Articles from Jinal Desai .NET

ASP.NET MVC 3 Interview Questions

2011-10-21 15:10:50 Jinal Desai

What is MVC?

MVC is a framework methodology that divides an application's implementation into three component roles: models, views, and controllers.

"Models" in a MVC based application are the components of the application that are responsible for maintaining state. Often this state is persisted inside a database (for example: we might have a Product class that is used to represent order data from the Products table inside SQL).
"Views" in a MVC based application are the components responsible for displaying the application's user interface. Typically this UI is created off of the model data (for example: we might create an Product "Edit" view that surfaces textboxes, dropdowns and checkboxes based on the current state of a Product object).
"Controllers" in a MVC based application are the components responsible for handling end user interaction, manipulating the model, and ultimately choosing a view to render to display UI. In a MVC application the view is only about displaying information – it is the controller that handles and responds to user input and interaction.

Which are the advantages of using MVC Framework?

MVC is one of the most used architecture pattern in ASP.NET and this is one of those ASP.NET interview question to test that do you really understand the importance of model view controller.
1. It provides a clean separation of concerns between UI and model.
2. UI can be unit test thus automating UI testing.
3. Better reuse of views and model. You can have multiple views which can point to the same model and also vice versa.
4. Code is better organized.

What is Razor View Engine?

Razor view engine is a new view engine created with ASP.Net MVC model using specially designed Razor parser to render the HTML out of dynamic server side code. It allows us to write Compact, Expressive, Clean and Fluid code with new syntaxes to include server side code in to HTML.

What is namespace of asp.net mvc?

ASP.NET MVC namespaces and classes are located in the System.Web.Mvc assembly.

System.Web.Mvc namespace
Contains classes and interfaces that support the MVC pattern for ASP.NET Web applications. This namespace includes classes that represent controllers, controller factories, action results, views, partial views, and model binders.

System.Web.Mvc.Ajax namespace
Contains classes that support Ajax scripts in an ASP.NET MVC application. The namespace includes support for Ajax scripts and Ajax option settings.

System.Web.Mvc.Async namespace
Contains classes and interfaces that support asynchronous actions in an ASP.NET MVC application

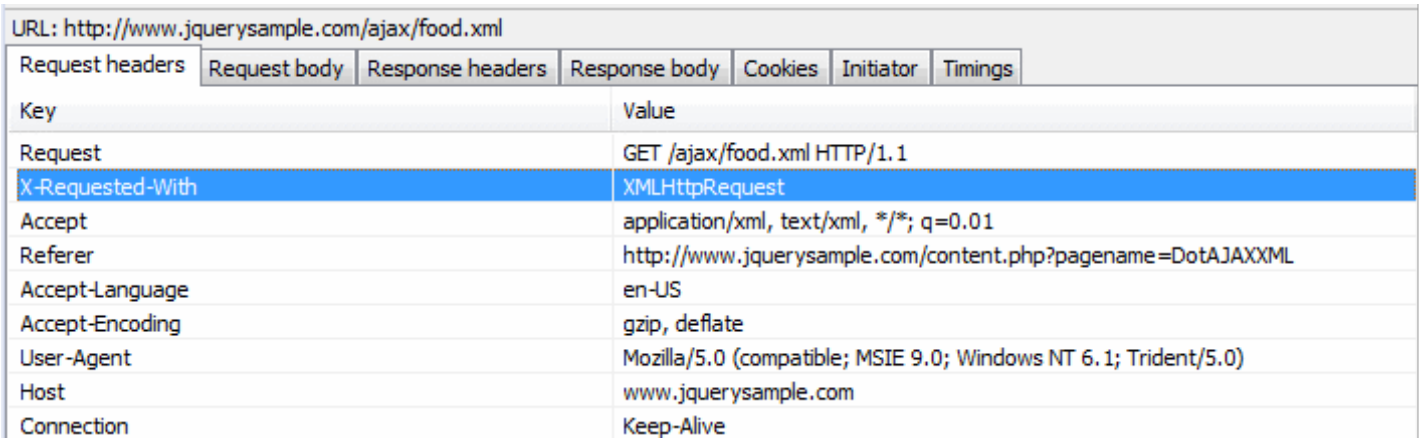
System.Web.Mvc.Html namespace

Contains classes that help render HTML controls in an MVC application. The namespace includes classes that support forms, input controls, links, partial views, and validation.

How to identify AJAX request with C# in MVC.NET?

The solution is in depended from MVC.NET framework and universal across server-side technologies. Most modern AJAX applications utilize XmlHTTPRequest to send async request to the server. Such requests will have distinct request header:

X-Requested-With = XMLHTTPREQUEST



MVC.NET provides helper function to check for ajax requests which internally inspects X-Requested-With request header to set IsAjax flag.

HelperPage.IsAjax Property
Gets a value that indicates whether Ajax is being used during the request of the Web page.

Namespace: System.Web.WebPages
Assembly: System.Web.WebPages.dll

However, same can be achieved by checking requests header directly:

Request["X-Requested-With"] == “XmlHttpRequest”

What is Repository Pattern in ASP.NET MVC?

Repository pattern is usefult for decoupling entity operations form presentation, which allows easy mocking and unit testing.

“The Repository will delegate to the appropriate infrastructure services to get the job done. Encapsulating in the mechanisms of storage, retrieval and query is the most basic feature of a Repository implementation”

“Most common queries should also be hard coded to the Repositories as methods.”

Which MVC.NET to implement repository pattern Controller would have 2 constructors on parameterless for framework to call, and the second one which takes repository as an input:

```
class myController: Controller
{
    private IMyRepository repository;

    // overloaded constructor
    public myController(IMyRepository repository)
    {
        this.repository = repository;
    }
```

```
    // default constructor for framework to call
    public myController()
    {
        //concreate implementation
        myController(new someRepository());
    }
...

    public ActionResult Load()
    {
        // loading data from repository
        var myData = repository.Load();
    }
}
```

What is difference between MVC(Model-View-Controller) and MVP(Model-View-Presenter)?

The main difference between the two is **how the manager (controller/presenter) sits in the overall architecture**.

All requests goes first to the Controller
MVC pattern puts the controller as the main 'guy' in charge for running the show. All application request comes through straight to the controller, and it will decide what to do with the request.

Giving this level of authority to the controller isn't an easy task in most cases. Users interaction in an application happen most of the time on the View.

Thus to adopt MVC pattern in a web application, for example, the url need to become a way of instantiating a specific controller, rather than 'simply' finding the right View (webform/ html page) to render out. Every requests need to trigger the instantiation of a controller which will eventually produce a response to the user.

This is the reason why it's alot more difficult to implement pure MVC using Asp.Net Webform. The Url routing system in Asp.Net webform by default is tied in to the server filesystem or IIS virtual directory structure. Each of these aspx files are essentially Views which will always get called and instantiated first before any other classes in the project. (Of course I'm overgeneralizing here. Classes like IHttpModule, IHttpHandler and Global.asax would be instantiated first before the aspx web form pages).

MVP (Supervising Controller) on the other hand, doesn't mind for the View to take on a bigger role. View is the first object instantiated in the execution pipeline, which then responsible for passing any events that happens on itself to the Presenter.

The presenter then fetch the Models, and pass it back to the view for rendering.

What is the 'page lifecycle' of an ASP.NET MVC?

Following process are performed by ASP.Net MVC page:
1) App initialization
2) Routing
3) Instantiate and execute controller
4) Locate and invoke controller action
5) Instantiate and render view

How to call javascript function on the change of Dropdown List in ASP.NET MVC?

Create a java-script function:

```
<script type="text/javascript">
            function selectedIndexChanged() {
            }
</script>
```

Call the function:

```
<%:Html.DropDownListFor(x => x.SelectedProduct,
new SelectList(Model.Products, "Value", "Text"),
"Please Select a product", new { id = "dropDown1",
onchange="selectedIndexChanged()" })%>
```

How route table is created in ASP.NET MVC?

When an MVC application first starts, the Application_Start() method is called. This method, in turn, calls the RegisterRoutes() method. The RegisterRoutes() method creates the route table.

How do you avoid XSS Vulnerabilities in ASP.NET MVC?

Use thesyntax in ASP.NET MVC instead of usingin .net framework 4.0.

Explain how to access Viewstate values of this page in the next page?

PreviousPage property is set to the page property of the nest page to access the viewstate value of the page in the next page.

```
Page poster = this.PreviousPage;
```

Once that is done, a control can be found from the previous page and its state can be read.

```
Label posterLabel = poster.findControl("myLabel");
 string lbl = posterLabel.Text;
```

How to create dynamic property with the help of viewbag in ASP.NET MVC?

PreviousPage property is set to the page property of the nest page to access the viewstate value of the page in the next page.

```
Page poster = this.PreviousPage;
```

Once that is done, a control can be found from the previous page and its state can be read.

```
Label posterLabel = poster.findControl("myLabel");
 string lbl = posterLabel.Text;
```

What is difference between Viewbag and Viewdata in ASP.NET MVC?

The basic difference between ViewData and ViewBag is that in ViewData instead creating dynamic properties we use properties of Model to transport the Model data in View and in ViewBag we can create dynamic properties without using Model data.

What is Routing?

A route is a URL pattern that is mapped to a handler. The handler can be a physical file, such as an .aspx file in a Web Forms application. Routing module is responsible for mapping incoming browser requests to particular MVC controller actions.

For detailed Interview Questions refer my second post regarding MVC Interview Questions.