Interview Questions You Should Know – Web

Before going into a web developer interview you should already have basic HTML and CSS Fundamentals. If not, use w3schools, codecademy, etc. There is **no excuse** for not knowing the basics and you will fail out of an interview quickly if you can't talk about the most common properties and tags.

Tags you should know- html, head, body, h1-h6, b, I, a, p, br, blockquote, ul, ol, li, div, img, table, tr, td, thead, tbody, form, input, hidden, select, option, button.

HTML Attributes you should know- id, name, class, placeholder, src, style, title, value, href, disabled

CSS Properties you should know- text-align, text-decoration, font-family, font-size, font-weight, width, height, margin, padding, color, background-color, background-image, border, border-color, list-style, float, clear, z-index.

What is MVC?

MVC stands for Model-View-Controller and it is an architecture pattern. Its purpose is to separate the representation from user interaction:

- The View is responsible for the look and feel
- The Model represents the data object that a view is bound to
- The Controller is responsible for handling user requests and loading the appropriate Views and Models

MVC is used in more than just the web, since it is a generic pattern for building graphical applications. MVC has two main benefits:

1. Separation of concerns- The code that assembles the view is separate from the view itself
2. Automated UI testing is easier because the controller logic is separated into its own class

What ways can we pass data to a web server?

There are several ways to get data from the client to the server:

1. As part of a URL segment (route). Example: customer/edit**/1**
2. As a variable in the URL's querystring. Example: product/list**?CategoryId=1&SubcategoryId=5**
3. Data contained inside a <form> tag. Input, hidden, and select elements will be sent back to the server as part of the request body.
4. We can also store small amounts of data in cookies in the browser.

Describe how a web request works in ASP.NET MVC?

Every web request has two main execution steps: the Request and the Response.

A request is initiated from a client (usually a browser), when the request is received by the server there are four main steps:

1. **Determine Route**- MVC requests first check if the URL points to a physical file on disk, if it does not the URL is mapped to a route in the Route.Config which determines which controller and action needs to be invoked. If a physical file is found (.html or a file/image) the file is served to the requestor directly.
2. **Populate Route Data-** ASP.NET MVC looks at the URL segments and determines if there are any variables that need their data registered. {controller}/{action}/{id} is the default route, so given a URL of customers/edit/1 it will pull the controller and action then create a RouteData key called id and set its value to 1.
3. **Request Context Created**- The request header and body from the client is used to create the C# Request object which is populated with all of the relevant information from the request (RouteData, QueryString, and Form data, among other things)
4. **Controller Instantiated**- The request object is sent to the MvcHandler which instantiates a controller instance and invokes the right ActionMethod. The Request object data is used to automatically bind any parameters of the controller method that is invoked. This is called **model binding.**

When the controller action is invoked we enter the second phase of processing, the **response** phase. The controller action executes and sends a response back to the client. The ActionResult base class is the most common return type in MVC controllers. An ActionResult can be one of the following (there are more, but the first 5 are the most common):

- ViewResult
- RedirectToAction (or RedirectToRoute)
- JsonResult
- HttpStatusCode
- FileResult (downloads a file)
- JavaScriptResult
- HttpUnauthorizedResult
- HttpNotFoundResult
- EmptyResult
- ContentResult

What is a HTML helper?

The HTML helper class assists in rendering html tags in the view as well as ensuring that the names of the generated tags facilitate the automatic model binding. An example of a HTML Helper class is @HTML.TextBoxFor(m=>m.LastName) which creates <input type=”text” id=”LastName” name=”LastName” />

What is the difference between HTML.TextBoxFor and HTML.TextBox?

Both of them provide a input tag of type text, however the Html.TextBoxFor is strongly typed to the view's model class while Html.TextBox is not.

@HTML.TextBoxFor(m=>m.LastName)
and
@Html.TextBox("LastName")

Will render the same output.

Can we map multiple URLs to the same action?

Yes, we could make additional entries into our Route.Config to map many URLs to the same action. This is often done when the URL structure of our site changes but we don't want to break the old links (which may be referenced on other sites or search engines).

How can we move between views using hyperlinks?

The HTML Helper has an ActionLink method. The ActionLink takes a anywhere from 2 to 5 parameters as follows:

@HTML.ActionLink(display text, action, controller, route data, html attributes)

@HTML.ActionLink("Product List", "Index", "Customer", new{ id=5 }, new{ @class="prettyLink"})
Produces
Product List

What are HTTP Verbs?

The HTTP specification defines verbs to indicate the desired action for the server to perform on a resource. The common verbs are:

- **GET**- Retrieve a resource. No data should be changed in the request
- **POST**- Create a brand new resource
- **PUT**- Update an existing resource
- **DELETE**- Delete a resource

Some older browsers and servers do not support PUT and DELETE and require all data changes to be done with POST requests.

There are some other verbs such as head, trace, options, connect, and patch, but they are beyond the scope of junior knowledge.

What is the difference between TempData, Session, Caching, and Cookies?

All of these are ways to maintain data state between calls. Normally the web is **stateless** which means that each request runs in isolation and all data associated with that request is destroyed after the response is sent back to the requestor.

- **TempData** is a dictionary object that can store data from one action to another. If you RedirectToAction and want to send some data along for the ride it will keep that data for one and only one extra call.
- **Session** is a dictionary object that will hold data for a user until it times out. As long as the user is active on the site the session data will remain. Session puts a cookie on the user's browser which identifies their unique session and holds information in memory until the user is no longer active and it times out. Session can be a cause of poor performance since if you create and store sessions for thousands of users at once your server can run low on memory.
- **Caching** is a dictionary object associated with the entire application. Session is per-user but the cache is shared among all users. It is common to put data that doesn't change very often into the cache (such as the list of US states) instead of making database calls every time a page is loaded that requires that common, relatively static information. Caching is one of the first techniques for improving website performance.
- **Cookies** are small bits of data stored in the browser which are used to hold common settings, security tokens, and other things. Since the browser user can view the contents of cookies it is not recommended to put private data in them and if you must the data should be encrypted.

What is the ViewBag?

The view bag is a dynamic wrapper that you can add additional view data to from the controller. It is often used for extra data which is not part of the model, such as status messages, etc.

How can we validate data in ASP.NET MVC?

There are three ways of doing validation in ASP.NET MVC:

1. **Data Annotations**- Built-in or custom annotations can be added to models and model properties to validate data.
2. **ModelState.AddModelError()**- Can be called in the controller to add error messages directly
3. **Implement IValidatableObject**- When this interface is added to a model class we can write custom validation logic and add errors.

Server side validators send messages to the @HTML.ValidationSummary or the @HTML.ValidationMessageFor() helpers.

We can use frameworks like angularjs and jquery validate on the browser side but we must always also validate on the server side since users can disable JavaScript.

What is WebAPI?

WebAPI is a framework for exposing data services to browsers and other devices. While WebAPI is similar to MVC in that you have routes and controllers, it requires programming against HTTP Verbs. Instead of returning view results, WebAPI returns data and HTTP status codes. It will automatically convert C# objects into JSON, XML and supports OData.

What is jQuery?

jQuery is a JavaScript library which helps to traverse the HTML DOM. It has many shortcuts for things like manipulating page content, animations, Ajax, and other commonly desired features which are cumbersome or difficult to do in basic JavaScript. It also has cross browser support to ensure consistency across browser experiences.

Is jQuery a replacement for JavaScript?

No. jQuery was written on top of JavaScript and anything but the most simple of interactions will require mixing in the JavaScript language into the jQuery functions.

How do you get jQuery into your site?

You need to add the library JavaScript files using a <script> tag. You can download the files, use NuGet to install it in your project, or reference a CDN (content delivery network) to link it into your pages.

What is the common starting point of jQuery execution?

The $(document).ready() function is executed when the DOM finishes loading. You can have multiple document ready methods on a page.

What does the dollar sign mean in jQuery?

The dollar sign is an alias for jQuery. $('p') and jQuery('p') would both select all paragraph tags on a page. The $ sign is more common.

What are selectors? Name some.

Selectors are used by jQuery to locate elements on a page. Selectors are:

- **Tag**- $('p') would get all paragraph tags
- **ID**- $('#myTable') would select the element with id="myTable"
- **Class**- $('.red') would select all elements with the 'red' css class
- **Multiple**- $('p, div') would select all paragraph and divs
- **Inner**- $('tr td:first') would select the first td in each table row

What are some common events in jQuery?

We can add custom logic that runs during certain events on a page. The most common are click() dblclick(), change(), blur(), focus(), keypress(), mouseenter(), mouseleave(), mousemove(), mouseout(), mouseover(), resize(), scroll(), and submit().

What are some ways to make ajax requests in jQuery?

jQuery provides several methods to make ajax calls:

- **Load()-** Load a piece of html from a server
- **$.getJSON()-** Send a GET request to a URL
- **$.getScript()-** Load a JavaScript file
- **$.get()-** Make a GET request that may or may not be JSON (configurable)
- **$.post()-** Make a POST call and handle the response.
- **$.ajax()-** Make other types of calls (PUT, DELETE) and control all parts of the request.