In ASP.NET MVC4, mobile enability to the Web Application can be given by running \_\_\_\_\_\_\_\_\_\_\_\_\_command in the package manager console in Visual studio.

a)Install-Package jquery.Mobile.MVC   
b)Install-Package jquery.MVC.Mobile  
c)Install-Package jquery.Web.Mobile  
d)Install-Package jquery.MVC.MobileSupport

which version of .NET Framework support, simple implementation of async programming using "async", "await" keywords and "Task" class ?

a).NET 4.5   
b).NET 4.0  
c) .NET 3.5  
d) .NET 3.0

Which one from the following techniques , helps reducing the number of HTTP requests that browsers initiate to download the web resources from the web Server ?

a)mobile enablility

b)Bundling

c)Minification

d)Compression

Which one from the following statements is true as per the naming convention in Bundling technique in ASP.NET MVC4?

a) var ScriptBundle1 = new ScriptBundle("~/MyScriptBundle");

b) var ScriptBundle1 = new ScriptBundle("~MyScriptBundle");

c) var ScriptBundle1 = new ScriptBundle("MyScriptBundle");

d)var ScriptBundle1 = new ScriptBundle("MyScriptBundle/~");

Which one from the following namespaces support Bundling & Minification.

a)System.Optimization

b) System.Web.Optimize

c)System.Web.Optimization

d)System.Web.Minify

Which one from the following options , enables you to create a bundles of multiple JavaScript or CSS files, .

a)Optimization

b)Bundling

c)Minification

d)Compression

which one from the following is not the correct statement for the Mobile support provided for ASP.NET MVC 4 Web Application ?

a)Mobile enablility can be obtained by "Install-Package jquery.Mobile.MVC" command

b)It will have a mobile master view called \_Layout.Mobile.cshtml

c)It will have \_ViewSwitcher.cshtml

d) After obaining the mobile enablity, the application will not work for Desktop users.

In ASP.NET MVC 4, \_\_\_\_\_\_\_\_\_\_\_ is used in the \_Layout.cshtml to render the Script Bundle.?

a) @Script.Render()

b) @Scripts.Render()

c) @Script.Rendering()

d) @Script.RenderStyle()

Which one from the following featuers of ASP.NET MVC4 is originally supported by Entity Framework ?

a)Bundling and minification

b)Page Inspector

c) Database Migration

d)Scaffolding

Which Type from the following, does not belong to Bundling technique in ASP.NET MVC4?

a)SytleBundle

b)BundleTable

c) BundleCollection

d) ResourceBundle

Which one from the following types, belongs to Bundling technique in ASP.NET MVC4?

a)ScriptBundle

b)BundleList

c) BundleResource

d)BundleQueue

Which one from the following statements is not ture for Page Inspector in ASP.NET MVC4 ?

a)Works on IE9 and above

b)Works on other browsers than IE as wll

c) Visual studio also plays role in it

d)It?s a design time site customiztaion tool

ASP.NET MVC4, supports "OAuth" through \_\_\_\_\_\_\_\_\_\_\_.

a)Microsoft.WebPages.OAuth.dll

b)Microsoft.Web.WebPages.OAuth.dll

c)System.Web.WebPages.OAuth.dll

d)Microsoft.Web.OAuth.dll

Which one from the following techniques, reduces the size of .js and .css files, thereby drastically minimizing the amount of content that has to be transmitted from Web Server to the browser .

a)mobile enablility

b)Bundling

c)Minification

d)Routing

In ASP.NET MVC 4,\_\_\_\_\_\_\_\_\_\_\_ is used in the \_Layout.cshtml to render the Style Bundle.?

a) @Style.Render()

b) @Styles.Render()

c) @Style.Rendering()

d) @Style.RenderStyle()

Which one from the following featuers of ASP.NET MVC4 is originally supported by .NET Framework 4.5 ?

a)Page Inspector

b)Database Migration

c) async programming using async and await keywords

d)Scaffolding

Which one from the following does not play any role in implementing asynchrounous programming in ASP.NET MVC 4 ?

a)async keyword

b)await keyword

c) IResult

d) Task class

Which one from the following is not true for Bundling technique in ASP.NET MVC4?

a)helps reducing the number of HTTP requests that browsers initiate to download the web resources from the web Server ?

b)enables you to create a bundles of multiple JavaScript and/or CSS files, .

c) Supported by System.Web.Optimization.dll

d) It has nothing to do with improving overall performance of the Web Site

**Questions**

1. Example of Eager loading (Eager loading does this by using the Include method)
2. Example of Explicit loading (we use load method to retrieve)
3. Jquery Mobile & Progressive Enhancement
4. Html.TextBox() in Razor View -- @Html.TextBox
5. Syntax for Including style bundle in HTML head

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

1. Syntax for Including script bundle in in HTML head  
   **@Scripts.Render("~/bundles/bootstrap")**
2. WEB API, web service & WCF
3. HTTP get or post what will u use if you have to allow only get or post requests
4. Bundling is used to combine the files
5. Bundling and magnification name space
6. Can we add authentication using social media sites. oAuth or dotnetoAuth
7. UI logic can be written in View
8. Which of the following is not included in View logic
9. Redirecting to other URL = @Html.ActionLink syntax and @Ajax.ActionLink

|  |  |
| --- | --- |
| 1. Which of the following feature is not included in MVC4 2. Validating the model syntax ModelState.IsValid |  |

1. Action filter

The ASP.NET MVC framework supports four different types of filters:1

1. Authorization filters – Implements the IAuthorizationFilter attribute.
2. Action filters – Implements the IActionFilter attribute.
3. Result filters – Implements the IResultFilter attribute.
4. Exception filters – Implements the IExceptionFilter attribute.
5. Which of the following is not a feature of asp.net mvc 4.
6. Async, await and task

<https://www.tektutorialshub.com/lazy-loading-entity-framework/>

static void LazyLoading()

{

using (AdventureWorks db = new AdventureWorks())

{

db.Database.Log = Console.Write;

var product = (from p in db.Products

where p.ProductID == 814

select p).ToList();

foreach (var p in product)

{

Console.WriteLine("{0} {1} {2}", p.ProductID, p.Name, p.ProductModel.Name);

}

Console.ReadLine();

}

}

The Code above retrieves the Product from the database when the ToList() method is called. The Entity framework does not retrieve the Product model at that time. The Product model name is accessed from inside the for the loop. At that point, Entity Framework sends the query to the database to retrieve the ProductModel. This behaviour is known as Lazy Loading.

<https://www.tektutorialshub.com/eager-loading-entity-framework/>

using (AdventureWorks db = new AdventureWorks())

{

//Disable Lazy Loading

db.Configuration.LazyLoadingEnabled = false;

//Log SQL Command to Console

db.Database.Log = Console.Write;

var product = (from p in db.Products

.Include("ProductModel") //ProductModel table to be included in the result

where p.ProductID == 814

select p).ToList();

foreach (var p in product)

{

Console.WriteLine("{0} {1} {2}", p.ProductID, p.Name, p.ProductModel.Name);

}

}

Eager loading solves this problem by loading all the product models along with products using a single query. Eager loading does this by using the Include method. Entity Framework creates a join query, when it sees the Include method, thus bringing all the records in one single query.

<https://www.tektutorialshub.com/explicit-loading-entity-framework/>

using (AdventureWorks db = new AdventureWorks())

{

//Disable Lazy Loading

db.Configuration.LazyLoadingEnabled = false;

//Log Database

db.Database.Log = Console.Write;

//List of Products queried here.

var product = (from p in db.Products

orderby p.ProductID descending

select p).Take(5).ToList();

foreach (var p in product)

{

//Product model is retrieved here

db.Entry(p).Reference(m => m.ProductModel).Load();

Console.WriteLine("{0} {1} Product Model => {2}", p.ProductID, p.Name, ( p.ProductModel==null) ? "" : p.ProductModel.Name );

Console.ReadKey();

}

}

In the above query, we retrieve five products from the database. Inside the for loop of each product, we use load method to retrieve the product model data

The Microsoft.Web.WebPages.OAuth namespace contains core classes that are used to work with OAuth and OpenID authentication.

The Microsoft.Owin.Security.OAuth namespace contains types related to OAuth providers.

**Bundling** is a new feature in ASP.NET 4.5 that makes it easy to combine or bundle multiple files into a single file. **Minification** performs a variety of different code optimizations to scripts or css, such as removing unnecessary white space and comments and shortening variable names to one character.

bundles.Add(new ScriptBundle("~/bundles/jquery").Include("~/Scripts/jquery-{version}.js"));  
<%: Scripts.Render("~/bundles/jquery") %>

ScriptBundle thirdPartyScripts = new ScriptBundle("~/Scripts/ThirdParty");  
thirdPartyScripts.Include("~/Scripts/jquery-{version}.js","~/Scripts/bootstrap.min.js");  
bundles.Add(thirdPartyScripts);

Optimistic Locking is a strategy where you read a record, take note of a version number (other methods to do this involve dates, timestamps or checksums/hashes) and check that the version hasn't changed before you write the record back. When you write the record back you filter the update on the version to make sure it's atomic. (i.e. hasn't been updated between when you check the version and write the record to the disk) and update the version in one hit.

If the record is dirty (i.e. different version to yours) you abort the transaction and the user can re-start it.

This strategy is most applicable to high-volume systems and three-tier architectures where you do not necessarily maintain a connection to the database for your session. In this situation the client cannot actually maintain database locks as the connections are taken from a pool and you may not be using the same connection from one access to the next.

Pessimistic Locking is when you lock the record for your exclusive use until you have finished with it. It has much better integrity than optimistic locking but requires you to be careful with your application design to avoid Deadlocks.

Defining Attributes in Metadata Classes

System.ComponentModel.DataAnnotations

@\*To call same controller view\*@

@Html.ActionLink("Index", "ActionName")

@\*To call another controller view\*@

@Html.ActionLink("Index", "ActionName", "ControllerName", null, new { id = "OT", style = "color: white”})

routes.MapRoute(

name: "DefaultRoute",

url: "Home/{action}",

defaults: new { controller = "Home", action = "Index" },

constraints: new { action = "[A-Za-z]\*" }

);