

java应用接入sso系统-指南

针对版本V1.0.0

©成都基础平台架构组

2016/11/02

修订记录

版本号	修订内容	作者	审核	修订日期
V1.0.0	初始版本	尹同强	成都基础平台架构组	2016/11/02

目 录

1	概述.....	4
2	专业术语	4
3	基本流程	4
3.1.1	Java sdk 在 sso 登录流程如下	4
3.1.2	Java sdk 在 sso 退出流程如下.....	6
4	接入步骤	6
4.1.1	添加 MAVEN 依赖	7
4.1.2	添加 sso.properties 配置文件	7
4.1.3	添加 spring-sso.xml 配置文件	7
4.1.4	提供删除 token 的接口 exit	8
4.1.5	退出应用接口改造	9
附件一	开发者联系方式	11

1 概述

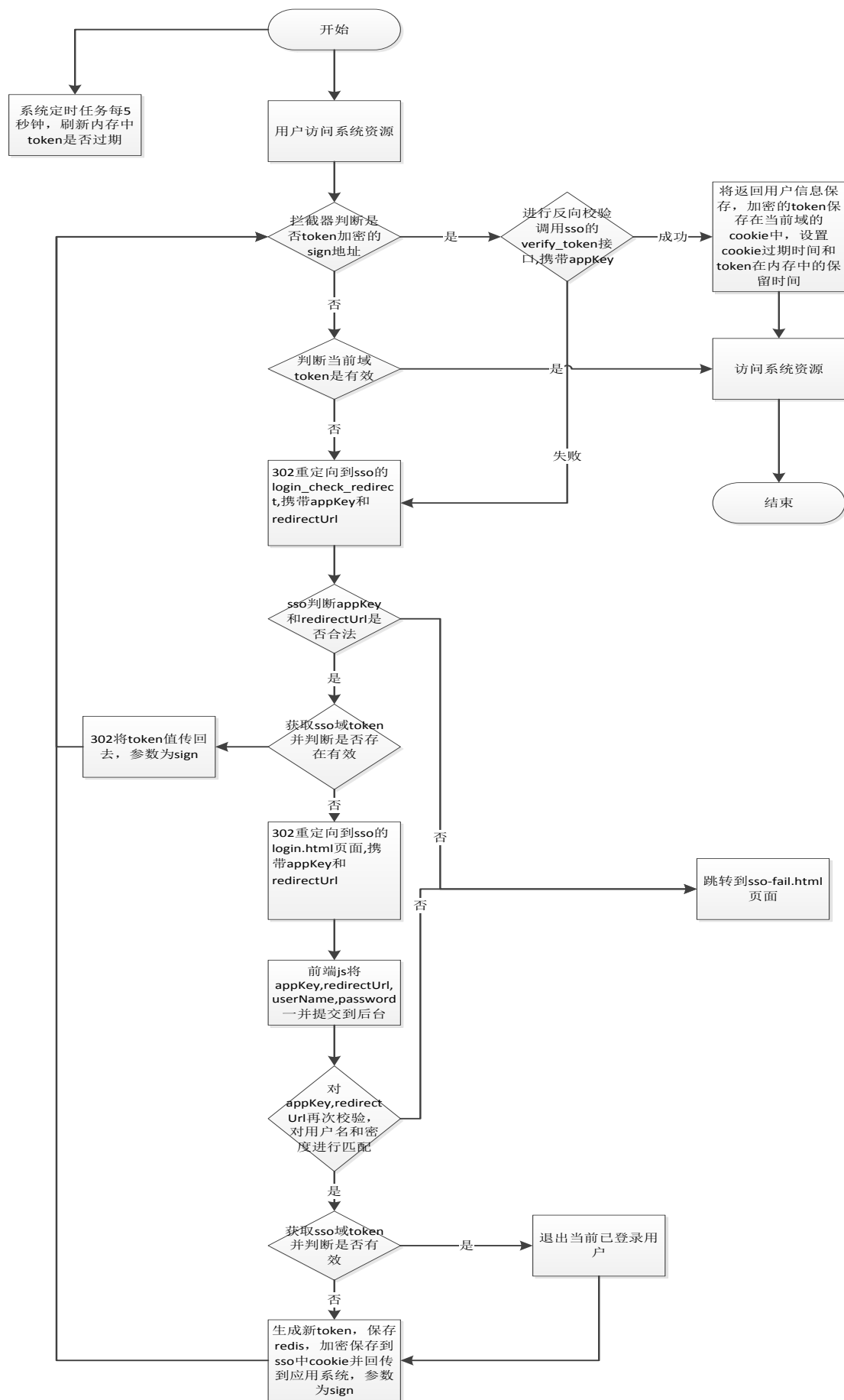
为国美电器运维系统、其他子系统，提供统一的单点登录方案 and 用户管理。

2 专业术语

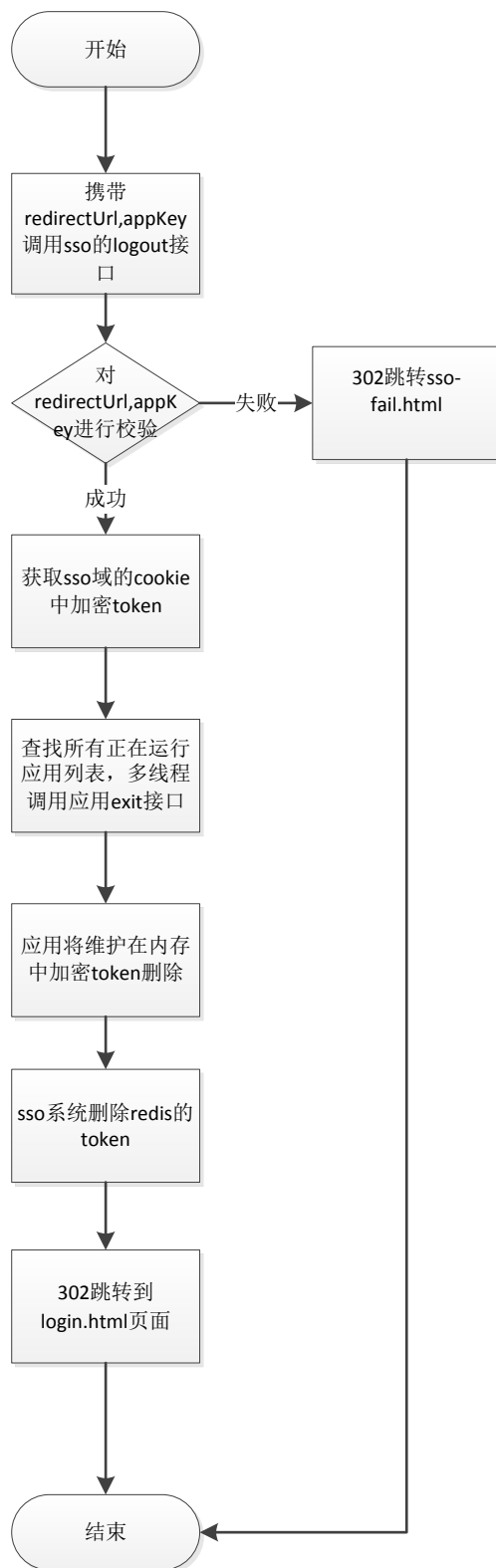
sso 单点登录及登录一次就可以访问所有相互信任的应用系统，退出一次所有系统都退出。

3 基本流程

3.1.1 Java sdk 在 sso 登录流程如下



3.1.2 Java sdk 在 sso 退出流程如下



4 接入步骤

当前 sdk 仅支持含有 spring 的 java 应用的接入。

4.1.1 添加 MAVEN 依赖

```
<dependency>

    <groupId>com.gome</groupId>

    <artifactId>gome-inc-sso-client</artifactId>

    <version>0.0.1-SNAPSHOT</version>

</dependency>
```

4.1.2 添加 sso.properties 配置文件

```
#sso 服务器配置

#token 校验地址

sso.tokenUrl=http://sso.com:8080/app/token

#sso 登录地址

sso.loginUrl=http://sso.com:8080/app/login

#sso 注销地址

sso.logoutUrl=http://sso.com:8080/app/logout


#-----

#应用配置

#应用登录成功首页

sso.app.homeUrl=http://a.com:8081/success

#应用的 appKey

sso.app.appKey=c806d10c617c4b48a5c9c34eab3856ea

#拦截器不需要拦截的 url, 多个以逗号隔开

sso.app.excludeUrl=/exit,/logout
```

4.1.3 添加 spring-sso.xml 配置文件

```
<?xml version="1.0" encoding="UTF-8"?>

<beans

    xmlns="http://www.springframework.org/schema/beans"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:p="http://www.springframework.org/schema/p"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

<context:property-placeholder location="classpath:sso.properties"
ignore-unresolvable="true"/>

<context:component-scan base-package="com.gome.*"/>

<bean

    id="gomeSSOServerConfig"

    class="com.gome.sso.client.config.GomeSSOServerConfig"

    p:tokenUrl="${sso.tokenUrl}"

    p:loginUrl="${sso.loginUrl}"

    p:logoutUrl="${sso.logoutUrl}"

    p:appHomeUrl="${sso.app.homeUrl}"

    p:appKey="${sso.app.appKey}"

    p:excludeUrl="${sso.app.excludeUrl}"

/>

<bean

    id="gomeSsoTokenService"

    class="com.gome.sso.client.service.impl.GomeSSOServiceImpl"

    init-method="init"

/>

</beans>
```

4.1.4 提供删除 token 的接口 exit


```
@RequestMapping("exit")

public RespData exit(HttpServletRequest httpServletRequest,
    HttpServletResponse httpServletResponse, @RequestBody String param) {

    String token = JSON.parseObject(param).getString("token");

    gomeSSOService.removeToken(token);

    return new RespData(RespCode.RESP_SUCCESS.getCode(),
        RespCode.RESP_SUCCESS.getMsg());
}
```

这是删除缓存中的 token 接口,每个应用都需要提供一个,名字可以改变。在 sso 后台配置应用时候需将此地址添加进去。

应用名称 *	<input type="text" value="测试应用A"/>
应用主页 *	<input type="text" value="http://a.com:8081/success"/>
应用域名 *	<input type="text" value="a.com"/>
服务器IP *	<input type="text" value="127.0.0.1"/>
token过期接口 *	<input type="text" value="http://a.com:8081/exit"/>
AppKey *	<input type="text" value="c806d10c617c4b48a5c9c34eab3856ea"/>
<input type="button" value="修改"/>	

4.1.5 退出应用接口改造

```
@RequestMapping("logout")

public void logout(HttpServletRequest httpServletRequest, HttpServletResponse
    httpServletResponse) {

    try {

        String localUrl = gomeSSOService.getAppHomeUrl();

        String appKey = gomeSSOService.getAppKey();
```

```
String ssoLogoutUrl = gomeSSOService.getSsoLogoutUrl();

String url = ssoLogoutUrl + "?redirectUrl=" +
URLLEncoder.encode(localUrl, "utf-8") + "&appKey="
    + appKey;

httpServletResponse.sendRedirect(url);
} catch (IOException e) {
    e.printStackTrace();
}
}
```

注意：此接口只是一个例子，每个应用在退出时候需删除自身应用的当前用户登录信息，并且重定向到 sso 的登录页面。

附件一 开发者联系方式

■ 成都研发中心·基础平台邮件组

cdxxjcpt@gome.com.cn

■ 成都基础平台组成员

姓名	联系方式	部门	更新日期
谭特贤	tantexian@gome.com.cn	成都研发中心	2016/11/02
郜焱磊	gaoyanlei@gome.com.cn	成都研发中心	2016/11/02
田玉粮	tianyuliang@gome.com.cn	成都研发中心	2016/11/02
尹同强	yintongjiang@gome.com.cn	成都研发中心	2016/11/02
罗继	luoji@gome.com.cn	成都研发中心	2016/11/02