

MVC Guide

Table of Contents

- 1 – DLL List 2
- 2 – Razor Reserved Words 2
- 3 – MVC Folders Application Structure 2
- 4 – Security 3
- 5 – Code Snippets 3
- Appendix 6

MVC Guide

1 – DLL List :

System.Web.Mvc	Web.Abstractions	Web.Routing	Web.WebPages
Web.Helpers	Microsoft.OWIN		
		System.Web	
Microsoft.CSharp	System	System.Core	System.Data
System.Data.DataSetExtensions	System.Xml	System.Xml.Linq	EntityFramework
EntityFramework.SqlServer			
ComponentModel.DataAnnotations			

2 – Razor Reserved Words :

- Razor code blocks are enclosed in @{ ... }
- Inline expressions (variables and functions) start with @
- Code statements end with semicolon
- Variables are declared with the var keyword
- Strings are enclosed with quotation marks
- C# code is case sensitive
- C# files have the extension .cshtml

@using	@(inherits)	@functions	@section
@helper	@model	@for	@if
@do	@try	@foreach	@while
@switch	@lock	@case	default
@Html.Partial			

3 – MVC Folders Application Structure :

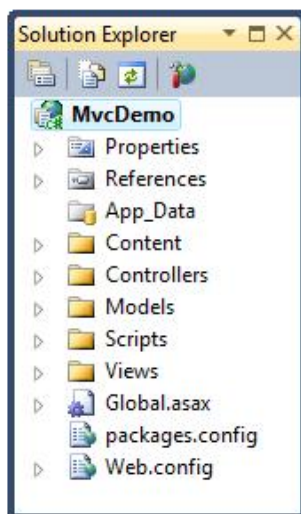
Here is list statements with their reserved words :

- Web.Config – configuration file containing application settings.
- Packages.Config – NuGet package configuration file. Check it to see Microsoft.AspNet.Mvc version.
- Global.asax – ApplicationStart method, AreaRegistration, GlobalConfiguration, FilterConfig, RouteConfig, BundleConfig.
- Startup.cs – Microsoft.OWIN (Katana Project), Configuration method & IAppBuilder Interface.
- App_Data – contains all application data.
- Areas –
- Fonts –

MVC Guide

- Content – static files like themes, stylesheets, icons, images, bootstrap.css, site.css
- Models – business objects.
- Controllers – Controllers that receive user input and respond. AccountController, HomeController
- Providers –
- Scripts –

A typical ASP.NET MVC web application has the following folder content:



Application information

Properties
References

Application folders

App_Data Folder
Content Folder
Controllers Folder
Models Folder
Scripts Folder
Views Folder

Configuration files

Global.asax
packages.config
Web.config

4 – Security :

The days of asp.net membership providers are now gone and the security layers have been moved into OWIN and Katana and they are interface driven with ability to do claims based authentication using OAUTH.

- OWIN has a startup function that uses the IApplicationBuilder interface to setup security.
- The interface has a Run function that accepts an IowinContext which has Request, Response, Environment etc.
- Use to build components instead of using the System.Web dll which is large.

5 – Code Snippets :

Some C# code snippets that are useful

```
// Declaring Property getters and setters for Entities
public int ProductID { get; set; }
public string ProductName { get; set; }
public int? SupplierID { get; set; }
```

MVC Guide

```
public string QuantityPerUnit { get; set; }
public decimal? UnitPrice { get; set; }
public short? UnitsInStock { get; set; }
public bool Discontinued { get; set; }
public string CategoryName { get; set; }
```

```
// Declare collections for Entities
public virtual ICollection<Product> Products { get; set; }
```

```
// Sample Controller
public class CustomerController : EntitySetController<Customer, string>
{
    private readonly NorthwindContext _northwindContext;

    public CustomerController()
    {
        _northwindContext = new NorthwindContext();
    }

    public override IQueryable<Customer> Get()
    {
        return _northwindContext.Customers;
    }

    protected override Customer GetEntityByKey(string key)
    {
        return _northwindContext.Customers.Find(key);
    }

    protected override Customer UpdateEntity(string key, Customer update)
    {
        _northwindContext.Customers.AddOrUpdate(update);
        _northwindContext.SaveChanges();
        return update;
    }

    public override void Delete(string key)
    {
        var customer = _northwindContext.Customers.Find(key);
        _northwindContext.Customers.Remove(customer);
        _northwindContext.SaveChanges();
    }
}
```

```
// Web configuration
public class BundleConfig
```

MVC Guide

```
{
    // For more information on bundling, visit
    http://go.microsoft.com/fwlink/?LinkId=301862
    public static void RegisterBundles(BundleCollection bundles)
    {
        bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
            "~/Scripts/jquery-{version}.js"));

        // Use the development version of Modernizr to develop with and learn from.
        Then, when you're
        // ready for production, use the build tool at http://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",
            "~/Scripts/respond.js"));

        bundles.Add(new StyleBundle("~/Content/css").Include(
            "~/Content/bootstrap.css",
            "~/Content/site.css"));
    }
}
// join keyword
// Sample Controller
public class AccountSetupController : MyAccountController
{
    // GET: /MyAccount/
    public ActionResult Index()
    {
        return View();
    }

    // POST: /MyAccount/AccountSetup/CreateProfile
    public ActionResult CreateProfile(ProfileData profile)
    {
        if (ModelState.IsValid)
        {
        }
        else
        {
            return View("Index", profile);
        }
    }
}
```

MVC Guide

Appendix :

Add links here