



# The Controller

Authored by : Sushant

Presented by : Sushant

# Agenda

- Introduction to Controller
- Routing
- MVC Request Processing
- Controller Actions
- Action Results
- Action Selectors
- Action Filters



# What is Controller

- A class responsible for
  - Accepting requests from client
  - Calling appropriate action methods
  - Interacting with models when required
  - Deciding which view to return
- All controllers inherits
  - Controller : ControllerBase : IController

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
}
```

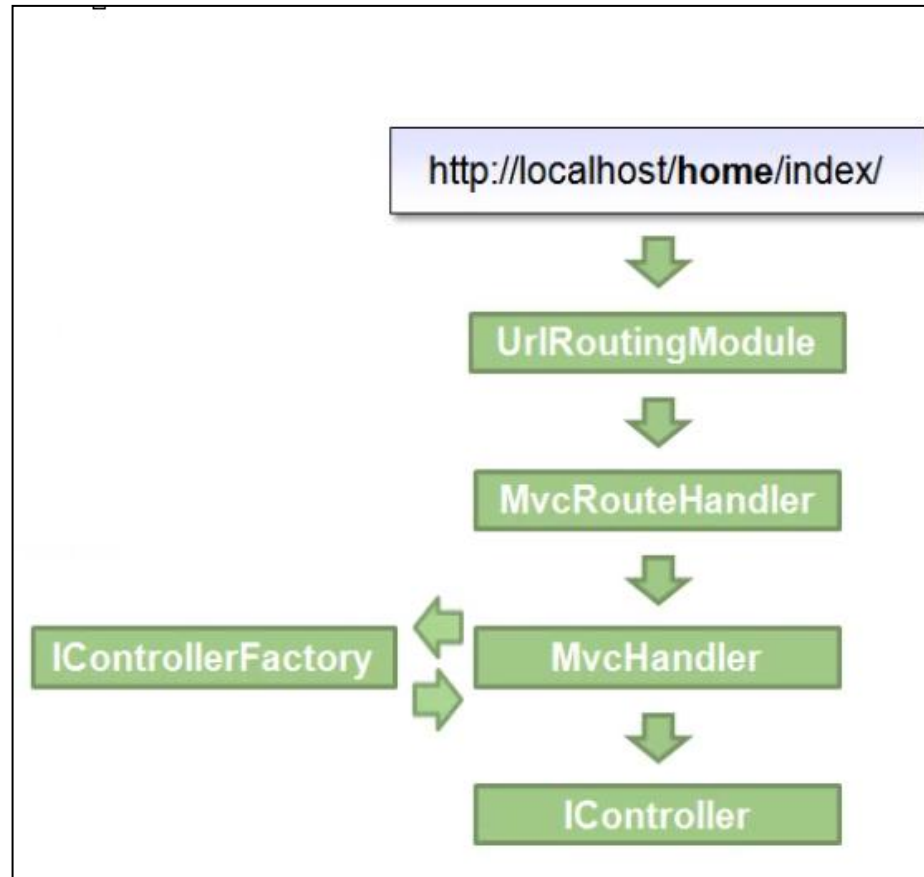
# ASP.NET Routing

- Get route info from URL
- Reads Route Table
- First match wins
- Creates a data structure
- Passes it to MVC framework

```
public class RouteConfig
{
    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 }
        );
    }
}
```

# MVC Request Processing



# Default Controller Factory

- Finds Controllers
  - Looks in all referenced assemblies
  - Looks in all namespaces
  - Type name ends with "Controller"
  - Type implements IController
- Instantiates controllers
  - Default constructor is must

# Controller Actions

- Actions are methods in a controller class
- An action method must be public
- Can be invoked by
  - Entering a URL
  - Clicking a link
  - Submitting a form
- An action method cannot be
  - Overloaded
  - Static
- Action method parameters
  - Primitive types such as int
  - Complex types such as object

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
}
```

## Action Results

Action Result	Helper Method	Description
ViewResult	View	Renders a view as a Web page.
PartialViewResult	PartialView	Renders a partial view, which defines a section of a view that can be rendered inside another view.
RedirectResult	Redirect	Redirects to another action method by using its URL.
RedirectToRouteResult	RedirectToAction RedirectToRoute	Redirects to another action method.
ContentResult	Content	Returns a user-defined content type.
JsonResult	Json	Returns a serialized JSON object.
JavaScriptResult	JavaScript	Returns a script that can be executed on the client.
FileResult	File	Returns binary output to write to the response.
EmptyResult	(None)	Represents a return value that is used if the action method must return a null result (void).



# Action Selectors

- Action Name
  - Create an alias for action method
  - Routing considers alias instead of method name
- Accept Verbs
  - Used to mark methods for specific HTTP request
  - Maps action methods to HTTP methods

```
[ActionName("MyDetails")]  
[AcceptVerbs(HttpVerbs.Get)]  
public ActionResult Details(int id)  
{  
    ViewData["id"] = id;  
    return View();  
}
```

# Action Filters

- Can be applied on
  - Action Methods
  - Controller Class
  - Global filters – applied to entire application
- Filters are basically attributes
  - Authorize
  - HandleError
  - OutputCache
  - ChildActionOnly
  - ValidateAntiForgeryToken

## Bibliography, Important Links

- <http://www.asp.net/mvc/tutorials/controllers-and-routing>
- [http://msdn.microsoft.com/en-in/library/dd410269\(v=vs.100\).aspx](http://msdn.microsoft.com/en-in/library/dd410269(v=vs.100).aspx)

# Any Questions?





Thank you!