

sun.misc.Launcher

原创

[N3verL4nd](#) 于 2020-02-23 14:48:52 发布 900 收藏

分类专栏: [Java学习笔记](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/x_itya/article/details/104459985

版权



[Java学习笔记](#) 专栏收录该内容

173 篇文章 16 订阅

订阅专栏

针对 jdk1.8, jdk9 以后就没有该文件了。

<http://hg.openjdk.java.net/jdk8u/jdk8u/jdk/raw-file/8477fd88653c/src/share/classes/sun/misc/Launcher.java>

```
public class Launcher {  
    // 自定义通讯协议设计  
    private static URLStreamHandlerFactory factory = new Launcher.Factory();  
    // 不伦不类, 说你是单例你却有个公有的构造方法  
    private static Launcher launcher = new Launcher();  
    // Bootstrap classLoader 加载的路径  
    private static String bootClassPath = System.getProperty("sun.boot.class.path");  
    // Application classLoader  
    private ClassLoader loader;  
    // 没有用到  
    private static URLStreamHandler fileHandler;  
    // ...  
}
```

Launcher 构造方法

```
public Launcher() {  
    // 创建 ExtClassLoader  
    ClassLoader extcl;  
    try {  
        extcl = ExtClassLoader.getExtClassLoader();  
    } catch (IOException e) {  
        throw new InternalError("Could not create extension class loader", e);  
    }  
}
```

```
}

try {

    // 创建 AppClassLoader

    // AppClassLoader 的爹设置为 ExtClassLoader

    loader = AppClassLoader.getAppClassLoader(extcl);

} catch (IOException e) {

    throw new InternalError("Could not create application class loader", e);

}

// 将线程上下文加载器设置为 AppClassLoader

Thread.currentThread().setContextClassLoader(loader);

// Finally, install a security manager if requested

String s = System.getProperty("java.security.manager");

if (s != null) {

    SecurityManager sm = null;

    if ("".equals(s) || "default".equals(s)) {

        sm = new java.lang.SecurityManager();

    } else {

        try {

            sm = (SecurityManager) loader.loadClass(s).newInstance();

        } catch (IllegalAccessException | InstantiationException | ClassNotFoundException | ClassCastException e) {

        }

    }

    if (sm != null) {

        System.setSecurityManager(sm);

    } else {

        throw new InternalError("Could not create SecurityManager: " + s);

    }

}
```

```
}  
  
}
```

BootClassPathHolder 私有内部类，也是一种线程安全的单例实现模式。

```
private static class BootClassPathHolder {  
    static final URLClassPath bcp;  
  
    static {  
        URL[] urls;  
        if (bootClassPath != null) {  
            urls = AccessController.doPrivileged(  
                new PrivilegedAction<URL[]>() {  
                    @Override  
                    public URL[] run() {  
                        File[] classPath = getClassPath(bootClassPath);  
                        // 加快索引速度, JDK9 以后移除了。  
                        int len = classPath.length;  
                        Set<File> seenDirs = new HashSet<File>();  
                        for (int i = 0; i < len; i++) {  
                            File curEntry = classPath[i];  
                            // Negative test used to properly handle nonexistent jars on boot class path  
                            if (!curEntry.isDirectory()) {  
                                curEntry = curEntry.getParentFile();  
                            }  
                            if (curEntry != null && seenDirs.add(curEntry)) {  
                                MetaIndex.registerDirectory(curEntry);  
                            }  
                        }  
                        return pathToURLs(classPath);  
                    }  
                }  
            );  
        } else {  
            urls = new URL[0];  
        }  
        bcp = new URLClassPath(urls, factory, null);  
        bcp.initLookupCache(null);  
    }  
}
```

可以看到 getAppClassLoader 每次都是重新 new 一个 AppClassLoader。

```

static class AppClassLoader extends URLClassLoader {
    // 并行加载
    static {
        ClassLoader.registerAsParallelCapable();
    }

    public static ClassLoader getAppClassLoader(final ClassLoader extcl) throws IOException {
        final String s = System.getProperty("java.class.path");
        final File[] path = (s == null) ? new File[0] : getClassPath(s);

        return AccessController.doPrivileged(
            new PrivilegedAction<AppClassLoader>() {
                @Override
                public AppClassLoader run() {
                    URL[] urls = (s == null) ? new URL[0] : pathToURLs(path);
                    return new AppClassLoader(urls, extcl);
                }
            });
    }

    final URLClassPath ucp;

    AppClassLoader(URL[] urls, ClassLoader parent) {
        super(urls, parent, factory);
        ucp = SharedSecrets.getJavaNetAccess().getURLClassPath(this);
        ucp.initLookupCache(this);
    }

    @Override
    public Class<?> loadClass(String name, boolean resolve)
        throws ClassNotFoundException {
        int i = name.lastIndexOf('.');
        if (i != -1) {
            SecurityManager sm = System.getSecurityManager();
            if (sm != null) {
                sm.checkPackageAccess(name.substring(0, i));
            }
        }
        // 在缓存中查找

        if (ucp.knownToNotExist(name)) {
            Class<?> c = findLoadedClass(name);
            if (c != null) {
                if (resolve) {
                    resolveClass(c);
                }
                return c;
            }
            throw new ClassNotFoundException(name);
        }
        return (super.loadClass(name, resolve));
    }
    // ...
}

```

整个 JVM 内只有一个 ExtClassLoader

```

static class ExtClassLoader extends URLClassLoader {
    static {
        ClassLoader.registerAsParallelCapable();
    }
    // 单例
    private static volatile ExtClassLoader instance = null;

    public static ExtClassLoader getExtClassLoader() throws IOException {
        if (instance == null) {
            synchronized (ExtClassLoader.class) {
                if (instance == null) {
                    instance = createExtClassLoader();
                }
            }
        }
        return instance;
    }

    private static ExtClassLoader createExtClassLoader() throws IOException {
        try {
            return AccessController.doPrivileged(
                new PrivilegedExceptionAction<ExtClassLoader>() {
                    @Override
                    public ExtClassLoader run() throws IOException {
                        final File[] dirs = getExtDirs();
                        int len = dirs.length;
                        for (int i = 0; i < len; i++) {
                            MetaIndex.registerDirectory(dirs[i]);
                        }
                        return new ExtClassLoader(dirs);
                    }
                });
        } catch (java.security.PrivilegedActionException e) {
            throw (IOException) e.getException();
        }
    }

    public ExtClassLoader(File[] dirs) throws IOException {
        super(getExtURLs(dirs), null, factory);
        SharedSecrets.getJavaNetAccess().
            getURLClassPath(this).initLookupCache(this);
    }

    // ...
}

```

测试

```
@Test
public void test3() {
    System.out.println(new Launcher());
    System.out.println(new Launcher());
    System.out.println(new Launcher().getClassLoader());
    System.out.println(new Launcher().getClassLoader());
    System.out.println(new Launcher().getClassLoader().getParent());
    System.out.println(new Launcher().getClassLoader().getParent());
}
```

/ 输出*

```
sun.misc.Launcher@18ef96
sun.misc.Launcher@6956de9
sun.misc.Launcher$AppClassLoader@769c9116
sun.misc.Launcher$AppClassLoader@6aceb1a5
sun.misc.Launcher$ExtClassLoader@61e4705b
sun.misc.Launcher$ExtClassLoader@61e4705b
*/
```

```
System.getProperty("sun.boot.class.path")
```

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/resources.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/rt.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/sunrsasign.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/jsse.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/jce.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/charsets.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/jfr.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/classes
```

```
System.getProperty("java.ext.dirs")
```

```
/Users/N3verL4nd/Library/Java/Extensions
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext
/Library/Java/Extensions
/Network/Library/Java/Extensions
/System/Library/Java/Extensions
/usr/lib/java
```

```
System.getProperty("java.class.path")
```

idea 相关

/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar
/Applications/IntelliJ IDEA.app/Contents/plugins/junit/lib/junit-rt.jar
/Applications/IntelliJ IDEA.app/Contents/plugins/junit/lib/junit5-rt.jar
/Users/N3verL4nd/Library/Caches/IntelliJIdea2019.2/captureAgent/debugger-agent.jar

jre 相关

/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/charsets.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/deploy.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/javaws.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/jce.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/jfr.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/jfxswt.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/jsse.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/management-agent.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/plugin.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/resources.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/rt.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/lib/ant-javafx.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/lib/dt.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/lib/javafx-mx.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/lib/jconsole.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/lib/packager.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/lib/sa-jdi.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/lib/tools.jar

jre ext

/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext/cldrdata.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext/dnsns.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext/jaccess.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext/jfxrt.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext/localedata.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext/nashorn.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext/sunec.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext/sunjce_provider.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext/sunpkcs11.jar
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/jre/lib/ext/zipfs.jar

项目相关

/Users/N3verL4nd/Desktop/MTThriftStudy/target/test-classes
/Users/N3verL4nd/Desktop/MTThriftStudy/target/classes

pom 配置的包

/Users/N3verL4nd/.m2/repository/org/springframework/spring-webmvc/5.0.8.RELEASE/spring-webmvc-5.0.8.RELEASE.jar
/Users/N3verL4nd/.m2/repository/org/springframework/spring-aop/5.0.8.RELEASE/spring-aop-5.0.8.RELEASE.jar

...