# Memcached + Tomcat7（2台）实现session共享

1. 下载安装Memcache

测试memcache是否连通

telnet localhost 11211（默认安装端口）

测试memcache是否工作

set abc 0 0 5

12345

get abc

1. 下载需要的jar包，放在tomcat7 lib目录下













1. 配置tomcat7

Tomcat7/Conf/Server.xml配置添加

<Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcat1">

omcat7/Conf/Context.xml配置添加

<Manager className="de.javakaffee.web.msm.MemcachedBackupSessionManager"

memcachedNodes="n1:localhost:11211"

lockingMode="auto"

sticky="false"

requestUriIgnorePattern= ".\*\.(png|gif|jpg|css|js)$"

sessionBackupAsync= "false"

sessionBackupTimeout= "100"

transcoderFactoryClass="de.javakaffee.web.msm.serializer.xstream.XStreamTranscoderFactory"

/>

1. Tomcat7中放入测试包



1. 启动tomcat，进行测试访问

黏性Session

安装在Tomcat上的MSM使用本机内存保存session，和StandardManager一样。另外，当一个请求结束时，session会被送回Memcached进行备份。当下一次请求开始时，本地Session可用，直接服务，请求结束后，session又被送回Memcached备份。

当集群中的一个Tomcat挂掉，下一次请求会被路由到其他Tomcat上。负责处理此此请求的Tomcat并不清楚Session的信息。此时它会从Memcached查找该Session，更新该Session并将其保存在本机内容。此次请求结束，session被修改，送回Memcached备份。

注意：当使用多台tomcat时，一定要使用non-sticky（非粘性）模式

**Manager标签属性说明  
1.className  必须**类名：de.javakaffee.web.msm.MemcachedBackupSessionManager

**2.memcachedNodes  必须**memcached节点：此属性应该包含所有运行的 memcached节点或者membase bucket的uri地址，每一个memcached节点的属性定义格式为<id>:<host>:<port>， 多个节点定义直接使用空格或者逗号分隔，形如：memcachedNodes="n1:app01:11211,n2:app02:11211"，如果只 有单个的memcached节点，则<id>是可选项，只需配置<host>:<port>即可，形 如：memcachedNodes="localhost:11211"。  
如果我们配置的是membase，那么从1.6.0版本开始，我们可以配置指定一个或者多个membase bucket uris，形如：<http://host1:8091/pools,http://host2:8091/pools>。Bucket 名称和密码通过属性username,password来定义。membase buckets连接需要遵循memcached协议，传输数据通过二进制流方式。

**3.failoverNodes 可选项**故障转移节点：可选项，对非黏性session不可用，属性必须包含memcached节点集群的所有ids。节点id之间用空格或者逗号分隔。

**4.username 可选项**从1.6.0版开始使用，并且是可选的。用来进行membase bucket或者SASL验证，密码可以为空。

**5.password 可选项**从1.6.0版开始使用，并且是可选的。用来进行membase bucket或者SASL验证，密码可以为空。

**6.memcachedProtocol    可选项**定义memcached协议，默认使用text文本，出属性指明memcached使用的存储协议。只支持text或者binary。

**7.sticky    可选项**定义session方式为黏性或非黏性，默认为true，多个tomcat时需使用非黏性

**8.lockingMode    可选项**只有非黏性session才使用，默认值为none  
none: 从不对session进行锁定  
all: session将一直被锁定，知道请求结束  
auto: 对于只读请求，session将不会被锁定，如果是非只读请求，则session会被锁定  
uriPattern:<regexp>: 通过正则表达式的方式来对请求uri以及查询字符串进行匹配，只有匹配上的才会被锁定。  
**9.requestUriIgnorePattern   可选项**

此属性是那些不能改备份Session的请求的正则表达式。如果像css,javascript,图片等静态文件被同一个Tomcat和同一个应用 上下文来提供，这些请求也会通过memcached-session-manager。但是这些请求在一个http会话中几乎没什么改变，所以他们没必要 触发Session备份。所以那些静态文件没必要触发Session备份，你就可以使用此属性定义。此属性必须符合java regex正则规范。  
    如：".\*\.(png|gif|jpg|css|js)$"

**10.sessionBackupAsync   可选项**指定Session是否应该被异步保存到Memcached中。 如果被设置为true，backupThreadCount设置起作用，如果设置false，通过sessionBackupTimeout设置的过期时间起作用。

**11.backupThreadCount    可选项** 用来异步保存Session的线程数，(如果sessionBackupAsync="true")。默认值为cup的内核数。

**12.sessionBackupTimeout    可选项** 设置备份一个Session所用的时间，如果操作超过时间那么保存失败。此属性只在sessionBackupAsync="false"是起作用。默认100毫秒

**13.operationTimeout    可选项**从1.6.0版开始使用， 默认值为1000

**14.sessionAttributeFilter    可选项** 此属性是用来控制Session 中的那个属性值保存到Memcached中的正则表达式。郑则表达式被用来匹配Session中属性名称。如 sessionAttributeFilter="^(userName|sessionHistory)$" 指定了只有"userName"和"sessionHistory"属性保存到Memcached中。依赖于选择的序列化策略。

**15.transcoderFactoryClass    可选项** 此属性值是创建序列化和反序列化 保存到Memcached中的Session的编码转换器的工厂类名。这个指定的类必须实现了 de.javakaffee.web.msm.TranscoderFactory和提供一个无参的构造方法。例如其他的有效的实现在其他 packages/jars中提供如：msm-kryo-serializer,msm-xstrea-serializer和msm- javolution-serializer.  
默认为 de.javakaffee.web.msm.JavaSerializationTranscoderFactory

**16.copyCollectionsForSerialization    可选项**默认值为false。

**17.customConverter    可选项**自己定义特殊的类注册到kryo自定义转换器中，实现序列化  
18.enableStatistics    可选项  
 用来指定是否进行统计。 默认值为true。

**19.enabled   可选项** 指定Session保存到Memcached中是否可用和是否可以通过JMX进行改变。只用于粘性Session。 默认值为true。