

2017年3月 (13)
2017年2月 (12)
2017年1月 (9)
2016年12月 (10)
2016年11月 (5)
2016年10月 (10)
2016年9月 (2)
2016年8月 (8)
2016年7月 (2)
2016年6月 (2)
2016年5月 (2)
2016年3月 (1)
2016年2月 (1)
2015年10月 (1)
2015年9月 (2)
2015年8月 (5)
2015年7月 (4)
2015年6月 (5)
2015年4月 (7)
2015年3月 (13)
2015年2月 (1)
2015年1月 (3)
2014年12月 (3)
2014年10月 (1)
2014年9月 (1)
2014年8月 (6)
2014年6月 (2)
2014年5月 (4)
2014年4月 (5)
2014年3月 (5)
2014年2月 (6)
2013年12月 (4)
2013年11月 (7)
2013年10月 (10)
2013年9月 (30)
2013年8月 (11)
2013年7月 (4)
2013年6月 (1)
2013年5月 (5)
2013年4月 (6)
2013年3月 (5)
2013年2月 (7)
2013年1月 (14)

收藏

Android 高手进阶

最新评论

1. Re:WebService 与 Socket 区别
可以不可以这样理解。socket更底层更原始一些（传输层）。而webService是应用层，方便但是没底层的灵活。
--梦中山河
2. Re:java单点登录系统CAS的简单使用
好文，感谢
--woncode
3. Re:漫谈单点登录（SSO）（淘宝天猫）（转载）
楼主，在V4中，你说T从SSO中复制一份cookie实现登录，也就是说T是自己保存的cookie，那你如何保证T、W、SSO三份cookie有效期一致呢？当SSO的cookie被我手动删除了，T和W怎.....
--xiebruce
4. Re:漫谈单点登录（SSO）（淘宝天猫）（转载）
@一斤奋斗引用SSO_V4 中，T 如何拿到W 的 cookie 的?如果没拿到,那么 SSO 如何知道来访问的是谁?T是先检查自己有没有cookie，没有就跳转到SSO，SSO是有cookie的.....
--Blackheart
5. Re:在.net中读写config文件的各种方法（转载）
博主，原文地址还有么
--乌鸦4ay

}

准备工作已经完成

我们先写一个方法,添加数据库依赖

```
void AddSqlDependency(string strCon, string strSql, OnChangeEventHandler sqlDep_OnChange)
{
    try
    {
        using (SqlConnection conn = new SqlConnection(strCon))
        {
            SqlCommand comm = new SqlCommand(strSql, conn);
            SqlDependency sqlDep = new SqlDependency(comm);
            sqlDep.OnChange += sqlDep_OnChange;
            if (conn.State == ConnectionState.Closed) conn.Open();
            comm.ExecuteNonQuery();
        }
    }
    catch (Exception ex)
    {
        LogUtility.Add(ex);
    }
}
```

//上面这个方法是告诉数据库,当你指定表数据改变,要移出缓存

我们现在可以来添加了

```
MyObject obj= HttpRuntime.Cache["cacheName"] as MyObject;
if (null == obj)
{
    try
    {
        obj= GetObj(...);
    }
    catch (Exception ex)
    {
        LogUtility.Add(ex);
        obj= null;
    }
    if (null != obj)
    {
        AddSqlDependency(strCon, "select id from dbo.tableName;select id1 from dbo.tableName1"
        delegate(object sender, SqlNotificationEventArgs e)
        {
            //do something

            HttpRuntime.Cache.Remove("cacheName");
        });
        HttpRuntime.Cache.Insert("cacheName", obj);
    }
}
```

上面SQL语句中用到的表,只要这些表中的任何数据有改动,数据库都会通知,这时缓存会移动,select的字段和Cache没有关系,只有表名有关系,所有你要选择最小的字段.很多时候,为了这个字段,在设计表的时候都多加一个最小的依赖列.

NOTE:任何Cache都可能丢失,使用前一定要做必要的检查,如:

```
MyObject obj=HttpContext.Current.Cache("CacheName") as MyObject;

if(null==obj)
```

阅读排行榜

1. Eclipse安装SVN插件(196970)
2. java byte 16进制转换(16655)
3. listView下拉刷新加载数据(16568)
4. Android-Universal-Image-Loader 图片异步加载类库的使用（超详细配置）(15378)
5. Eclipse与Android源码中ProGuard工具的使用（代码混淆）(14411)

评论排行榜

1. 漫谈单点登录（SSO）(淘宝天猫)（转载）(10)
2. Eclipse安装SVN插件(5)
3. 在.net中读写config文件的各种方法（自定义config节点）(3)
4. HttpContext.Current.Cache 过期时间(2)
5. listView下拉刷新加载数据(2)

推荐排行榜

1. Eclipse安装SVN插件(8)
2. HttpContext.Current.Cache 过期时间(3)
3. Activity与服务通信(2)
4. Razor视图引擎 语法学习（一）(2)
5. WCF如何使用X509证书（安装和错误）（二）(2)

```
{
obj=.....

HttpContext.Current.Cache.Insert("CacheName",obj);
}
Cache用法之页面声明
```

```
<%@ outputCache
Duration="#ofseconds"
Location="Any|Client|Downstream|Server|None"
VaryByControl="ControlName"
VaryByCustom="browser|customstring"
VaryByHeader="headers"
VaryByParam="Parametername" %>
```

Cache用法之代码控制

HttpCachePolicy类是专门用来控件Cache的，可以用Response.Cahce来访问这个类的实例

```
Response.Cache.SetExpires(DateTime.Now.AddSeceonds(10));
Response.Cache.SetCacheability(HttpCacheability.Public);
Response.Cache.SetValidUnitlExpires(true);
```

-----以上都是缓存页面的，下面是缓存数据的-----

Cache 类的生存周期等于应用程序的生命周期

三种用法

```
1: 存: Cache["key"] = MyData;取:
MyData = Cache["key"];
if(MyData != null)
use(MyData);
```

此法存入Cache的数据生命周期等于应用程序生命周期，不支持清除、过期、依赖性等功能。

2:存:

```
Cache.Insert(
string key,
object value,
CacheDependency dependencies,//依赖，设置缓存有效的依赖性，比如设置和一个文件相关，文件一变，就失效
DateTime absoluteExpiration, //设置固定的过期时间
TimeSpan slidingExpiration, //设置最后一次访问后多长时间过期
CachePriority priority, //设置内存不足，缓存自动清除时，缓存的重要性，可不可以清除
CacheItemRemovedCallback onRemoveCallback // 设置在清除时引发的事件
)
```

Example:

```
Cache.Insert("Mydata",MyData,new
Caching.CacheDependency(Server.MapPath("Mydata.XML")));//设置有效性和一个文件有关
Cache.Insert("Mydata",myData,null,DateTime.Now.AddDays(1),Cache.NoSlidingExpiration);//
两种过期时间设了其中一种，另一种要设为0,用NoAbsolute(Sliding)Expiration枚举
Cache.Insert("MyData",myData,null,Cache.NoAbsoluteExpiration,TimeSpan.FromMinutes(10
));//不能过一年不能小于0
Cache.Insert("MyData",myData,null,Cache.NoAbsoluteExpiration,TimeSpan.FromMinutes(10
,
Caching.CacheItemPriority.NotRemovable,null);
// AboveNormal|BelowNormal|Default|High|Low|Normal|NotRemovable
```

```
public void RemovedCallback(string key,object value,CacheItemRemovedReason reason)
{
if(reason == CacheItemRemovedReason.DependencyChanged)
Response.Write("文件变了，快去看看");
}
Cache.Insert("Mydata",MyData,new
Caching.CacheDependency(Server.MapPath("Mydata.XML"),
DateTime.Now.AddDays(1),Cache.NoSlidingExpiration,CacheItemPriority.High,
```

HttpContext.Current.Cache 过期时间 - V青山绿水 - 博客园

```
new CacheItemRemovedCallback(this.RemovedCallback));
```

清除就可以用Cache.Remove("key");方法

3:
Cache.Add方法，用法和Insert差不多，区别在于Add碰到该key原来有赋过值会失败，Insert则不会，而会替换原有值；Add会返回被缓存数据项，Insert不会

分类: [C#.Net](#)

好文要顶

关注我

收藏该文

[V青山绿水](#)
[关注 - 0](#)
[粉丝 - 43](#)
[+加关注](#)

3

0

« 上一篇: [依赖倒置原则](#)
» 下一篇: [单点登录（SSO）（原创）](#)

posted @ 2013-08-09 15:56 V青山绿水 阅读(7650) 评论(2) 编辑 收藏

发表评论

#1楼 2014-02-08 11:17 | 待需而动

已收藏

支持(0) 反对(0)

#2楼 2015-12-23 13:02 | xchsp

HttpContext.Current.Cache 过期时间，谢谢分享。

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 最新IT新闻：
- 微软发布Windows Server 预览版Build 16267
 - 进展顺利：NASA已对第二艘猎户座乘员舱进行首次上电测试
 - Supercell宣布将与Facebook合作，《部落冲突》也要推出AR版了
 - 红帽对于甲骨文将Java EE移交开源组织的看法
 - 微软披露Project Brainwave实时人工智能平台的更多细节
- » 更多新闻...

- 最新知识库文章：
- 做到这一点，你也可以成为优秀的程序员
 - 写给立志做码农的大学生
 - 架构腐化之谜
 - 学会思考，而不只是编程
 - 编写Shell脚本的最佳实践
- » 更多知识库文章...