在 IDEA 中使用 SecurityManager 调试 Tomcat

在 IDEA 中调试 Tomcat 源码的时候，要让 Tomcat 使用 `SecurityManager`，只要添加以下虚拟机参数即可：

```
-Djava.security.manager -Djava.security.policy=launch/conf/catalina.policy
```

本以为很简单,其实发现会遇到好些问题,，启动 Tomcat,很不幸出错了：

```
java.lang.ExceptionInInitializerError
Caused by: java.security.AccessControlException: access denied ("java.util.PropertyPermission" "idea.launcher.bin.path" "read")
    at java.security.AccessControlContext.checkPermission(AccessControlContext.java:372)
    at java.security.AccessController.checkPermission(AccessController.java:559)
    at java.lang.SecurityManager.checkPermission(SecurityManager.java:549)
    at java.lang.SecurityManager.checkPropertyAccess(SecurityManager.java:1302)
    at java.lang.System.getProperty(System.java:708)
    at com.intellij.rt.execution.application.AppMain.<clinit>(AppMain.java:39)
Exception in thread "main"
Process finished with exit code 1
```

问题很明显，只要修改下 `catalina.policy` 文件即可，添加如下的代码：

```
permission java.util.PropertyPermission "idea.launcher.bin.path", "read";
```

然后再运行,发现还是有问题,如下图：

```
java.lang.ExceptionInInitializerError
Caused by: java.security.AccessControlException: access denied ("java.lang.RuntimePermission" "loadLibrary.E:\Java\soft\IntelliJ IDEA 13.0.2\bin\breakgen64.dll")
    at java.security.AccessControlContext.checkPermission(AccessControlContext.java:372)
    at java.security.AccessController.checkPermission(AccessController.java:559)
    at java.lang.SecurityManager.checkPermission(SecurityManager.java:549)
    at java.lang.SecurityManager.checkLink(SecurityManager.java:835)
    at java.lang.Runtime.load0(Runtime.java:789)
    at java.lang.System.load(System.java:1062)
    at com.intellij.rt.execution.application.AppMain.<clinit>(AppMain.java:66)
Exception in thread "main"
Process finished with exit code 1
```

需要添加如下的权限：

```
permission java.lang.RuntimePermission "loadLibrary.*";
```

可以发现,解决了上面的问题后还存在很多其它的权限问题(一个坑接着一个坑).就不一一列举了,想一劳永逸,只需要添加 `permission java.security.AllPermission;`，但是这是不太合理的，而且对于后面的 `SecurityClassLoad` 的测试会造成影响,无法看到我们想要的结果.现在提供一种比较简单的方式，修改 `catalina.policy` 文件：

```
// bin 目录是项目编译后的 class 文件存放的位置，这里给它们添加所有的权限.
grant codeBase "file:${user.dir}/bin/-" {
        permission java.security.AllPermission;
};
```

可以当启动项目的时候发现，根本行不通，感到很奇怪，配置啥的明明应该没什么问题，为何行不通.为此，做了如下的实验：

权限策略在 Main 方法 和 Junit 测试中及在 Eclipse 和 IDEA 中运行的差异

以下程序先在 Eclipse 中测试：

```
/**
```

```
 * vm args: -Djava.security.manager -Djava.security.policy=src/security/myPolicy.policy
 */
public class PropertyPermissionTest {
    public static void main(String[] args) {
        System.out.println(System.getProperty("user.dir"));
        System.out.println(System.getProperty("xxx"));
    }


    @Test
    public void test() {
        System.out.println(System.getProperty("user.dir"));
        System.out.println(System.getProperty("xxx"));
    }
}
```

`myPolicy.policy` 内容如下：

```
grant codeBase "file:${user.dir}/bin/-" {
    permission java.util.PropertyPermission "user.dir", "read";
};
```

- 运行 `main` 函数,结果和预期的一致：

```
E:\Java\workspace\security\JavaSecurity
Exception in thread "main" java.security.AccessControlException: access denied ("java.util.PropertyPermission" "xxx" "read")
```

- 使用 `Junit` 运行.
  首先需要在策略文件中添加如下的内容：

```
grant {
    // junit 运行的时候需要
    permission java.net.SocketPermission "localhost:1024-", "listen,connect,resolve";
    permission java.lang.RuntimePermission "accessDeclaredMembers";
};
```

然后运行,发现程序不能如期的运行：

```
java.security.AccessControlException: access denied ("java.util.PropertyPermission" "user.dir" "read")
```

接下来,我们在 IDEA 中运行,发现即使是直接使用 `main` 函数运行的方式,也是不行的,更不用说 `junit` 的方式.
其实 Junit 的运行方式和 IDEA 中 `main` 直接运行的方式其实有些类似，使用 Junit 相当于使用了如下的命令启动：

```
java -Djava.security.manager -Djava.security.policy=src/security/myPolicy.policy -cp .;../junit-4.4.jar org.junit.runner.JUnitCore security.PropertyPermissionTest
```

我们在看下 IDEA 中运行的方式：

所以在调试安全这部分代码的时候，从 IDEA 切换到 Eclipse 是最好的方式了.
这次可以启动,不过 `webapps` 下的应用程序是不能被正常部署的.会出现以下的错误：

```
org.apache.catalina.LifecycleException: Failed to start component [/docs]
    at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:152)
    at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:682)
    at org.apache.catalina.core.ContainerBase.access$000(ContainerBase.java:129)
    at org.apache.catalina.core.ContainerBase$PrivilegedAddChild.run(ContainerBase.java:148)
    at org.apache.catalina.core.ContainerBase$PrivilegedAddChild.run(ContainerBase.java:139)
    at java.security.AccessController.doPrivileged(Native Method)
    at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:657)
    at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:637)
    at org.apache.catalina.startup.HostConfig.deployDirectory(HostConfig.java:1060)
    at org.apache.catalina.startup.HostConfig$DeployDirectory.run(HostConfig.java:1640) <4 internal calls>
```

接下来紧着着看下 `SecurityClassLoad` 类的作用.

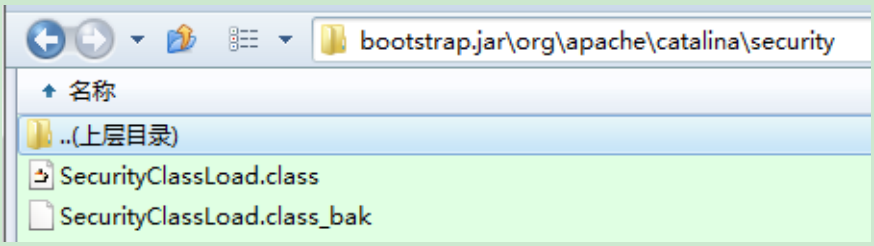
SecurityClassLoad 类的作用

`SecurityClassLoad` 类到底有什么作用,看注释写的确实比较简单,而且从字面理解上看,嗯,是和安全相关的,但到底具体的情况是什么样的?做个简单的实验,将下面的 loadXXX 都注释.

```
        System.out.println("********************SecurityClassLoad not preload java classes********************");

//      loadCorePackage(loader);
```

如果想在 IDEA 中做实验,想看到 `SecurityClassLoad`, 是基本上行不通的,目前没找到啥好的方法.如果你嫌麻烦的,可以将 `SecurityClassLoad` 类替换二进制安装包 `{catalina.home}/bin/bootstrap.jar` 中的 `SecurityClassLoad`.



如果你想进行 DEBUG 的话, 上面的方法就行不通了.不过你可以切换到 Eclipse 中,下面的权限在 Eclipse 中是生效的,前面已经解释过.

```
1.  grant codeBase "file:${user.dir}/bin/-" {
2.          permission java.security.AllPermission;
3.  };
```

启动 Tomcat,发现,咦,Tomcat 正常启动,貌似 `SecurityClassLoad` 根本一点用都木用.别急,让我们访问应用程序:

HTTP Status 500 - access denied ("java.lang.RuntimePermission" "accessClassInPackage.org.apache.catalina.connector")

type Exception report

message access denied ("java.lang.RuntimePermission" "accessClassInPackage.org.apache.catalina.connector")

description The server encountered an internal error that prevented it from fulfilling this request.

exception

```
java.security.AccessControlException: access denied ("java.lang.RuntimePermission" "accessClassInPackage.org.apache.catalina.connector")
        java.security.AccessControlContext.checkPermission(AccessControlContext.java:372)
        java.security.AccessController.checkPermission(AccessController.java:559)
        java.lang.SecurityManager.checkPermission(SecurityManager.java:549)
        java.lang.SecurityManager.checkPackageAccess(SecurityManager.java:1529)
        sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:305)
        java.lang.ClassLoader.loadClass(ClassLoader.java:412)
        java.lang.ClassLoader.loadClass(ClassLoader.java:358)
        org.apache.catalina.connector.RequestFacade.getParameter(RequestFacade.java:377)
        mypack.Demo08_CheckServlet.service(Demo08_CheckServlet.java:15)
```

现在终于是看到效果了,  .但是为啥 `SecurityClassLoad` 就预加载了那些类.目前还未深究.