ASP.Net MVC

Table of Contents

[Flow 2](#_Toc32612455)

[Routes 3](#_Toc32612456)

[RouteConfig 3](#_Toc32612457)

[Custom Route 3](#_Toc32612458)

[Add Route 3](#_Toc32612459)

[Map to controllers 3](#_Toc32612460)

[Attribute Routes 3](#_Toc32612461)

[Init in RegisterRoutes 3](#_Toc32612462)

[Decorate controller with attribute 3](#_Toc32612463)

[Constraint 3](#_Toc32612464)

[Controllers 5](#_Toc32612465)

[Controller Parameters 5](#_Toc32612466)

[Action Results -> View 5](#_Toc32612467)

[RedirectToAction 5](#_Toc32612468)

[Passing Data to Views 5](#_Toc32612469)

[Model 5](#_Toc32612470)

[In action 5](#_Toc32612471)

[In View 5](#_Toc32612472)

[ViewBag 6](#_Toc32612473)

[In action 6](#_Toc32612474)

[In View 6](#_Toc32612475)

[ViewBag 6](#_Toc32612476)

[In action 6](#_Toc32612477)

[In View 6](#_Toc32612478)

[Views 7](#_Toc32612479)

[Razor Views 7](#_Toc32612480)

[Partial Views 7](#_Toc32612481)

# Flow

Route -> Controller -> View

# Routes

## RouteConfig

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

}

);

}

}

## Custom Route

### Add Route

routes.MapRoute(

"MoviesByReleaseDate",

"movies/released/{year}/{month}",

new {

controller = "Movies",

action = "ByReleaseDate"

},

new {

year = “2015|2016", // require year to 2015 or 2016 month = "\\d{2}" // require month to be 2 digits

}

);

### Map to controllers

public ActionResult ByReleaseDate(int year, int month) {

return Content(year + "/" + month);

}

## Attribute Routes

### Init in RegisterRoutes

routes.MapMvcAttributeRoutes();

### Decorate controller with attribute

[Route("movies/released/{year}/{month:regex(\\{2}):range(1, 12)}")]

public ActionResult ByReleaseDate(int year, int month) {

return Content(year + "/" + month);

}

### Constraint

* regex
* range
* min
* max
* minlength
* maxlength
* int
* float
* guid

# Controllers

public class MoviesController : Controller {

// GET: Movies/Random

public ActionResult Random() {

Movie movie = new Movie() { Name = "Back To The Future" };

return View(movie);

}

}

## Controller Parameters

• Embedded in the URL: /movies/edit/1

• In the query string: /movies/edit?id=1

• In the form data

public ActionResult Edit(int id) {

return Content("id=" + id);

}

## Action Results -> View

Type Helper Method response

ViewResult View() View

PartialViewResult PartialView()

ContentResult Content() plain text

RedirectResult Redirect()

RedirectToRouteResult RedirectToAction() redirect to other action

JsonResult Json()

FileResult File()

HttpNotFoundResult HttpNotFound() 404 error

EmptyResult new EmptyResult() null

## RedirectToAction

return RedirectToAction("actionName", "controllerName", new { page = 1, sortBy = "name" });

http://.../ controller/action?page=1&sortBy=name

# Passing Data to Views

## Model

### In action

public ActionResult Random() {

Movie movie = new Movie() { Name = "Back To The Future" };

return View(movie);

}

### In View

@model Vidly.Models.Movie

@{

ViewBag.Title = "Random";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

## ViewBag

### In action

public ActionResult Random() {

Movie movie = new Movie() { Name = "Back To The Future" };

ViewData["key1"] = "value1";

ViewData["key1"] = "value1";

return View(movie);

}

### In View

<h2>@(((Movie)ViewData["key1"]).Name)</h2>

## ViewBag

### In action

public ActionResult Random() {

ViewBag.Message = “Message”;

return View(movie);

}

### In View

<h2>@ViewBag.Message</h2>

# Views

## Razor Views

@if (…) {

// C# code or HTML

}

@foreach (…) {

}

@{

var className = Model.Movie.Name == string.Empty ? "popular" : null;

}

<h2 class=“@className”>@Model.Movie.Name</h2>

@if (Model.Customers.Count == 0) {

<p>No Customers</p>

} else {

<ul>

@foreach (var customer in Model.Customers) {

<li>@customer.Name</li>

}

</ul>

}

## Comment

@\*

Comment

\*@

## Partial Views

To render \_NavBar.cshtml partial view

@Html.Partial("\_NavBar")