8、自定义多个登录页面主题

# 接7中代码

# 1、主题

## 先介绍下什么叫主题，主题就意味着风格不一，目的就是为了在不同的接入端（service）展示不同的页面，就例如淘宝登录、天猫登录，其中登录点还是一个sso，但淘宝登录卖的广告是淘宝的，而天猫登录卖的广告是天猫的

# 2、静态资源存放位置

## 1、静态资源(js,css)存放目录为src/main/resources/static

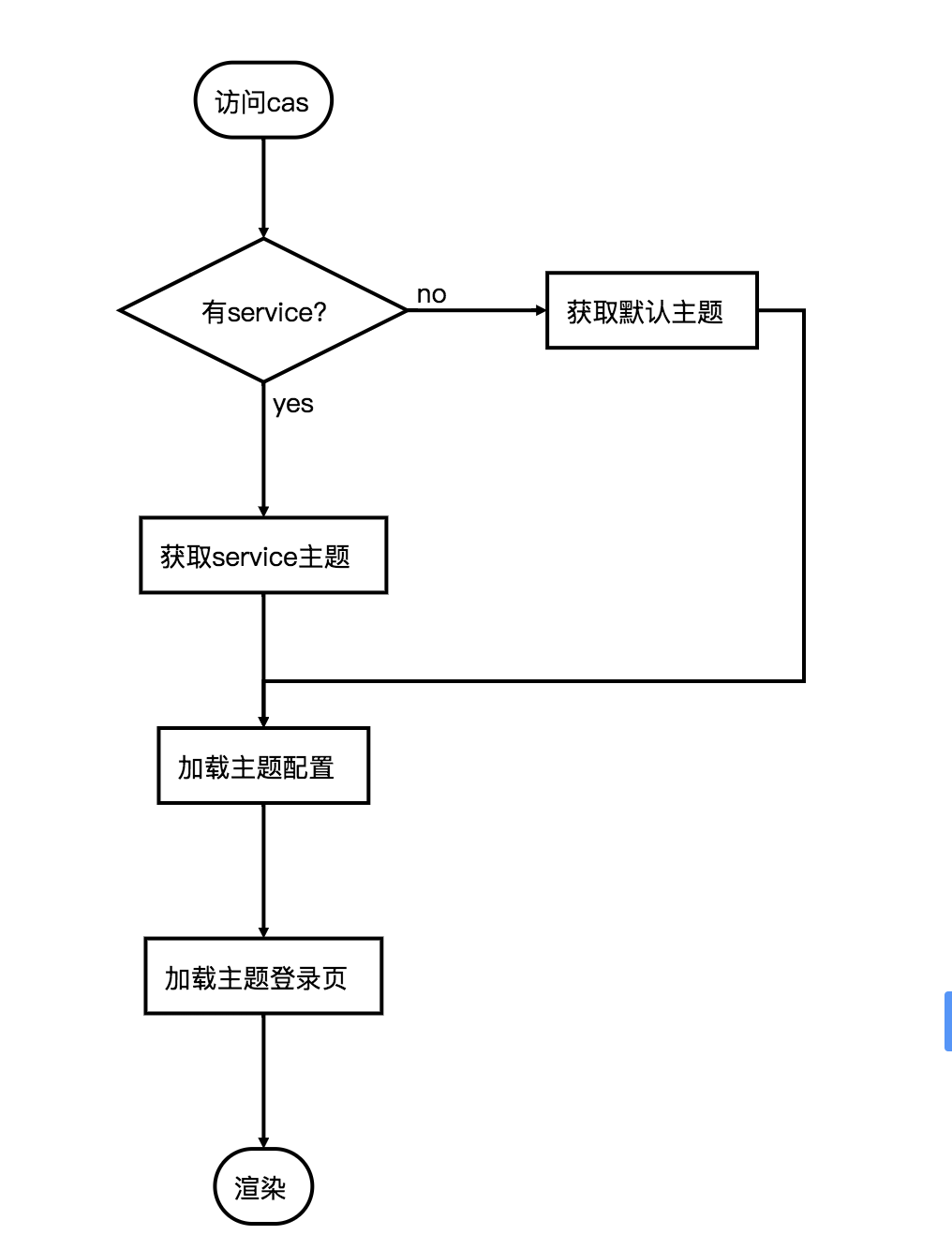
## 2、html资源存(thymeleaf)放目录为src/main/resources/templates

## 3、主题配置文件存放在src/main/resources并且命名为[theme\_name].properties

## 4、主题页面html存放目录为src/main/resources/templates/[theme\_name]

# 3、主题渲染是怎么样的？

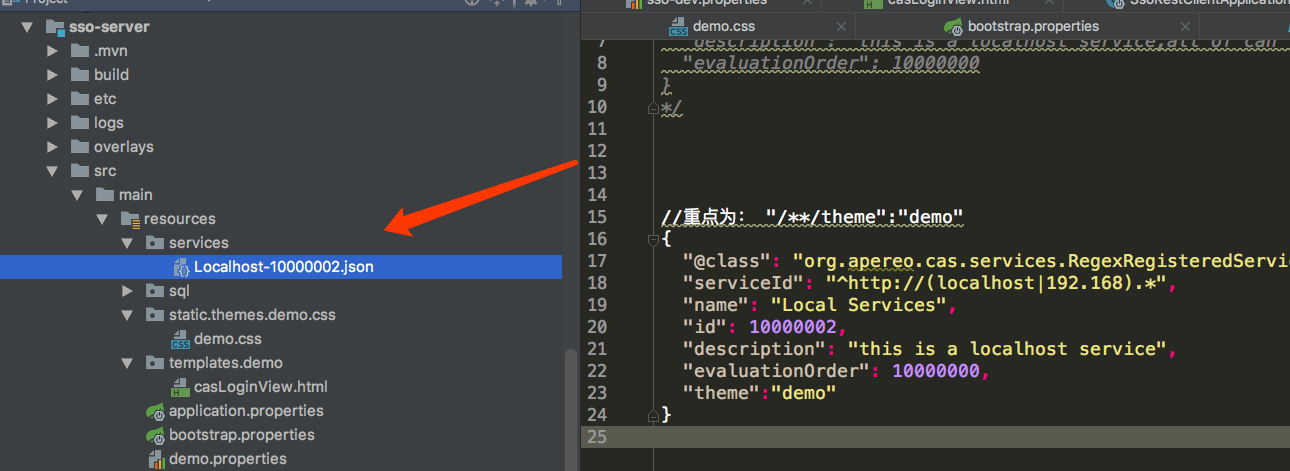
## 1、官方文档明确说明，登录页渲染文件为casLoginView.html,那意味我们在主题具体目录下新增改文件并且按照cas要求写那就可以了，最终目的还是获取到对应的配置文件，渲染对应主题的登录页



# 1、首先，新建主题设置service，确定主题的名字和id ，用来确定主题的登录页面

## theme为key指定配置文件id，若主题配置文件为test\_theme.application则"theme":"test\_theme"

//重点为： "/\*\*/theme":"demo"  
{  
 "@class": "org.apereo.cas.services.RegexRegisteredService",  
 "serviceId": "^http://(localhost|192.168).\*",  
 "name": "Local Services",  
 "id": 10000002,  
 "description": "this is a localhost service",  
 "evaluationOrder": 10000000,  
 "theme":"demo"  
}

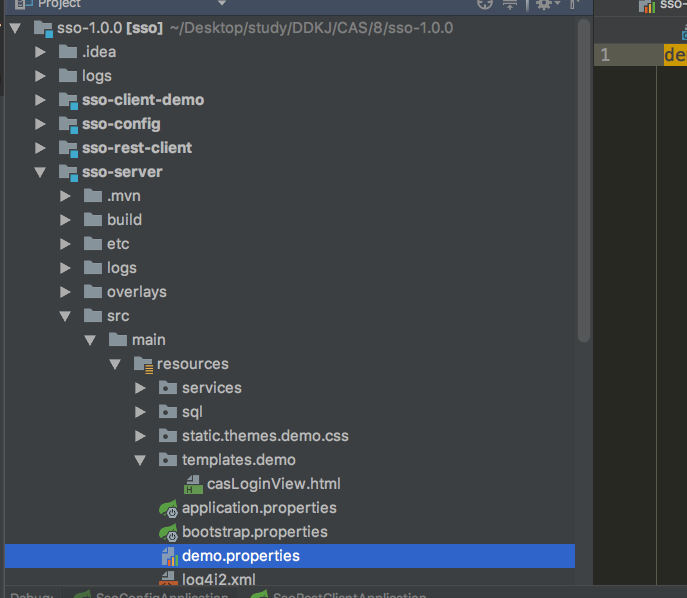


# 2、sso-config 中修改默认的主题为上面的demo （可以不设置默认的主题，这样login还是原来的，但是针对localhost的就变成了上面的主题）

# 7、修改默认的主题  
#通过service中确定主题的id（这里的id其实为thyme的值）  
cas.theme.defaultThemeName=demo  
# 由于第一次访问的时候会对页面进行缓存，为了防止不显示我们修改过的主题，这里设置为false  
spring.thymeleaf.cache=false

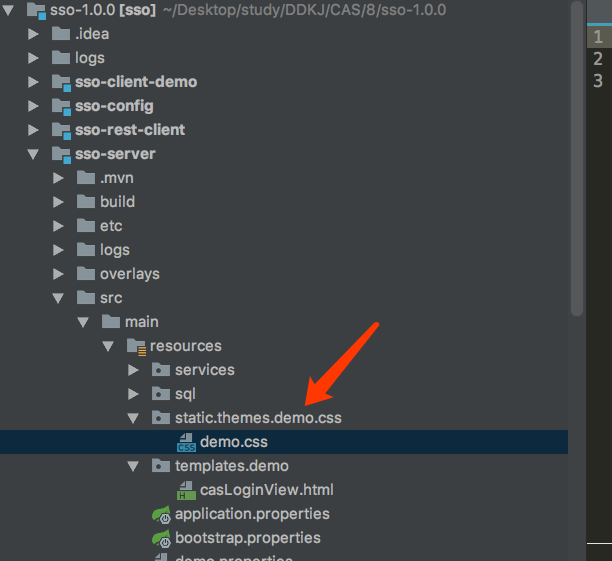
# 3、sso-server中新建主题配置文件，命名为demo.properties，确定主题所使用的样式

demo.css.file=/themes/demo/css/demo.css



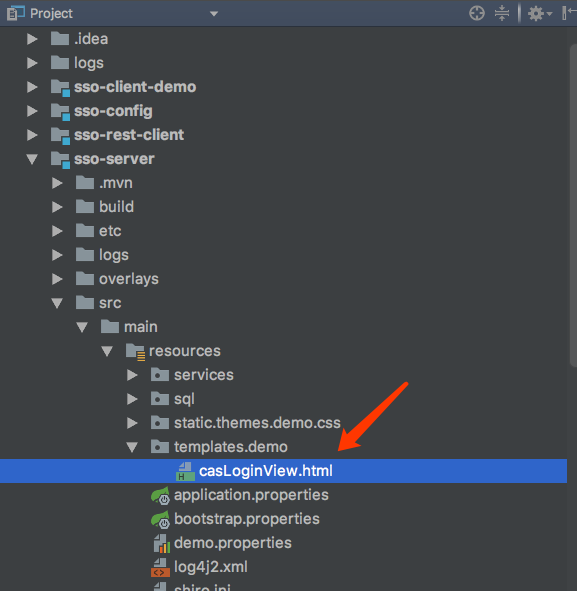
# 4、添加css样式 demo.css（目录位置随意在static中）

h1 {  
 color: blue;  
}



# 5、新建主题登录页面。规定目录为 明码规定文件名为casLoginView.html，路径为src/main/resources/templates/demo(主题名字也叫id)

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="UTF-8"/>  
 <meta name="viewport" content="width=device-width, initial-scale=1"/>  
 <meta http-equiv="X-UA-Compatible" content="IE=edge"/>  
 <title th:text="${#themes.code('demo.pageTitle')}"></title>  
 <link rel="stylesheet" th:href="@{${#themes.code('demo.css.file')}}"/>  
</head>  
  
<body>  
<h1 th:text="${#themes.code('demo.pageTitle')}"></h1>  
<div>  
 <form method="post" th:object="${credential}">  
 <div th:if="${#fields.hasErrors('\*')}">  
 <span th:each="err : ${#fields.errors('\*')}" th:utext="${err}"/>  
 </div>  
 <h2 th:utext="#{screen.welcome.instructions}"></h2>  
  
 <section class="row">  
 <label for="username" th:utext="#{screen.welcome.label.netid}"/>  
 <div th:unless="${openIdLocalId}">  
 <input class="required"  
 id="username"  
 size="25"  
 tabindex="1"  
 type="text"  
 th:disabled="${guaEnabled}"  
 th:field="\*{username}"  
 th:accesskey="#{screen.welcome.label.netid.accesskey}"  
 autocomplete="off"/>  
 </div>  
 </section>  
  
 <section class="row">  
 <label for="password" th:utext="#{screen.welcome.label.password}"/>  
 <div>  
 <input class="required"  
 type="password"  
 id="password"  
 size="25"  
 tabindex="2"  
 th:accesskey="#{screen.welcome.label.password.accesskey}"  
 th:field="\*{password}"  
 autocomplete="off"/>  
 </div>  
 </section>  
  
 <section>  
 <input type="hidden" name="execution" th:value="${flowExecutionKey}"/>  
 <input type="hidden" name="\_eventId" value="submit"/>  
 <input type="hidden" name="geolocation"/>  
 <input class="btn btn-submit btn-block"  
 name="submit"  
 accesskey="l"  
 th:value="#{screen.welcome.button.login}"  
 tabindex="6"  
 type="submit"/>  
 </section>  
 </form>  
</div>  
</body>  
</html>



# 7、这里用到的thyme所以添加下thyme的依赖吧

<!-- https://mvnrepository.com/artifact/org.thymeleaf/thymeleaf -->  
<dependency>  
 <groupId>org.thymeleaf</groupId>  
 <artifactId>thymeleaf</artifactId>  
 <version>3.0.9.RELEASE</version>  
</dependency>

# 8、启动sso-config ,sso-rest-client sso-client-demo,sso-server 浏览器中访问 <http://localhost:8080/sample/wangwu.jsp>



## 2、输入用户名和密码 rest-admin/123 登录成功

