‌شود:

```
[ValidateInput(false)] //برنامه نویس‌ها نیاز دارند تگ‌ها را جستجو کنند
  public virtual ActionResult Index(string term)
  {
```

ب) مدیریت کنترلرهای یافت نشد

اگر از یک IoC Container در برنامه‌ی ASP.NET MVC خود مانند StructureMap استفاده می‌کنید ، نوشتن کد متداول زیر کافی نیست:

```
public class StructureMapControllerFactory : DefaultControllerFactory
    {
        protected override IController GetControllerInstance(RequestContext requestContext, Type
controllerType)
        {
            return ObjectFactory.GetInstance(controllerType) as Controller;
        }
    }
```

از این جهت که اگر کاربر آدرس http://www.dotnettips.info/test را وارد کند، controllerType درخواستی نال خواهد بود؛ چون جزو کنترلرهای سایت نیست. به همین جهت نیاز است موارد نال را هم مدیریت کرد:

```
public class StructureMapControllerFactory : DefaultControllerFactory
    {
        protected override IController GetControllerInstance(RequestContext requestContext, Type
controllerType)
        {
            if (controllerType == null)
            {
                var url = requestContext.HttpContext.Request.RawUrl;
                //string.Format("Page not found: {0}", url).LogException();

                requestContext.RouteData.Values["controller"] = MVC.Search.Name;
                requestContext.RouteData.Values["action"] = MVC.Search.ActionNames.Index;
                requestContext.RouteData.Values["term"] = url.GetPostSlug().Replace("-", " ");
                return ObjectFactory.GetInstance(typeof(SearchController)) as Controller;
            }
            return ObjectFactory.GetInstance(controllerType) as Controller;
        }
    }
```

کاری که در اینجا انجام شده، هدایت خودکار کلیه کنترلرهای یافت نشد برنامه، به کنترلر Search است. اما در این بین نیاز است سه مورد را نیز اصلاح کرد. در RouteData.Values جاری، نام کنترلر باید به نام کنترلر Search تغییر کند. زیرا مقدار پیش فرض آن، همان عبارتی است که کاربر وارد کرده. همچنین باید مقدار action را نیز اصلاح کرد، چون اگر آدرس وارد شده برای مثال http://www.dotnettips.info/mvc/test بود، مقدار پیش فرض action همان test می‌باشد. بنابراین صرف بازگشت وهله‌ای از

SearchController تمام موارد را پوشش نمی‌دهد و نیاز است دقیقا جزئیات سیستم مسیریابی نیز اصلاح شوند. همچنین پارامتر term اکشن متد index را هم در اینجا می‌شود مقدار دهی کرد. برای مثال در اینجا عبارت وارد شده اندکی تمیز شده (مطابق روش متد تولید Slug) و سپس به عنوان مقدار term تنظیم می‌شود.

ج) مدیریت آدرس‌های یافت نشد پسوند دار

تنظیمات فوق کلیه آدرس‌های بدون پسوند را مدیریت می‌کند. اما اگر درخواست رسیده به شکل http://www.dotnettips.info/mvc/test/file.aspx بود، خیر. در اینجا حداقل سه مرحله را باید جهت مدیریت و هدایت خودکار آن به صفحه‌ی جستجو انجام داد

- باید فایل‌های پسوند دار را وارد سیستم مسیریابی کرد :

```
routes.RouteExistingFiles = true; //نیاز هست دانلود عمومی فایل‌ها تحت کنترل قرار گیرد
```

- در ادامه نیاز است مسیریابی Catch all اضافه شود:

پس از مسیریابی پیش فرض سایت (نه قبل از آن)، مسیریابی ذیل باید اضافه شود:

```
routes.MapRoute(
                "CatchAllRoute", // Route name
                "{*url}", // URL with parameters
                new { controller = "Search", action = "Index", term = UrlParameter.Optional, area = ""
}, // Parameter defaults
                new { term = new UrlConstraint() }
            );
```

مسیریابی پیش فرض، تمام آدرس‌های سازگار با ساختار MVC را می‌تواند مدیریت کند. فقط حالتی از آن عبور می‌کند که پسوند داشته باشد. با قرار دادن این مسیریابی جدید پس از آن، کلیه آدرس‌های مدیریت نشده به کنترلر Search و اکشن متد Index آن هدایت می‌شوند.

مشکل! نیاز است پارامتر term را به صورت پویا مقدار دهی کنیم. برای اینکار می‌توان یک RouteConstraint سفارشی نوشت:

```
public class UrlConstraint : IRouteConstraint
    {
        public bool Match(System.Web.HttpContextBase httpContext,
                          Route route, string parameterName,
                          RouteValueDictionary values,
                          RouteDirection routeDirection)
        {
            var url = httpContext.Request.RawUrl;
            //string.Format("Page not found: {0}", url).LogException();

            values["term"] = url.GetPostSlug().Replace("-", " ");
            return true;
        }
    }
```

UrlConstraint مطابق تنظیم CatchAllRoute فقط زمانی فراخوانی خواهد شد که برنامه به این مسیریابی خاص برسد (و نه در سایر حالات متداول کار با کنترلر جستجو). در اینجا فرصت خواهیم داشت تا مقدار term را به RouteValueDictionary آن اضافه کنیم.