**[Maven项目中用Memcached实现缓存配置](http://jason61719.iteye.com/blog/1002141)**

下面记录Memcached 在Maven 中的配置。

首先介绍如何手动添加第三方类库到自己电脑的本地仓库。

首先添加Memcached 的Java 客户端到项目依赖中，如下执行：

1. 将java\_memcached-release\_2.6.6.jar 放到E 盘下面；

2. cmd 到打开命令窗口以后开始安装，运行如下指令：

E:\User>mvn install:install-file -Dfile=e:/java\_memcached-release\_2.6.6.jar -DgroupId=com.danga -DartifactId=memcached -Dversion=2.6.6 -Dpackaging=jar -DgeneratePom=true

3. 这样它就会自动在本地仓库里面加入自定义的jar 依赖。

接着在项目添加以下依赖：

pom.xml 中代码

<dependency>

<groupId>com.danga</groupId>

<artifactId>memcached</artifactId>

<version>2.6.6</version>

</dependency>

<dependency>

<groupId>commons-pool</groupId>

<artifactId>commons-pool</artifactId>

<version>1.6</version>

</dependency>

接着在项目中写Memcached 的应用类，下面是根据官方demo 修改的，仅供参考：

Java代码  [收藏代码](javascript:void())

import com.danga.MemCached.MemCachedClient;

import com.danga.MemCached.SockIOPool;

/\*\*

\* @since

\* @author Hmily chu

\*/

public class MemCached {

private static MemCachedClient cachedClient = new MemCachedClient();

static {

String[] servers = { "192.168.8.52:11211" };

// 创建一个连接池

SockIOPool pool = SockIOPool.getInstance();

// 设置缓存服务器

pool.setServers(servers);

// 设置初始化连接数，最小连接数，最大连接数以及最大处理时间

pool.setInitConn(50);

pool.setMinConn(50);

pool.setMaxConn(500);

pool.setMaxIdle(1000 \* 60 \* 60);

// 设置主线程睡眠时间，每30秒苏醒一次，维持连接池大小

pool.setMaintSleep(30);

// 关闭套接字缓存

pool.setNagle(false);

// 连接建立后的超时时间

pool.setSocketTO(3000);

// 连接建立时的超时时间

pool.setSocketConnectTO(0);

// 初始化连接池

pool.initialize();

}

protected MemCached() {

}

public static MemCachedClient getInstance() {

return cachedClient;

}

public static void main(String[] args) {

final MemCachedClient mcc = MemCached.getInstance();

for (int i = 0; i < 10; i++) {

boolean success = mcc.set("" + i, "Hello!");

String result = (String) mcc.get("" + i);

System.out.println(String.format("set( %d ): %s", i, success));

System.out.println(String.format("get( %d ): %s", i, result));

}

new Thread(){

public void run() {

for (int i = 0; i < 10; i++) {

String result = (String) mcc.get("" + i);

System.out.println(String.format("sub Thread get( %d ): %s", i, result));

}

};

}.start();

}

}