Atitit 得到所有类的继承树木

目录

[1.1. 遍历jar文件 得到路径与类名 classload 加载 1](#_Toc5349)

[1.2. 使用Class类中的getSuperClass()方法能够得到一个类的父类 1](#_Toc28457)

[1.3. code 1](#_Toc22179)

## 遍历jar文件 得到路径与类名 classload 加载

## 使用Class类中的getSuperClass()方法能够得到一个类的父类

如果此 Class 表示 Object 类、一个接口、一个基本类型或 void，则返回 null。如果此对象表示一个数组类，则返回表示该 Object 类的 Class 对象。

## code

**private** **static** String clsTree(Class cls ) {

**if**(cls.toString().contains("ConcurrentRuntimeException"))

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

Class **parent** = cls.getSuperclass();

**if** (parent == Object.**class**)

**return** parent.getName() + ">" + cls.getName();

**else** **if** (parent == **null**) // interface

**return** "null>" + cls.getName();

**else** {

**return** *clsTree*(parent)+">"+cls.getName();

}

}

测试代码：

package com.attilax.util;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import org.apache.commons.io.FileUtils;

import org.apache.jasper.tagplugins.jstl.core.ForEach;

import org.apache.log4j.Logger;

import com.attilax.io.DirTraveService;

import com.attilax.io.PathService;

import com.attilax.ref.refService;

import com.cnhis.cloudhealth.module.log.Cls1;

import com.google.common.base.Function;

@SuppressWarnings("all")

public class PackageUtil {

public static Logger logger = Logger.getLogger(Cls1.class);

public static void main(String[] args) {

String packageName = "com.itkt.mtravel.hotel";

// com.google.common.annotations.Beta

String f = "C:\\Users\\attilax\\Desktop\\beked\\miniPrjjars\\guava-17.0.jar";

// guava-17.0.jar

new DirTraveService().traveV4Vs427("C:\\Users\\attilax\\Desktop\\beked\\miniPrjjars",

new Function<File, Object>() {

@Override

public Object apply(File f) {

logger.info("arg0:" + f.getAbsolutePath());

try {

List<String> classNames = refService.getClassName(f.getAbsolutePath());

for (String className : classNames) {

try {

Class cls = refService.loadclass(f.getAbsolutePath(), className);

String clstree = clsTree(cls);

logger.info("className:" + className);

logger.info("clstree:" + clstree);

} catch (Throwable e) {

logger.error(e);

}

}

} catch (Throwable e) {

e.printStackTrace();

logger.error(e);

}

return null;

}

}, new Function<Integer, Object>() {

@Override

public Object apply(Integer cnt\_index) {

// Map m=Maps.newConcurrentMap();m.put("index",

// cnt\_index)

logger.info("cnt\_index:" + cnt\_index);

return null;

}

});

// Class cls=

// List<String> getClassName = getClassName(f);

// for (String clsname : getClassName) {

// Class cls = refService.loadclass(f, clsname);

// System.out.println(cls);

// }

}

- clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>java.lang.IllegalArgumentException>org.apache.commons.beanutils.BeanAccessLanguageException

Line 45: 2018-05-16 16:42:56,833 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>org.apache.commons.beanutils.ConversionException

Line 209: 2018-05-16 16:42:56,924 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>java.lang.IllegalArgumentException>org.apache.commons.beanutils.BeanAccessLanguageException>org.apache.commons.beanutils.NestedNullException

Line 247: 2018-05-16 16:42:56,945 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>java.util.NoSuchElementException>org.apache.commons.collections.BufferUnderflowException

Line 515: 2018-05-16 16:42:57,292 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>org.apache.commons.collections.FunctorException

Line 517: 2018-05-16 16:42:57,293 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>org.apache.commons.collections.BufferOverflowException

Line 519: 2018-05-16 16:42:57,293 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>java.util.NoSuchElementException>org.apache.commons.collections.BufferUnderflowException

Line 1687: 2018-05-16 16:42:57,780 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>org.apache.commons.lang3.concurrent.ConcurrentRuntimeException

Line 1719: 2018-05-16 16:42:57,791 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>org.apache.commons.lang3.exception.CloneFailedException

Line 1723: 2018-05-16 16:42:57,793 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>org.apache.commons.lang3.exception.ContextedRuntimeException

Line 1761: 2018-05-16 16:42:57,811 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>java.lang.UnsupportedOperationException>org.apache.commons.lang3.NotImplementedException

Line 1801: 2018-05-16 16:42:57,825 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>org.apache.commons.lang3.SerializationException

Line 1997: 2018-05-16 16:42:57,985 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>com.alibaba.fastjson.JSONException>com.alibaba.fastjson.asm.ASMException

Line 2029: 2018-05-16 16:42:58,000 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>com.alibaba.fastjson.JSONException

Line 2081: 2018-05-16 16:42:58,021 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>com.alibaba.fastjson.JSONException>com.alibaba.fastjson.JSONPathException

Line 2713: 2018-05-16 16:42:58,412 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>com.google.common.base.VerifyException

Line 2779: 2018-05-16 16:42:58,441 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>com.google.common.cache.CacheLoader$InvalidCacheLoadException

Line 2783: 2018-05-16 16:42:58,442 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>java.lang.UnsupportedOperationException>com.google.common.cache.CacheLoader$UnsupportedLoadingOperationException

Line 3169: 2018-05-16 16:42:58,567 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>com.google.common.collect.ComputationException

Line 4275: 2018-05-16 16:42:58,884 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>java.lang.ClassCastException>com.google.common.collect.Ordering$IncomparableValueException

Line 5509: 2018-05-16 16:42:59,229 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>java.lang.IllegalStateException>com.google.common.util.concurrent.CycleDetectingLockFactory$ExampleStackTrace

Line 5523: 2018-05-16 16:42:59,234 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>java.lang.IllegalStateException>com.google.common.util.concurrent.CycleDetectingLockFactory$ExampleStackTrace>com.google.common.util.concurrent.CycleDetectingLockFactory$PotentialDeadlockException

Line 5771: 2018-05-16 16:42:59,323 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>com.google.common.util.concurrent.UncheckedExecutionException

Line 5773: 2018-05-16 16:42:59,323 - INFO main [PackageUtil$1] - clstree:java.lang.Object>java.lang.Throwable>java.lang.Exception>java.lang.RuntimeException>com.google.common.util.concurrent.UncheckedTimeoutException

Java反射机制demo（四）—获取一个类的父类和实现的接口 - AaronCui - 博客园.htmlJava反射机制demo（四）—获取一个类的父类和实现的接口 - AaronCui - 博客园.html