ASP.NET Caching Job Interview Questions And Answers

Interview Questions Answers

https://interviewquestionsanswers.org/

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on ASP.NET Caching will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit ASP.NET Caching Interview Questions And Answers to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in ASP.NET Caching category. To ensure quality, each submission is checked by our team, before it becomes live. This ASP.NET Caching Interview preparation PDF was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
https://twitter.com/InterviewQA

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
Support@InterviewQuestionsAnswers.ORG

ASP.NET Caching Interview Questions And Answers Guide.

Question - 1:

Do you know using SQL Cache Invalidation?

Ans:

The Cache API introduced with ASP.NET 1.x was a powerful feature that can be immensely useful in increasing the performance of a web application. The Cache API also allows you to invalidate items in the cache based on some predefined conditions, such as change in an XML file, change in another cache item, and so on. Using this feature, you can remove or invalidate an item from the cache when the data or another cached item changes. However, the Cache API in ASP.NET 1.x versions did not provide a mechanism to invalidate an item in the cache when data in a SQL Server database changed. This is a very common capability that many web applications require. Now with ASP.NET 2.0, Microsoft has introduced a new cache invalidation mechanism that works with SQL Server as well. Using this new capability, you can invalidate an item in the Cache object whenever the data in a SQL Server database changes. This built-in cache invalidation mechanism works with SQL Server 7.0 and above. However, with SQL Server 7.0 and 2000, only table-level cache invalidation mechanism is supported. The next release of SQL Server (named SQL Server 2005) will also feature a row-level cache invalidation mechanism, providing a finer level of accuracy over the cached data. To enable the SQL Server-based cache invalidation mechanism, you need to do the following:

Add a <caching> element to the Web.config file, and specify the polling time and the connection string information.

Enable SQL cache invalidation at the database and table levels by using either the aspnet_regsql utility or the SqlCacheDependencyAdmin class. This is not required if you are using SQL Server 2005 as your database.

Specify the SqlCacheDependency attribute in the SqlDataSource control.

That's all you need to do to leverage SQL Server cache invalidation from your ASP.NET pages.

View All Answers

Question - 2:

Do you know about caching with the DataSource Controls?

Ans:

The DataSource controls enable you to cache database data while connecting a .NET application to a database. The DataSource control provides various properties, such as EnableCaching, which you can use to automatically cache the data represented by a DataSource control. The syntax to cache a database table in a memory for 120 seconds is:

```
<asp:SqlDataSource ID="SqlDataSource1" EnableCaching="True" CacheDuration="120"
ConnectionString="Server=localhost;database=AdventureWorks;uid=user;pwd=word;"
SelectCommand="SELECT * FROM Production.ProductCategory" Runat="server"/>
```

The above syntax caches a database table, ProductCategory, by setting the EnableCaching property of the DataSource control to True. The CacheDuration property of the DataSource control specifies the time, in seconds, for caching the data before it is updated in a database containing the ProductCategory table. The value of the Time parameter is set to 120 to cache data for two minutes.

View All Answers

Question - 3:

Do you know Caching Feature?

Ans:

Caching is defined as temporary storage of data for faster retrieval on subsequent requests. In ASP .NET 2.0, the caching support is integrated with the DataSource controls to cache data in a web page. ASP.NET 2.0 also now includes automatic database server cache invalidation. This powerful and easy-to-use feature allows developers to aggressively output cache database-driven page and partial page content within a site, and have ASP.NET automatically invalidate these cache entries and refresh the content whenever the back-end database changes. ASP .NET 2.0 also introduces the Substitution control, which allows you to link dynamic and cached content in a web page.

View All Answers

Question - 4:

What is ASP.NET Caching?

Ans:

Caching technique allows to store/cache page output or application data on the client. The cached information is used to serve subsequent requests that avoid the overhead of recreating the same information. This enhances performance when same information is requested many times by the user.

View All Answers

Question - 5:

What is Data Caching?

Ans:

Data Caching is implemented by using Cache object to store and quick retrieval of application data.
Cache object is just like application object which can be access anywhere in the application.
The lifetime of the cache is equivalent to the lifetime of the application. .

View All Answers

Question - 6:

What is Page Fragment Caching?

Ans:

This technique is used to store part of a Web form response in memory by caching a user control.

View All Answers

Question - 7:

Explain Page Output Caching?

Ans:

This type of caching is implemented by placing OutputCache directive at the top of the .aspx page at design time.
For example:
<%@OutputCache Duration= "30" VaryByParam= "DepartmentId"%>
The duration parameter specifies for how long the page would be in cache and the VaryByParam parameter is used to cache different version of the page.
The VaryByParam parameter is useful when we require caching a page based on certain criteria.

View All Answers

Question - 8:

Explain advantages of Caching?

Ans:

It increases performance of the application by serving user with cached output.
It decreases server round trips for fetching data from database by persisting data in the memory.
It greatly reduces overhead from server resources.

View All Answers

Question - 9:

Explain Transparent caching with AOP?

Ans:

Often, the objects that compose applications perform the same operations with the same arguments and obtain the same results. Sometimes, these operations are costly in terms of CPU usage, or may be there is a lot of I/O going on while executing those operations.
To get better results in terms of speed and resources used, it's suggested to use a cache. We can store in it the results corresponding to the methods' invocations as key-value pairs: method and arguments as key and return object as value.
Once you decide to use a cache you're just halfway. In fact, you must decide which part of the application is going to use the cache. Let's think about a web application backed by a database. Such a web application usually involves Data Access Objects (DAOs), which access the relational database. Such objects are usually a bottleneck in the application as there is a lot of I/O going on. In other words, a cache can be used there.
The cache can also be used by the business layer that has already aggregated and elaborated data retrieved from repositories, or it can be used by the presentation layer putting formatted presentation templates in the cache, or even by the authentication system that keeps roles according to an authenticated username.
There are almost no limits as to how you can optimize an application and make it faster. The only price you pay is having RAM to dedicate the objects that are to be kept in memory, besides paying attention to the rules on how to manage the life of the objects in cache.
After these preliminary remarks, using a cache could seem common and obvious. A cache essentially acts as a hash into which key-value pairs are put. The keys are used to retrieve objects from the cache. Caching usually has confi guration parameters that allow you to change its behavior.
Now let's have a look at an example with ehcache (http://ehcache.sourceforge. net). First of all let's confi gure it with the name methodCache so that we have at the most 1000 objects. The objects are inactive for a maximum of fi ve minutes, with a maximum life of 10 minutes. If the objects count is over 1000, ehcache saves them on the fi lesystem, in java.io.tmpdir.

```
<ehcache>
    ...
    <diskStore path="java.io.tmpdir"/>
    ....
    <defaultCache
        maxElementsInMemory="10000"
        eternal="false"
        timeToIdleSeconds="120"
        timeToLiveSeconds="120"
        overflowToDisk="true"
       diskPersistent="false"
       diskExpiryThreadIntervalSeconds="120"
    />
        ...
    <cache name="methodCache"
        maxElementsInMemory="1000"
        eternal="false"
        overflowToDisk="false"
        timeToIdleSeconds="300"
        timeToLiveSeconds="600"
     />
</ehcache>
```

Now let's create a CacheAspect. Let's defi ne the cacheObject to which the ProceedingJoinPoint is passed. Let's recover an unambiguous key from the ProceedingJoinPoint with the method getCacheKey. We will use this key to put the objects into the cache and to recover them.
Once we have obtained the key, we ask to cache the Element with the instruction cache.get(cacheKey). The Element has to be evaluated because it may be null if the cache didn't fi nd an Element with the passed cacheKey.
If the Element is null, advice invokes the method proceed(), and puts in the cache the Element with the key corresponding to the invocation. Otherwise, if the Element recovered from the cache is not null, the method isn't invoked on the target class, and the value taken from the cache is returned to the caller.

```
package org.springaop.chapter.five.cache;
import it.springaop.utils.Constants;
import net.sf.ehcache.Cache;
import net.sf.ehcache.Element;
import org.apache.log4j.Logger;
import org.aspectj.lang.ProceedingJoinPoint;
public class CacheAspect {
public Object cacheObject(ProceedingJoinPoint pjp) throws Throwable
{
     Object result;
     String cacheKey = getCacheKey(pjp);
     Element element = (Element) cache.get(cacheKey);
     logger.info(new StringBuilder("CacheAspect invoke:").append("
 get:") .append(cacheKey).append(" value:").append(element). toString());
    if (element == null) {
         result = pjp.proceed();
         element = new Element(cacheKey, result);
         cache.put(element);
         logger.info(new StringBuilder("
 put:").append(cacheKey). append( " value:").append(result).toString());
    }
    return element.getValue();
}
public void flush() {
    cache.flush();
}
private String getCacheKey(ProceedingJoinPoint pjp) {
    String targetName = pjp.getTarget().getClass().getSimpleName();
    String methodName = pjp.getSignature().getName();
    Object[] arguments = pjp.getArgs();
     StringBuilder sb = new StringBuilder();
    sb.append(targetName).append(".").append(methodName);
     if ((arguments != null) && (arguments.length != 0)) {
        for (int i = 0; i < arguments.length; i++) {
            sb.append(".").append(arguments[i]);
        }
       }
       return sb.toString();
}
public void setCache(Cache cache) {
    this.cache = cache;
}
   private Cache cache;
   private Logger logger = Logger.getLogger(Constants.LOG_NAME);
}
```

Here is applicationContext.xml :

```
<beans xmlns=Â»http://www.springframework.org/schema/beansÂ»
xmlns:xsi=Â»http://www.w3.org/2001/XMLSchema-instanceÂ»
xmlns:aop=Â»http://www.springframework.org/schema/aopÂ»
xsi:schemaLocation=Â»http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-2.5.xsdÂ»> ...
<bean id="rockerCacheAspect" class="org.springaop.chapter.five. cache.CacheAspect" >
<property name="cache">
    <bean id="bandCache" parent="cache">
        <property name="cacheName" value="methodCache" />
    </bean>
</property>
</bean>
<!-- CACHE config -->
<bean id="cache" abstract="true" class="org.springframework.cache.ehcache.EhCacheFactoryBean">
<property name="cacheManager" ref="cacheManager" />
</bean>,
<bean id="cacheManager"
class="org.springframework.cache.ehcache.
EhCacheManagerFactoryBean">
<property name="configLocation" value="classpath:org/springaop/
chapter/five/cache/ehcache.xml" />
</bean>
...
</beans>
```

The idea about the caching aspect is to avoid repetition in our code base and have a consistent strategy for identifying objects (for example using the hash code of an object) so as to prevent objects from ending up in the cache twice.
Employing an around advice, we can use the cache to make the method invocations return the cached result of a previous invocation of the same method in a totally transparent way. In fact, to the methods of the classes defi ned in the interception rules in pointcuts will be given back the return values drawn from the cache or, if these are not present, they will be invoked and inserted in the cache. In this way, the classes and methods don't have any knowledge of obtaining values retrieved

from the cache.

View All Answers

Microsoft .Net Technologies Most Popular & Related Interview Guides

1 : **MSF Interview Questions and Answers.**

2 : .Net Architecture Interview Questions and Answers.

3 : ASP.Net MVC Interview Questions and Answers.

4 : Entity Framework Interview Questions and Answers.

5 : C# (Sharp) Programming Language Interview Questions and Answers.

6 : VB .Net Interview Questions and Answers.

7 : ADO.NET Interview Questions and Answers.

8 : WCF (Windows Communication Foundation) Interview Questions and Answers.

9 : Crystal Reports Interview Questions and Answers.

10 : .Net Database Interview Questions and Answers.

Follow us on FaceBook
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter
https://twitter.com/InterviewQA

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG