



# Controllers

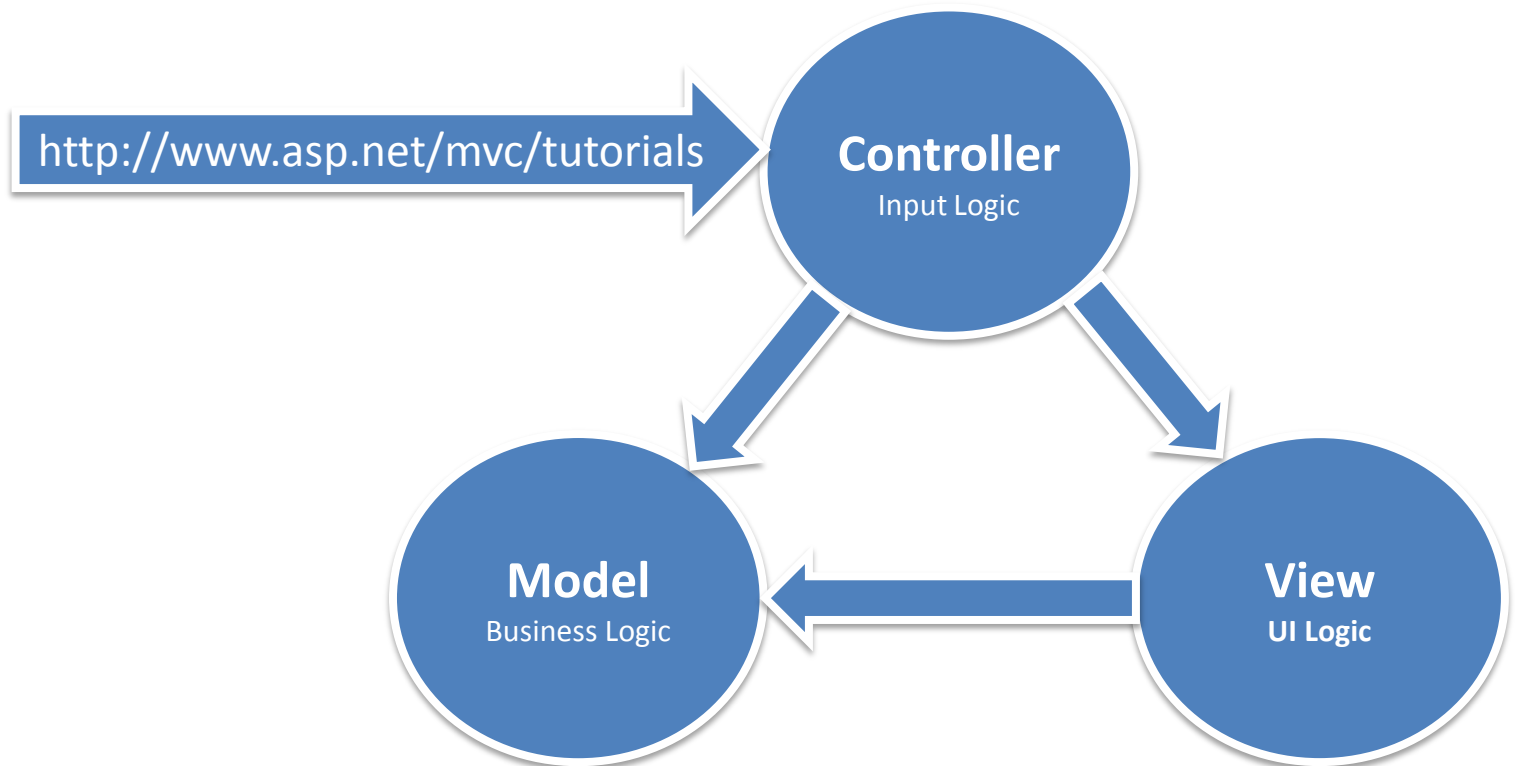
Author and Presenter : Sushant B.

# Agenda

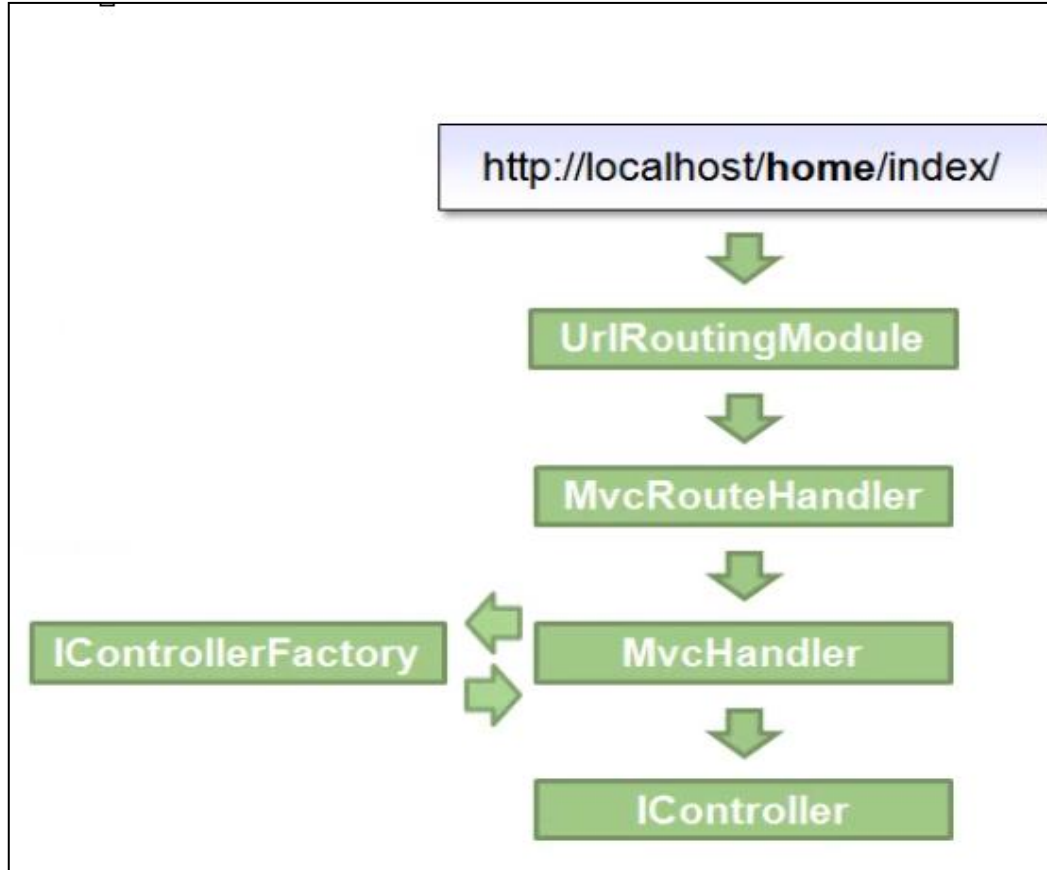
- Introduction
- MVC Request Processing
- Controller Actions
- Routing
- Action Results
- Action Selectors
- Action Filters



## MVC – A Design Pattern...



# MVC Request Processing



- Read more [here](#) and [here](#).

# What is Controller

- Accepts requests from client
- Calls appropriate action methods
- Decides which model to pass
- Decides which view to call
- Inherits Controller : ControllerBase : IController

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

## How Controllers Are Found

- Default Controller Factory finds controllers
- Looks in all referenced assemblies
- Looks in all namespaces
- Type name ends with "Controller"
- Type implements IController
- Creates controllers instance - default constructor is must

# Controller Actions

- Actions are methods in a controller class
  - Call by entering a URL
  - Call by clicking a link
  - Call by submitting a form
- An action method
  - Must be public
  - Can't Overload
  - Can not be Static
- Action Parameters
  - Can be primitive types such as int
  - Can be complex types such as object

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

# Demo



# 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 }
        );
    }
}
```

# Demo

# 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).

# Demo

# 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();  
}
```

# Demo

# 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

# Demo



# Summary

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

## Bibliography, Important Links

- <http://www.asp.net/mvc>

# Any Questions?



Email : [SushantBa@cybage.com](mailto:SushantBa@cybage.com)  
Extn. : 7221

Thank you!