Data Validation

Code Migration

Bundling / Minification

Layout > they are used to create template, so that all the views have same look and feel

The common contents/formatting we keep in Layout file .cshtml

\_Layout.cshtml

\_ViewStart.cshtml

@{

Layout=

}

Bundling

BundleConfig.cs file

Class BundleConfig

RegisterBundles () Method

using System.Web;

using System.Web.Optimization;

namespace WebApplication15

{

public class BundleConfig

{

// For more information on bundling, visit https://go.microsoft.com/fwlink/?LinkId=301862

public static void RegisterBundles(BundleCollection bundles)

{

bundles.Add(new ScriptBundle("~/bundles/jquery").Include(

"~/Scripts/jquery-{version}.js"));

bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(

"~/Scripts/jquery.validate\*"));

// Use the development version of Modernizr to develop with and learn from. Then, when you're

// ready for production, use the build tool at https://modernizr.com to pick only the tests you need.

bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(

"~/Scripts/modernizr-\*"));

bundles.Add(new ScriptBundle("~/bundles/bootstrap").Include(

"~/Scripts/bootstrap.js"));

bundles.Add(new StyleBundle("~/Content/css").Include(

"~/Content/bootstrap.css",

"~/Content/site.css",

**"~/Content/StyleSheet1.css"));**

}

}

}

Global.asax file

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Optimization;

using System.Web.Routing;

namespace WebApplication15

{

public class MvcApplication : System.Web.HttpApplication

{

protected void Application\_Start()

{

AreaRegistration.RegisterAllAreas();

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

**RouteConfig.RegisterRoutes(RouteTable.Routes);**

**BundleConfig.RegisterBundles(BundleTable.Bundles);**

}

}

}

<!Doctype html>

<html>

<head>

We don’t want to make calls to css or js files like this

@\*<link rel="stylesheet" href="~/Content/StyleSheet1.css" />

<link rel="stylesheet" href="~/Content/bootstrap-theme.css" />

<link rel="stylesheet" href="~/Content/bootstrap.css.map" />

<link rel="stylesheet" href="~/Content/StyleSheet1.css" />

<link rel="stylesheet" href="~/Content/StyleSheet1.css" />

<link rel="stylesheet" href="~/Content/StyleSheet1.css" />

<link rel="stylesheet" href="~/Content/StyleSheet1.css" />

<link rel="stylesheet" href="~/Content/bootstrap.css"/>\*@

This is a call to server as a bundle

**@Styles.Render("~/Content/css")**

<title> ASP Application </title>

</head>

<body>

<center>

<div style="align-items:center">

<h1> ABC Ltd. </h1>

<p>

Attribute Routing gives us more control over the URIs in our MVC web application. The earlier way of routing (convention-based routing) is fully supported by this version of MVC. We can also use both type of routing in the same project.

</p>

<p>

Attribute Routing gives us more control over the URIs in our MVC web application. The earlier way of routing (convention-based routing) is fully supported by this version of MVC. We can also use both type of routing in the same project.

</p>

</div>

</center>

@RenderSection("A1", required:false)

@RenderBody()

</body>

</html>

Bundling : Grouping of similar type of files( css , js) so that they can be fetched from server in a single call

Minification > Removing extra lines, gaps so that the file size is reduced. It also save time while downloading the files (css.js)