

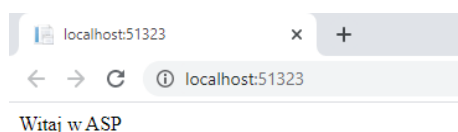
Państwowa Wyższa Szkoła Zawodowa w Nowym Sączu Programowanie aplikacji WEB - laboratorium			
Temat: Routing w ASP.NET MVC.			Symbol: Rt
Nazwisko i imię: Oleś Adrian		Ocena sprawozdania	Zaliczenie:
Data wykonania ćwiczenia: 30.03.2020	Grupa: L2		

Ćwiczenie.

1. Utwórz pusty projekt w ASP.NET MVC. Nazwa projektu: Routing.
2. Utwórz pusty kontroler zapisz pod nazwą HomeController.
3. Zmodyfikuj metodę Index()

```
public string Index()
{
    return "Witaj w ASP.NET MVC";
}
```

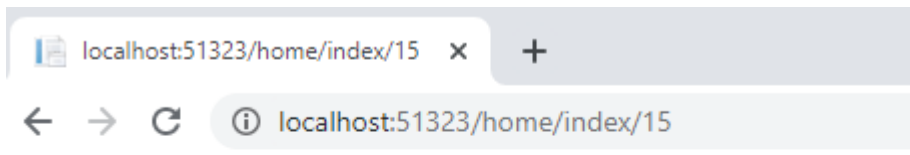
4. Uruchom aplikację. Wklej zrzut ekranu.



5. Ponownie zmodyfikuj metodę Index()

```
public string Index()
{
    string wynik = "";
    foreach (string routekey in RouteData.Values.Keys)
    {
        wynik += "Key: " + routekey + " Value: " + RouteData.Values[routekey] as string;
        wynik += "<br />";
    }
    return wynik;
}
```

6. Uruchom aplikację. W przeglądarce wpisz adres:
<http://localhost:xxxx/Home/Index/15>



Key: controller Value: home
 Key: action Value: index
 Key: id Value: 15

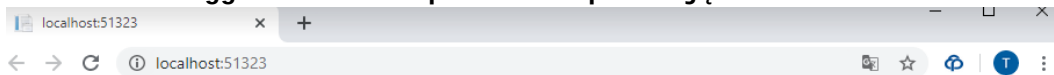
7. Generowanie adresu URL - metoda wstecznego Routingu. Zmodyfikuj metodę Index()

```
public string Index()
{
    var r = RouteTable.Routes.GetVirtualPath(null, new RouteValueDictionary
    {
        new { controller = "Kontroler", action = "Akcja", id = "12", parametr = "Adam" }
    });

    string s = r.VirtualPath;

    return s;
}
```

8. Zainstaluj Route Debugger. Kliknij PPM na nazwie projektu „Routing”, wybierz „Menage NuGet Packages”, wyszukaj i zainstaluj Route Debugger. Uruchom ponownie aplikację.



Key: controller Value: Home
 Key: action Value: Index

Route Debugger

Type in a url in the address bar to see which defined routes match it. A {catchall} route is added to the list of routes automatically in case none of your routes match.

To generate URLs using routing, supply route values via the query string. example: <http://localhost:14290/?id=123>

Matched Route: {controller}/{action}/{id}

Route Data		Data Tokens	
Key	Value	Key	Value
controller	Home		
action	Index		

All Routes

Matches Current Request	Url	Defaults	Constraints	DataTokens
False	{resource}.axd/{*pathInfo}	(null)	(empty)	(null)
True	{controller}/{action}/{id}	controller = Home, action = Index, id = UrlParameter.Optional	(empty)	(empty)
True	{*catchall}	(null)	(null)	(null)

Current Request Info

AppRelativeCurrentExecutionFilePath is the portion of the request that Routing acts on.

AppRelativeCurrentExecutionFilePath: ~/

9. Zmodyfikuj plik RouteConfig.sc

```
namespace Routing
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "ProductDetails",
                url: "album-{id}",
                defaults: new { controller = "Store", action = "Details" }
            );

            routes.MapRoute(
                name: "ProductList",
                url: "gatunki/{genrename}",
                defaults: new { controller = "Store", action = "List" },
                constraints: new { genrename = @"[\w& ]+" }
            );

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
            );
        }
    }
}
```

10. Definiowanie tras z wykorzystaniem atrybutów. Utwórz pusty kontroler zapisz pod nazwą StoreController. Uzupełnij nową trasą routingu.

```
public class StoreController : Controller
{
    [Route("to/jest/{moja}/trasa")]
    public string Index(string moja)
    {
        return "Atrybuty routingu";
    }
}
```

Zadanie:

Zaprojektuj tablicę routingu w nowo utworzonej, pustej aplikacji ASP.NET MVC tak, aby spełniała następujące wymagania:

- Domyślnie, pierwszy człon adresu URL powinien oznaczać nazwę kontrolera a drugi człon nazwę akcji (np. <http://twojadomena.com/products/list> ma prowadzić do kontrolera ProductsController i jego akcji List). Jeśli parametry nie zostaną wprowadzone, domyślnie trasa powinna prowadzić do kontrolera StaticController i jego akcji Index.

- **Adresy, który drugich człon to "gatunki" - takie jak <http://twojadomena.com/gatunki/rock>, <http://twojadomena.com/gatunki/pop> powinny prowadzić do kontrolera ProductsController i jego akcji Index. Ostatni człon adresu (np. rock, pop) powinien być przekazywany jako parametr string metody Index.**

- **Adres, którego pierwszy człon to "onas" (np. <http://twojadomena.com/onas>) powinien kierować do kontrolera StaticController i jego akcji About.**

1

Rozwiązanie zadania:

Plik. RouteConfig.cs

```
namespace RoutingTask
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "About",
                url: "onas",
                defaults: new { controller = "Static", action = "About" }
            );

            routes.MapRoute(
                name: "Genres",
                url: "gatunki/{musicgenre}",
                defaults: new { controller = "Products", action = "Index" }
            );

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Static", action = "Index", id = UrlParameter.Optional }
            );
        }
    }
}
```

2.

Plik. ProductsController.cs

```
namespace RoutingTask.Controllers
{
    public class ProductsController : Controller
    {
        // GET: Products
        public string Index(string musicgenre)
        {
            return "Trafiłem do Index, gatunek: " + musicgenre;
        }

        public string SampleAction()
        {
            return "Przykładowa akcja w kontrolerze Products";
        }
    }
}
```

3.

Plik StaticController.cs

```
namespace RoutingTask.Controllers
{
    public class StaticController : Controller
    {
        public string Index()
        {
            return "Przykładowa, statyczna strona domowa";
        }

        public string About()
        {
            return "Strona o nas";
        }
    }
}
```