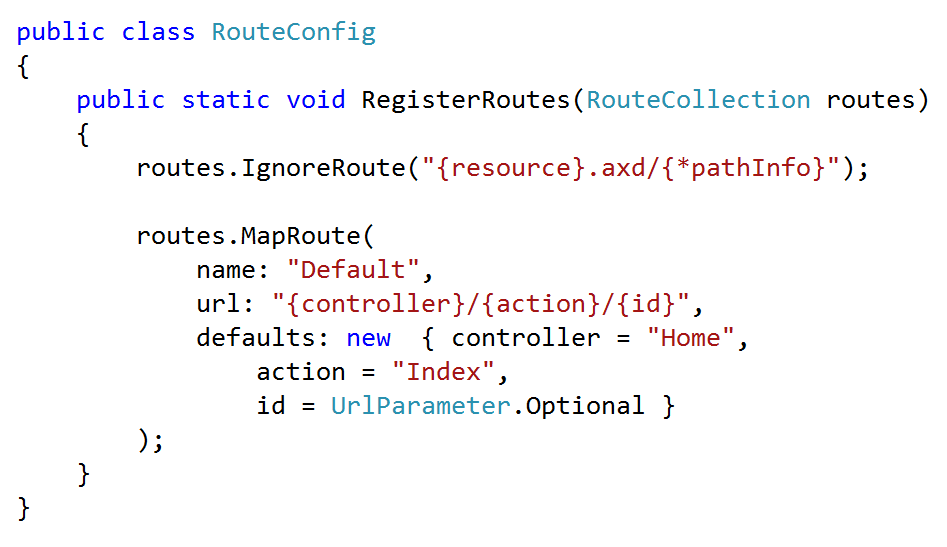
## Routing

Pattern matching system

Routing engine forwards the request to the corresponding IRouteHandler

(MvcHandler)



IRouteHandler – Default Rout handler – MVCRoutHandler

GetHttpHandler

GetSessionStateBehavior

GetControllerSessionBehavior

## MvcHandler

void IHttpHandler.ProcessRequest(HttpContext httpContext)

{

ProcessRequest(httpContext);

}

protected virtual void ProcessRequest(HttpContext httpContext)

{

HttpContextBase iHttpContext = new HttpContextWrapper(httpContext);

ProcessRequest(iHttpContext);

}

protected internal virtual void ProcessRequest(HttpContextBase httpContext) {

SecurityUtil.ProcessInApplicationTrust(() => {

IController controller;

IControllerFactory factory;

ProcessRequestInit(httpContext, out controller, out factory);

try

{

controller.Execute(RequestContext);

}

finally

{

factory.ReleaseController(controller);

}

});

}

## Controller

protected void Application\_Start()

{

AreaRegistration.RegisterAllAreas();

WebApiConfig.Register(GlobalConfiguration.Configuration);

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

RouteConfig.RegisterRoutes(RouteTable.Routes);

BundleConfig.RegisterBundles(BundleTable.Bundles);

}

## Action Execution

Controller's ActionInvoker determines which specific action to invoke on the controller

## View Result

The action method receives user input, prepares the appropriate response data, and then executes the result by returning a result type. The result type can be ViewResult, RedirectToRouteResult, RedirectResult, ContentResult, JsonResult, FileResult, and EmptyResult.

## View Engine

protected void Application\_Start()

{

//Remove All View Engine including Webform and Razor

ViewEngines.Engines.Clear();

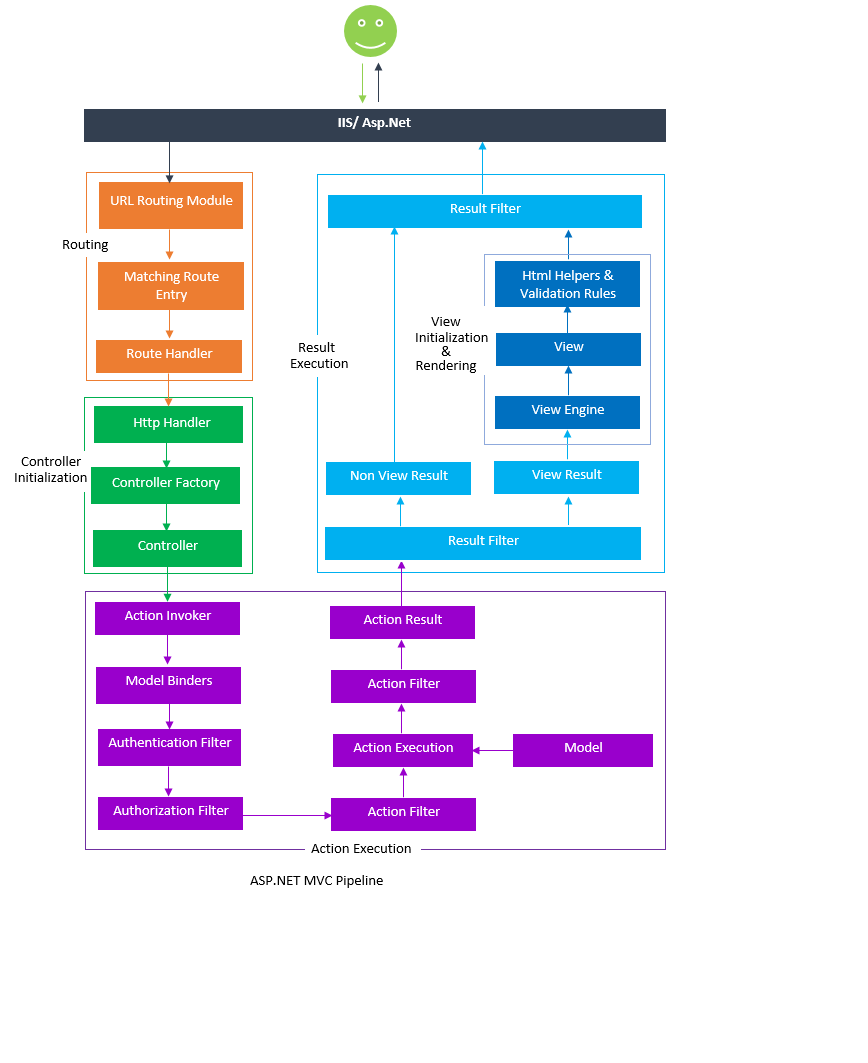
//Register Your Custom View Engine

ViewEngines.Engines.Add(new CustomViewEngine());

//Other code is removed for clarity

}

## View



Action filters are executed before(OnActionExecuting) and after(OnActionExecuted) an action is executed.

Result filters are executed before(OnResultnExecuting) and after(OnResultExecuted) the ActionResult is executed.

## Use of filters:

1. Custom Authentication
2. Custom Authorization(User based or Role based)
3. Error handling or logging
4. User Activity Logging
5. Data Caching
6. Data Compression
7. User Session handling

## Types of Filters

The ASP.NET MVC framework provides five types of filters.

1. Authentication filters (New in ASP.NET MVC5)

IAuthenticationFilter

OnAuthentication

1. Authorization filters

IAuthorizationFilter

OnAuthorization

1. Action filters

IActionFilter

OnActionExecuted, OnActionExecuting

1. Result filters

IResultFilter

OnResultExecuted, OnResultExecuting

1. Exception filters

IExceptionFilter

OnException

**Configuring Filters**

You can configure your own custom filter into your application at following three levels:

**Global level**

By registering your filter into Application\_Start event of Global.asax.cs file with the help of FilterConfig class.

protected void Application\_Start()

{

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

}

**Controller level**

By putting your filter on the top of the controller name as shown below-

[Authorize(Roles="Admin")]

public class AdminController : Controller

{

//

}

**Action level**

By putting your filter on the top of the action name as shown below-

public class UserController : Controller

{

[Authorize(Users="User1,User2")]

public ActionResult LinkLogin(string provider)

{

// TODO:

return View();

}

}

## HTML Helpers

Different types of HTML Helpers

There are three types of HTML helpers as given below:

* Inline Html Helpers
* Built-In Html Helpers
* Strongly Typed HTML Helpers

# Layouts, RenderBody, RenderSection and RenderPage

\_viewstart.cshtml

## Areas:

routes.MapRoute(  
 name: "Default",  
 url: "{controller}/{action}/{id}",  
 defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional },  
 namespaces: new [] { "TemplateSite.Mvc.Controllers" }  
).DataTokens["UseNamespaceFallback"] = false ;