

Routing in MVC



Lesson Objectives

- *Routing in MVC*
- *URL Pattern in MVC*
- *Custom Routing*
- *Multiple Routes*
- *Route Constraint*

- URL: Universal Resource Locator
- Each URL should be matched with a specific physical file
- Example:
 - ✓ `http://example.com?id=12`

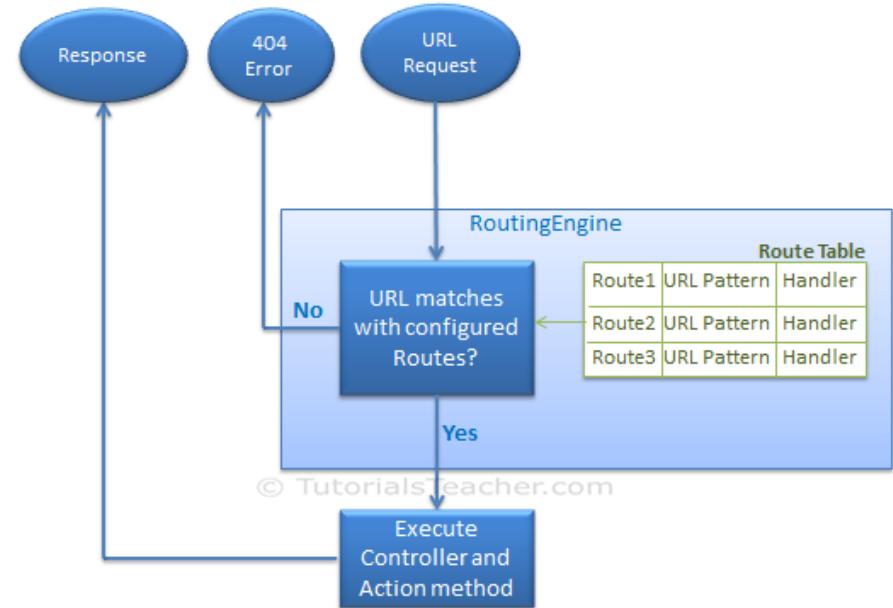
URL disadvantages

- May leak of security information
- Is not friendly to search engines
- Unfriendly to user

- Enable to define URL pattern that maps to the request handler.
- The request handler can be a file or class.
- Routing engine determines appropriate handler class or file for an incoming request.

Route

- Defines URL pattern and handler information.
- Stored in RouteTable
- Will be used by Routing engine



Configure a Route

- using the MapRoute() extension method of RouteCollection class
- Every MVC application must configure (register) at least one route
- Can add multiple routes
- Route name must be unique

Configure a Route

- using the MapRoute() extension method of RouteCollection class
- Every MVC application must configure (register) at least one route
- Can add multiple routes
- Route name must be unique

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

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


- The URL pattern is considered only after domain name part in the URL.
 - ✓ For example, the URL pattern `{controller}/{action}/{id}` would look like `localhost:1234/{controller}/{action}/{id}`.
 - ✓ Anything after "localhost:1234/" would be considered as controller name.
 - ✓ Anything after controller name would be considered as action name and then value of id parameter.

- If any segment is lack, MVC use default value in **defaults** definitions

URL	Controller	Action	Id
http://localhost/home	HomeController	Index	null
http://localhost/home/index/123	HomeController	Index	123
http://localhost/home/about	HomeController	About	null
http://localhost/home/contact	HomeController	Contact	null
http://localhost/student	StudentController	Index	null
http://localhost/student/edit/123	StudentController	Edit	123

- You can configure a custom route using MapRoute extension method.
- You need to provide at least two parameters in MapRoute, route name and URL pattern.

- The Defaults parameter is optional.
- You can register multiple custom routes with different names.
- MVC framework evaluates each route in sequence.

- It starts with first configured route and if incoming URL doesn't satisfy the URL pattern of the route then it will evaluate second route and so on.
- Therefor:
 - ✓ Try to put routes in right order
 - ✓ Specific route should be first, general route should be last

- To apply restrictions on the value of parameter by configuring route constraints
- Use a set of expressions to specify values for the URL parameter.

- Example:
 - ✓ `id= @"\d+"` means: id is digits only
 - ✓ `code= "\w{4}"` means: code is 4 characters or digits only
- If incoming URL doesn't satisfy the constraints of the route then it will evaluate next route and so on.

- By default, router is registered in the Application_Start() event in the Global.asax

0 references

```
public class MvcApplication : System.Web.HttpApplication
```

```
{
```

0 references

```
protected void Application_Start()
```

```
{
```

```
    AreaRegistration.RegisterAllAreas();
```

```
    FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
```

```
    RouteConfig.RegisterRoutes(RouteTable.Routes);
```

```
    BundleConfig.RegisterBundles(BundleTable.Bundles);
```

```
}
```

```
}
```


Thank you

