JOAuth2授权加密办法

护理平台接入指南

1. 文件

|  |  |  |
| --- | --- | --- |
| 文件名 | 路径 | 修改内容 |
| hutool-all-4.5.0.jar | /WEB-INF/lib | 复制到该目录 |
| UserAgentUtils-1.21.jar | /WEB-INF/lib | 复制到该目录 |
| db.properties | /WEB-INF/classes | 详细请见二.2 |
| SpringContext.xml | /WEB-INF/classes | 替换 |
| joauth2/\*.class | /WEB-INF/classes/com/joauth2 | 在com目录下新建目录joauth2，将文件复制到joauth2目录中 |

1. 接入说明
2. 在授权平台填写相关配置，创建新的应用
3. 替换所需文件，在db.properties中添加如下内容：

|  |
| --- |
| auth.app\_key=5c4dce10853146c099429d7cffefa2d0  auth.app\_secret=8fce4887f925998665f4e521029a0cf0  auth.redirect\_uri=http://127.0.0.1:8282/connect/auth/getToken  auth.url=http://www.njedusoft.com/joauth2/connect/oauth2  auth.app\_encrypt=xxx |

|  |
| --- |
| auth.app\_key：创建应用后提示的app key  auth.app\_secret: 创建应用后提示的app secret  auth.redirect\_uri: 固定格式，http://【护理平台IP】:【护理平台端口】/connect/auth/getToken  auth.url：固定地址  auth.app\_encrypt：创建应用后提示的app encrypt |

以上配置为授权平台使用，如需结合解密狗，应继续添加如下配置：

|  |
| --- |
| dog.codetype=65365BE28B3E29E6  dog.decode=CB1863036E45458872AFC5758935DF59  dog.encode=A9E03510D68B7F942A131AEE81D2866E  dog.beginpwd=B30FA70B10A8AAC6080BE9FE67749A0A  dog.endpwd=C328CE80CBB36F9AC1935826325D7D2F |

1. 仍需做的修改：

修改过滤器（拦截器），检查授权许可，当无有效授权时禁止登录；

修改登录方法，提示授权信息；

1. 上述工作完成后，启动程序，进行测试
2. 使用说明
3. 打开浏览器，输入地址：<http://www.njedusoft.com/joauth>2 ，填写登录名密码，进入授权平台
4. 菜单介绍：

系统管理

└ 用户管理：系统管理员的维护管理

└ 操作日志：平台内部操作的记录

应用管理

└ 应用列表：授权应用的维护管理

└ 创建应用：填写信息，创建一个新的应用，属性说明：

|  |  |
| --- | --- |
| 名称 | 描述 |
| 应用名称 | 标识一个应用 |
| IP | 应用所在服务器的IP |
| 间隔时间 | 只有在间隔时间点上发起的授权才会被允许 |
| 授权时间 | 允许授权的时间范围 |
| 登陆数量 | 允许登录的用户数量 |
| 应用描述 | 应用的介绍、备注信息 |

└ 数据统计：应用授权量排行，点击柱状图打开该应用的每日授权次数折现图

插件

└ 加密狗：加密狗的密文加解密

└ JSON相关：包括但不限于JSON相关的工具集合

└ 获取服务器IP

└ 获取内网IP

1. 接口列表
2. 请求说明：

1.1 相关的接口调用方法都封装在Client.java中，可以直接调用

1.2 接口请求方式分为GET和POST两种，首先调用AuthSecureUtil..encodeKeys()对请求参数进行加密，得到加密后的字符串，请求时只发送加密字符串，对应的key为“s”，示例：<http://www.njedusoft.com/joauth2/connect/oauth/getToken?s=1qaz>，其中1qaz即为参数加密后的字符串

1. 获取授权码

Client.getCode();

1. 获取Token

Client.getToken();

1. 用户上线

Client.saveLoginRecord();

1. 用户下线

Client.setLoginRecordOffline();

1. 初始化登录用户

Client.initLoginRecord();