

Questions:

1. What is Web.config?
2. What can be stored in Web.config file?
3. What is the use of customErrors tag in Web.config file?
4. Give an example for customErrors tag in Web.config file.
5. How to Encrypt/Decrypt the Web.config file in .NET?
6. How to use two connection strings on development vs production?
7. What is URL re-writing?
8. Give an example for URL re-writing?
9. Leaving Tracing Enabled in Web-Based Applications is good practice?
10. Does Web.config file case-sensitive?
11. How many type of modes we can say for <authentication> element in Web.config?
12. Can one directory contain multiple Web.config files?
13. Leaving cookieless Session State Enabled is good practice?
14. Can you tell the location of the root Web.config file from which all Web.config file inherit?
15. What is the root tag of Web.config file?
16. For which purpose you use <appSettings> tag?
17. Explain <httpHandlers> tag in Web.config?
18. What are the default HTTP handlers available in ASP.NET?
19. Web.config file is stored in which form?
20. What is the difference between Web.config and Machine.config?
21. What is the difference between Web.config and App.config?
22. In what scenarios will a project have two Web.config files?
23. What is "ConfigSource" attribute in "appSettings" section?

Questions and Answers:

1. What is Web.config?

Answer: Web.config is main settings and configuration file for an ASP.NET and Asp.net MVC web applications. It is an XML document that resides in the root directory of the site or application. Web.config contains all application settings and decides how the web application will act.

2. What can be stored in Web.config file?

Answer: There are number of important settings that can be stored in the configuration file. Here are some of the most frequently used configurations, stored conveniently inside Web.config file.

- a) Database connections.
- b) Session States
- c) Error Handling (CustomError Page Settings.)
- d) Security (Authentication modes)

3. What is the use of customErrors tag in Web.config file?

Answer: CustomErrors tag provides information about custom error messages for an ASP.NET application. The customErrors element can be defined at any level in the application file hierarchy.

4. Give an example for customErrors tag in Web.config file.

Answer: The customErrors section consists of defaultRedirect and mode attributes which specify the default redirect page and the on/off mode respectively. The subsection of customErrors section allows redirecting to specified page depending on the error status code.

```
<customErrors defaultRedirect="/testApp/resources/error/ErrorAppLevel.aspx" mode="On">  
    <error redirect="/testApp/resources/error/Error404.aspx" statusCode="404" />  
</customErrors>
```

5. How to Encrypt/Decrypt the Web.config file in .NET?

Answer: You can Encrypt/Decrypt Web.config or any other configuration files, there are two options to complete this task.

- a) Using protected configuration classes in the System.Configuration namespace to encrypt and decrypt sections of a Web configuration file.
- b) You can use the ASP.NET IIS Registration Tool (Aspnet_regiis.exe) to encrypt or decrypt sections of a Web configuration file.
- c) ASP.NET will automatically decrypt encrypted configuration elements when the Web.config file is processed.

6. How to use two connection strings on development vs production?

Answer: Visual Studio has a best feature that allows you to have more than one Web.config file for different environments. Web.config transformations in Visual Studio allow you to override Web.config

settings based on Solution configuration (Debug / Release). However, you might want to have a different Web.config file for each environment - for example Web.Staging.Config or Web.Live.Config. The best thing about this feature is that it is easy to set up and you can be up and running in no time.

7. What is URL re-writing?

Answer: URL Re-Writing is a process where the path of the URL is rewritten. This is done with the help of Context.RewritePath(....). An alternate way is add UrlMapping in web.config.

8. Give an example for URL re-writing?

Answer:

```
<urlMappings>
<add url="~/jobs.aspx" mappedUrl="~/Category.aspx?CatId=1"/>
<add url="~/development.aspx" mappedUrl="~/Category.aspx?CatId=2"/>
<add url="~/training.aspx" mappedUrl="~/Category.aspx?CatId=3"/>
</urlMappings>
```

9. Leaving Tracing Enabled in Web-Based Applications is good practice?

Answer: Unfortunately tracing enabled in production environment is dangerous because it is also one of the most useful tools that a hacker can use to attack your applications.

a) Typical misconfiguration:

```
<configuration>
  <system.Web>
    <trace enabled="true" localOnly="false"/>
  </system.Web>
</configuration>
```

b) Secure use:

```
<configuration>
  <system.Web>
    <trace enabled="false" localOnly="true"/>
  </system.Web>
</configuration>
```

10. Does Web.config file case-sensitive?

Answer: Yes, Web.config file is case-sensitive.

11. How many type of modes we can say for <authentication> element in Web.config?

Answer: Configures ASP.NET authentication support. This element can be declared only at the machine, site, or application level.

Mode:

- A. Windows
- B. Forms
- C. Passport
- D. None

Example:

```
<authentication mode="Forms">  
  <forms loginUrl="~/Account/Login" timeout="2880" defaultUrl="~/ />  
</authentication>
```

12. Can one directory contain multiple Web.config files?

Answer: No. One directory can contain only one file.

13. Leaving cookieless Session State Enabled is good practice?

Answer: Allowing session storing in the page URLs is dangerous because side effect of making those applications much more vulnerable to session hijacking attacks.

Secure Use:

```
<configuration>  
  <system.Web>  
    <sessionState cookieless="UseCookies"/>  
  </system.Web>  
</configuration>
```

14. Can you tell the location of the root Web.config file from which all Web.config file inherit?

Answer: All the Web.config files inherit the root Web.config file available at the following location
systemroot\Microsoft.NET\Framework\versionNumber\CONFIG\Web.config

15. What is the root tag of Web.config file?

Answer: For Web.config <configuration> tag is the root element under which it has all the remaining sub elements.

16. For which purpose you use <appSettings> tag?

Answer: <appSettings> helps us to store information like connectionstring, url, path, port numbers, custom key values. Also you can add custom setting values for application i.e. Timeout for transactions.

Example:

```
<add key="loginUrl" value="~/Account/LogOn" />
```

```
<add key="TimeOut" value="1000" />
```

17. Explain <httpHandlers> tag in Web.config?

Answer: Maps incoming requests to the appropriate handler according to the URL and the HTTP verb that is specified in the request.

ASP.NET HTTPHandlers are responsible for intercepting requests made to your ASP.NET web application server. They run as processes in response to a request made to the ASP.NET Site.

```
<httpHandlers>
    <add.../>
    <remove.../>
    <clear/>
</httpHandlers>
```

18. What are the default HTTP handlers available in ASP.NET?

Answer: ASP.NET offers a few default HTTP handlers:

1. Page Handler (.aspx): handles Web pages
2. User Control Handler (.ascx): handles Web user control pages
3. Web Service Handler (.asmx): handles Web service pages
4. Trace Handler (trace.axd): handles trace functionality

19. Web.config file is stored in which form?

Answer: Web.config files are stored in XML format.

20. What is the difference between Web.config and Machine.config?

Answer: Main difference between Web.config and Machine.config, Web.config file is specific to a web application whereas Machine.config is specific to a machine or server. The settings made in the Web.config file are applied to that particular web application only whereas the settings of Machine.config file are applied to the whole asp.net application.

21. What is the difference between Web.config and App.config?

Answer: App.config is used for Windows Forms, Windows Services, Console Apps and WPF applications. For standalone application you can use App.config, for web application you can use Web.config.

22. In what scenarios will a project have two Web.config files?

Answer: If your project is divided into modules and modules are hierarchically connected, in that case you there are multiple Web.config files used.

Example:

Financial inventory web application has modules Sales, Accounts, Administration for all modules you can use separate Web.config.

23. What is "ConfigSource" attribute in "appSettings" section?

Answer: ConfigSource is used for linking the Web.config to another file.

a) To the project add a file "AppSettings.xml"

```
<appSettings>
  <add key="k1" value="v11"/>
  <add key="k2" value="v2"/>
</appSettings>
```

b) In Web.config:

```
<appSettings configSource="AppSettings.xml"/>
```

Advantage: Even if AppSettings.xml is modified the existing AppDomain is not abandoned.