Data Caching

By: Mosh Hamedani

Use Data Caching with Expensive Queries

In the demo, I stored the list of genres in the cache. Getting the list of genres would issue a "**SELECT * FROM Genres**" query to the database, and given that this is a simple and fast query, caching the result would just waste server's resources.

Use data caching for complex queries over large tables that take several seconds to execute. This way, you can argue that using additional memory on the server can be better (but not necessarily) than querying your database several times. You need to profile **before and after** your optimization to ensure your assumptions are correct and are not based on some theory you read in a book or tutorial.

MemoryCache.Add()

If you want to have more control over the objects you put in their cache, it's better to use the Add method of MemoryCache class.

```
MemoryCache.Default.Add(
    new CacheItem("Key", value),
    cacheItemPolicy);
```

As you see, the second argument to this method is a **cacheItemPolicy** object. With this object you can set the expiration date/time (both absolute and sliding) and you can also register callbacks to be called when the item is removed from the cache.

You can read more about this class on MSDN:

https://msdn.microsoft.com/en-us/library/system.runtime.caching.memorycache.add(v=vs.110).aspx