

java 模拟脚本精灵

作者: [windvibra](#)

原文链接: <https://ld246.com/article/1574150347868>

来源网站: [链滴](#)




许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

脚本精灵实现

```
RunApp "E:\xx.exe"  
//查找退出所在位置  
FindPic 0,0,1920,1080,"Attachment:\exit.bmp",0.5,intX,intY  
//如果找到退出的位置  
If intX > 0 And intY > 0 Then  
//将鼠标移动到该位置  
MoveTo intX, intY  
//点击退出所在位置  
LeftClick 1  
//延时2秒  
Delay 2000  
//点击回车，确定退出  
KeyPress "Enter", 1  
End If  
//运行程序  
RunApp "E:\xx.exe"  
//查找开启服务所在位置  
FindPic 0,0,1920,1080,"Attachment:\start.bmp",0.5,intX,intY  
//如果找到开启服务的位置  
If intX > 0 And intY > 0 Then  
//将鼠标移动到该位置  
MoveTo intX, intY  
//点击开启服务所在位置  
LeftClick 1  
End If  
//延时2秒  
Delay 2000  
//回到桌面  
KeyDown 91, 1  
KeyPress 68, 1  
KeyUp 91, 1  
//延时10秒  
Delay 10000  
RunApp "E:\xx.exe"  
EndScript
```

而脚本精灵不能断网运行

所以使用java 这是需要的jar

 jna-platform-5....	2,681,931	711,546	JAR 文件	2019/11/18 1...	9E273B...
 jna-5.5.0.jar	1,506,993	1,486,510	JAR 文件	2019/11/18 1...	ADCAF7...
 hutool-all-5.0.5...	1,495,965	1,355,909	JAR 文件	2019/11/18 1...	06FA83C8

首先通过spy++获取到应用程序的窗口标题

然后放代码

```
private static WinDef.RECT initalizeApp(String appPath, String exeTitle) {  
    // 获取exeTitle程序句柄  
    User32 user32 = User32.INSTANCE;  
    HWND hwnd = user32.FindWindow(null, exeTitle);  
    if (hwnd == null) {  
        // 启动程序
```

```

        RuntimeUtil.exec(appPath);
        ThreadUtil.sleep(2000);
    }
    //重新查找程序句柄
    hwnd = user32.FindWindow(null, exeTitle);
    if (hwnd == null) {
        logger.error(DateUtil.now() + exeTitle + "---程序未运行");
        System.exit(0);
    }

    // 获取工具栏父级句柄
    HWND tdxDockControl = user32.FindWindowEx(hwnd, null, "TdxDockControl", null);
    if (tdxDockControl == null) {
        logger.info(DateUtil.now() + "tdxDockControl 对象获取失败");
        System.exit(0);
    }
    // 获取工具栏句柄
    HWND tdxBarControl = user32.FindWindowEx(tdxDockControl, null, "TdxBarControl", nul
);
    if (tdxBarControl == null) {
        logger.info(DateUtil.now() + "tdxBarControl 对象获取失败");
        System.exit(0);
    }
    WinDef.RECT Bar_RECT = getRECT(user32, hwnd, tdxBarControl);

    return Bar_RECT;
}

private static WinDef.RECT getRECT(User32 user32, HWND hwnd, HWND tdxBarControl) {
    // 打开窗口
    user32.ShowWindow(tdxBarControl, 9);
    // 切换窗口到最前
    user32.SetForegroundWindow(tdxBarControl);
    // 获取hwnd前台窗口
    WinDef.RECT hwnd_RECT = new WinDef.RECT();
    user32.GetWindowRect(hwnd, hwnd_RECT);
    // 获取bar前台窗口
    WinDef.RECT Bar_RECT = new WinDef.RECT();
    user32.GetWindowRect(tdxBarControl, Bar_RECT);
    return Bar_RECT;
}

public static void main(String[] args) {
    // 读取classpath下的config目录下的XXX.setting, 不使用变量
    Setting setting = new Setting("config.setting");
    String appPath = setting.getStr("appPath");
    String exeTitle = setting.getStr("exeTitle");
    String otherApp = setting.getStr("otherApp");
    String otherAppPath = setting.getStr("otherAppPath");
    // 初始化程序, 并获取工具栏尺寸
    WinDef.RECT Bar_RECT = initalizeApp(appPath, exeTitle);
    // 获取退出按钮尺寸
    int exit_width = (Bar_RECT.right - Bar_RECT.left) * 7 / 8 + Bar_RECT.left;
    int exit_height = (Bar_RECT.bottom - Bar_RECT.top) / 2 + Bar_RECT.top;
}

```

```
// 获取开始服务按钮尺寸
int start_width = (Bar_RECT.right - Bar_RECT.left) * 5 / 8 + Bar_RECT.left;
int start_height = (Bar_RECT.bottom - Bar_RECT.top) / 2 + Bar_RECT.top;
// 调用机器人第一次退出程序
RobotUtil.mouseMove(exit_width, exit_height);
RobotUtil.click();
//键盘回车
RobotUtil.keyClick(KeyEvent.VK_ENTER);
ThreadUtil.sleep(5000);
// 第二次初始化程序, 并获取工具栏尺寸
Bar_RECT = initializeApp(appPath, exeTitle);
// 调用机器人第二次开启服务
RobotUtil.mouseMove(start_width, start_height);
RobotUtil.click();
//显示桌面
RobotUtil.keyPressWithCtrl(KeyEvent.VK_D);
//判断是否启动第三方程序
if("true".equals(otherApp)) {
    ThreadUtil.sleep(5000);
    RuntimeUtil.exec(otherAppPath);
}
}
```

最后把它打包成一个exe文件放开机自启里

spy++下载地址: [spy++](#)