# V青山绿水

博客园 首页 新闻 新随笔 联系 管理 订阅

> Ŧī. 六 5

11 19

12

随笔-299 文章-0 评论-27

### HttpContext.Current.Cache 过期时间

昵称:V青山绿水 园龄: 4年7个月 粉丝: 43 关注: 0 +加关注

7

13 14 15

4

21 22

8 9

5 6 7 8 9

30 31

20

27 28 29

我的随笔 我的评论

我的参与 最新评论

我的标签 更多链接

ASP C#.Net(34)

IOS(12) Java(24) JMS(3)

JS(3) Linux(1)

Log4j(2)

Maven(1) MSSQL数据库(5)

MyBatis(3) MyCat(6) mysql读写分离(10)

MYSOL数据库(6) PHP(4)

RabbitMQ(4)

spring mvc(9) SSO(9)

Web Service(7) 负载均衡(6) 高并发(3)

Redis(24) SEO

WCF(6)

工具(11) 其他(2)

设计模式(1)

2017年4月 (9)

android 教程(97) Android实例(7)

为了更快的读取数据,我们一般会把常用到的数据加载到Cache中

在.NET中,Cache的存在可以依赖多中方式,主要用到HttpContext.Current.Cache类

在这里,我主要写几种依赖方式

1:不依赖任何条件

HttpContext.Current.Cache.Insert(string cacheName,object obj)

理论上是Cache会永久保存,但是当服务器重新启动,内存紧张的时候也会丢失.

找找看 谷歌搜索

2017年8月

2 3 4

16 17 18

30

搜索

常用链接

随笔分类

<u>23</u>

10

24 25 26

31 1 2

> 2:HttpContext.Current.Cache.Insert(string key, object value, CacheDependency dependencies, DateTime absoluteExpiration, TimeSpan slidingExpiration);

//CacheDependency缓存依赖项,absoluteExpiration绝对过期时间,slidingExpiration最后一次访问的 时间隔

//我们主要讲两种Cache依赖

2.1:文件依赖,so simple//只要文件改动,Cache移出

HttpContext.Current.Cache.Insert(cacheName,ojb, new System.Web.Caching.CacheDependency(FilePath));

2.2:SqlServer数据库依赖//我这里是SQL2005

首先看数据库通知是否可用,记得一定要开启通知

Select DATABASEpRoPERTYEX('数据库名','IsBrokerEnabled');如果值为1表示可用

alter database Game176Admin set ENABLE\_BROKER;//开启 alter database Game176Admin set DISABLE\_BROKER;//关闭

在Global.asax文件中,我们在应用程序开始和结束时要做一些事情

void Application\_Start(object sender, EventArgs e)

try

// 在应用程序启动时运行的代码

System.Data.SqlClient.SqlDependency.Start(string strCon);//开启

} catch { }

void Application\_End(object sender, EventArgs e)

{ try

// 在应用程序关闭时运行的代码

随笔档案

2017年8月 (4) System.Data.SqlClient.SqlDependency.Stop(string strCon); 2017年7月 (11) }

2017年6月 (10) 2017年5月 (11) catch { }

http://www.cnblogs.com/ruiati/p/3248451.html

```
2017年3月 (13)
                                    }
2017年2月 (12)
2017年1月 (9)
                                    准备工作已经完成
2016年12月 (10)
2016年11月 (5)
                                    我们先写一个方法,添加数据库依赖
2016年10月 (10)
2016年9月 (2)
2016年8月 (8)
                                    void AddSqlDependency(string strCon, string strSql, OnChangeEventHandler
2016年7月 (2)
                                    sqlDep OnChange)
2016年6月 (2)
                                    {
2016年5月 (2)
                                    try
2016年3月 (1)
                                    {
2016年2月 (1)
                                    using (SqlConnection conn = new SqlConnection(strCon))
2015年10月(1)
2015年9月 (2)
2015年8月 (5)
                                    SqlCommand comm = new SqlCommand(strSql, conn);
2015年7月 (4)
                                    SqlDependency sqlDep = new SqlDependency(comm);
2015年6月 (5)
                                    sqlDep.OnChange += sqlDep OnChange;
2015年4月 (7)
                                    if (conn.State == ConnectionState.Closed) conn.Open();
2015年3月 (13)
2015年2月 (1)
                                    comm.ExecuteNonQuery();
2015年1月 (3)
                                    }
2014年12月 (3)
                                    }
2014年10月 (1)
                                    catch (Exception ex)
2014年9月 (1)
2014年8月 (6)
                                    LogUtility.Add(ex);
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)
                                    //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
2013年12月 (4)
2013年11月 (7)
                                    我们现在可以来添加了
2013年10月 (10)
2013年9月 (30)
2013年8月 (11)
                                    MyObject obj= HttpRuntime.Cache["cacheName"] as MyObject;
2013年7月 (4)
                                    if (null == obj)
2013年6月 (1)
                                    {
2013年5月 (5)
                                    try
2013年4月 (6)
                                    {
2013年3月(5)
2013年2月 (7)
                                    obj = GetObj(...);
2013年1月 (14)
                                    }
                                    catch (Exception ex)
            收藏
Android 高手进阶
                                    LogUtility.Add(ex);
                                    obj= null;
           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
                                    if (null != obj)
始一些(传输层)。 而webservice是应用
层,方便但是没底层的灵活。
                                    AddSqlDependency(strCon, "select id from dbo.tableName; select id1 from dbo.tableName1"
                     --梦中山河
                                    delegate(object sender, SqlNotificationEventArgs e)
2. Re:java单点登录系统CAS的简单使用
好文,感谢
                    --woncode
                                    //do something
3. Re:漫谈单点登录(SSO)(淘宝天
猫) (转载)
                                    HttpRuntime.Cache.Remove("cacheName");
楼主,在V4中,你说T从SSO中复制一份co
okie实现登录,也就是说T是自己保存的co
                                    HttpRuntime.Cache.Insert("cacheName", obj);
okie, 那你如何保证T、W、SSO三份cooki
e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
```

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,在设计表的时候都分 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)
```

```
http://www.cnblogs.com/ruiati/p/3248451.html
```

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coo

5. Re:在.net中读写config文件的各种方

--Blackheart

--乌鸦4av

kie的......

法(转载)

博主,原文地址还有么

2017/8/23

#### 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)(15 378)
- 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Service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(HttpCacheablility.Public);
Response.Cache.SetValidUnitlExpires(true);
    ------以上都是缓存页面的,下面是缓存数据的-----以上都是缓存页面的,下面是缓存数据的------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 absoluteExpireation, //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.MapPah("Mydata.XML")));//设置有效性和一个文件有关
Cache.Insert("Mydata",myData,null,DateTime.Now.AddDays(1),Cache.NoSlidingExpiratin);//
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.MapPah("Mydata.XML"),
DateTime.Now.AddDays(1),Cache.NoSlidingExpiration,CacheItemPriority.High,
```

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



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

» 下一篇: 单点登录(SSO)(原创)

已收藏

支持(0) 反对

#2楼 2015-12-23 13:02 | xchsp

HttpContext.Current.Cache 过期时间,谢谢分享。

支持(0) 反对

刷新评论 刷新页面 返回顶

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...

Copyright ©2017 V青山绿水