Atitit clsldr uti

//import net.sf.ehcache.util.ClassLoaderUtil;

*@SuppressWarnings*("all")

public class ClassloaderUti {

public static void main(String[] **args**) throws Exception {

// io.netty.buffer.PoolArena

// io.netty.channel.AbstractChannel

// com.lambdaworks.redis.RedisException

// com.lambdaworks.redis.RedisException

// reactor.core.scheduler.Schedulers

// FactoryConfigurationError

// io.netty.channel.group.ChannelGroup

// org.springframework.data.geo.Metric

// io.lettuce.core.AbstractRedisClient

// org.springframework.beans.factory.BeanClassLoaderAware

// org.springframework.data.redis.connection.RedisConnectionFactory

// org.springframework.dao.support.PersistenceExceptionTranslator

// com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryImpl

//,Thread.currentThread().getContextClassLoader()

// javax.security.cert.X509Certificate

for (int i=0;i<1000;i++) {

try {

ResourceBundle resource = ResourceBundle.*getBundle*("redis");

String host = resource.getString("spring.redis.host");

String pwd = resource.getString("spring.redis.password");

RedisURI RedisURI1 = RedisURI.*create*("redis://"+host+":6379");

RedisURI1.setPassword(pwd);

RedisClient client = RedisClient.*create*(RedisURI1);

// client.se

StatefulRedisConnection<String,String> connect = client.connect();

// connect.se

/\*\*

\* 同步调用

\*/

RedisCommands<String,String> commands = connect.sync();

commands.select(10);

commands.set("hello"+i,"hello world"+i);

// String str = commands.get("hello");

// System.out.println(str);

connect.close();

client.shutdown();

// foreachClsldr(i);

} catch (Exception e) {

e.printStackTrace();

}

*TimeUnit*.***SECONDS***.sleep(3);

}

*TimeUnit*.***SECONDS***.sleep(5000);

System.***out***.println("ff");

// org.xml.sax.ErrorHandler

// com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryImpl

}

private static void foreachClsldr(int **i**) throws ClassNotFoundException, NoSuchMethodException,

IllegalAccessException, InvocationTargetException, IOException {

List li\_libdir=Lists.*newArrayList*();

li\_libdir.add("D:\\libs\_junbo");

String classpath = "D:\\apiJubo\\api\\common\\target\\classes";

List li\_classDir=Lists.*newArrayList*();

li\_classDir.add(classpath);

SimpleURLClassLoader myClassLoader = (SimpleURLClassLoader) *getURLClassLoader*(li\_libdir,li\_classDir);

Class RedisUtilsCls=myClassLoader.loadClass("com.wz.common.redis.RedisUtils");

Object RedisTemplate1 = MethodUtils.*invokeStaticMethod*(RedisUtilsCls, "getRedisTemplate", 13);

MethodUtils.*invokeStaticMethod*(RedisUtilsCls,"setX","key" + **i**, "v"+**i**, 10000,RedisTemplate1);

System.***out***.println(new Date());

RedisTemplate1=null; RedisUtilsCls=null;

myClassLoader.close();myClassLoader=null;

System.*gc*();

}

private static URLClassLoader getURLClassLoader(List<String> **li\_libdir**, List<String> **li\_classDir**) {

List<URL> li\_urls=Lists.*newArrayList*();

for (String libpath : **li\_libdir**) {

File file\_directory = new File(libpath);

File[] fileNames =file\_directory.listFiles();

// URL urls[] = new URL[fileNames.length];

for (int i = 0; i < fileNames.length; i++) {

try {

li\_urls.add(fileNames[i].~~toURL~~());

} catch (MalformedURLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

for (String clsD : **li\_classDir**) {

try {

li\_urls.add(new File(clsD).~~toURL~~());

} catch (MalformedURLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

URL[] urls=li\_urls.toArray(new URL[li\_urls.size()]) ;

URLClassLoader loader = new SimpleURLClassLoader(urls,null);

//new URLClassLoader(urls,null);

return loader;

// return null;

}

public static URLClassLoader getURLClassLoader(String **jarLibDir**) {

//String pathname = "lib";

File file\_directory = new File(**jarLibDir**);

File[] fileNames =file\_directory.listFiles();

URL urls[] = new URL[fileNames.length];

for (int i = 0; i < fileNames.length; i++) {

try {

urls[i] = fileNames[i].~~toURL~~();

} catch (MalformedURLException e) {

throw new RuntimeException("加载lib目录下jar文件出错！"+fileNames[i].getAbsolutePath(), e);

}

}

URLClassLoader loader = new URLClassLoader(urls,null);

return loader;

}