

Building Applications with ASP.NET MVC

Controllers

K. Scott Allen



Routing

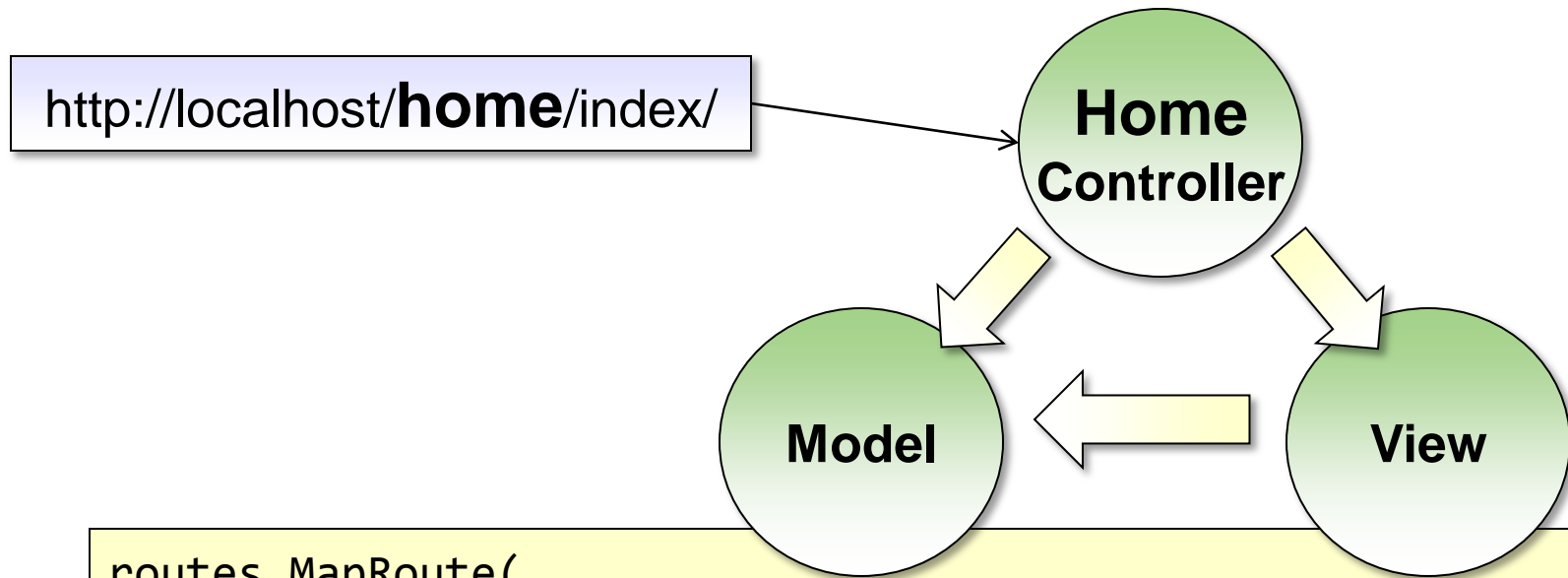
**Action
Results**

**Controller
Actions**

**Action
Filters**

**Action
Parameters**

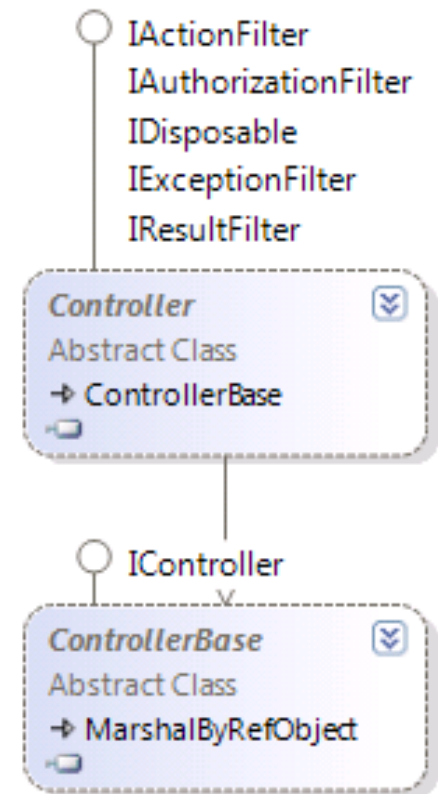
Routes & Controllers



```
routes.MapRoute(  
    "Default",                // Route name  
    "{controller}/{action}/{id}", // URL with parameters  
    new { controller = "Home",  
          action = "Index",  
          id = ""  
    } // Parameter defaults
```

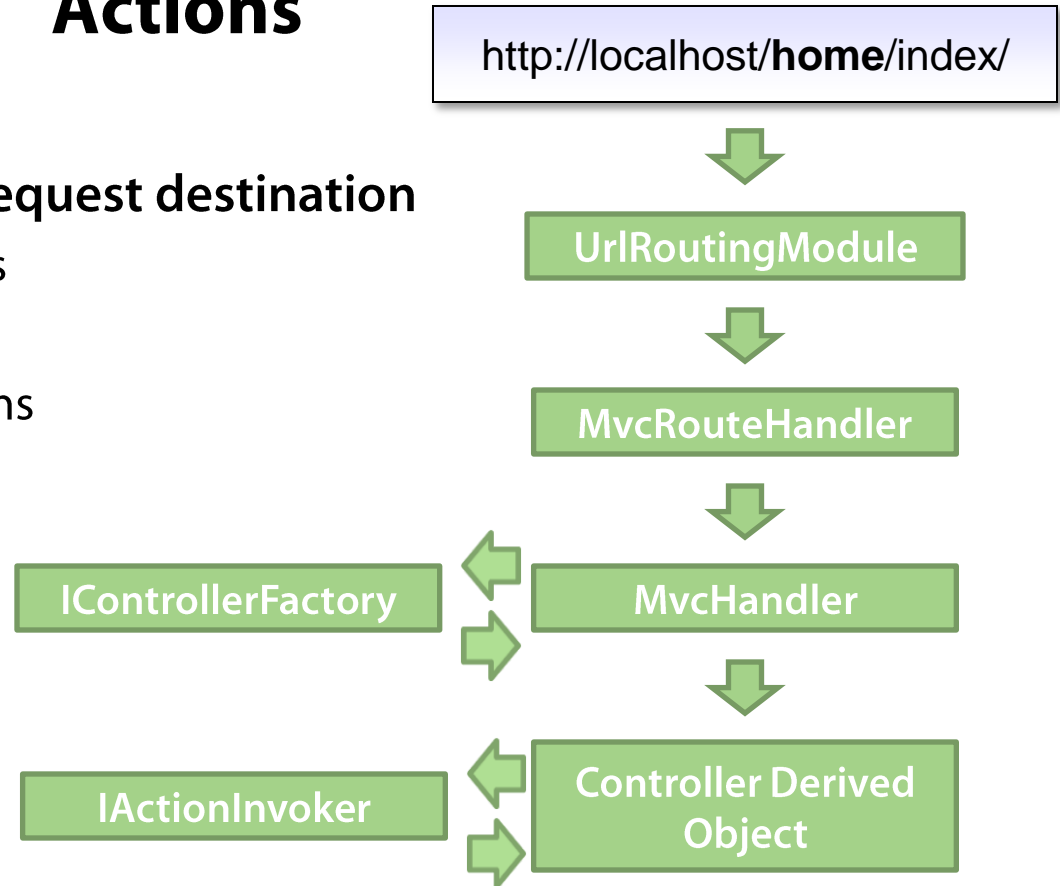
Controller Execution

- **Controller**
 - Execute method invoked by MvcHandler
 - Writes to the response stream
- **ControllerBase**
 - Introduces ViewData and ControllerContext
- **Controller**
 - Default base class in MVC
 - Introduces Actions
 - Includes helper methods for rendering content



Actions

- **Actions are the ultimate request destination**
 - Public controller methods
 - Non-static
 - No return value restrictions



Action Parameters

- **Actions can take a variety of parameters**
 - Primitive parameters (int, bool, string)
 - Complex parameters
- **Model Binder looks for parameter values**
 - In route data
 - In form data
 - In query string

Results

- Actions typically return an ActionResult

Name	Framework Behavior	Producing Method
ContentResult	Returns a string literal	Content
EmptyResult	No response	
FileContentResult / FilePathResult / FileStreamResult	Return the contents of a file	File
HttpUnauthorizedResult	Returns an HTTP 403 status	
JavaScriptResult	Returns a script to execute	JavaScript
JsonResult	Returns data in JSON format	Json
RedirectResult	Redirects the client to a new URL.	Redirect
RedirectToRouteResult	Redirect to another action, or another controller's action	RedirectToRoute / RedirectToAction
ViewResult PartialViewResult	Response is the responsibility of a view engine	View / PartialView

Action Selectors

- **ActionName**
- **AcceptVerbs**
 - HttpPost, HttpGet

```
[ActionName("Modify")]  
[AcceptVerbs(HttpVerbs.Post)]  
public ActionResult Edit(string departmentName)  
{  
    // ...  
}
```


Action Filters

Name	Description
OutputCache	Cache the output of a controller
ValidateInput	Turn off request validation and allow dangerous input
Authorize	Restrict an action to authorized users or roles
ValidateAntiForgeryToken	Helps prevent cross site request forgeries
HandleError	Can specify a view to render in the event of an unhandled exception

Custom Action Filters

- Derive from `ActionFilterAttribute` base class

```
public class LogAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    { /* */ }

    public override void OnActionExecuted(ActionExecutedContext filterContext)
    { /* */ }

    public override void OnResultExecuting(ResultExecutingContext filterContext)
    { /* */ }

    public override void OnResultExecuted(ResultExecutedContext filterContext)
    { /* */ }
}
```

```
[Log]
[HandleError]
public class DepartmentController : Controller
{
    // ...
}
```

Summary

Routing

**Action
Results**

**Controller
Actions**

**Action
Filters**

**Action
Parameters**