Springboot + OAuth2 + JWT 单点登录

OAuth2基本概念和运作流程

源码-Spring Security OAuth2

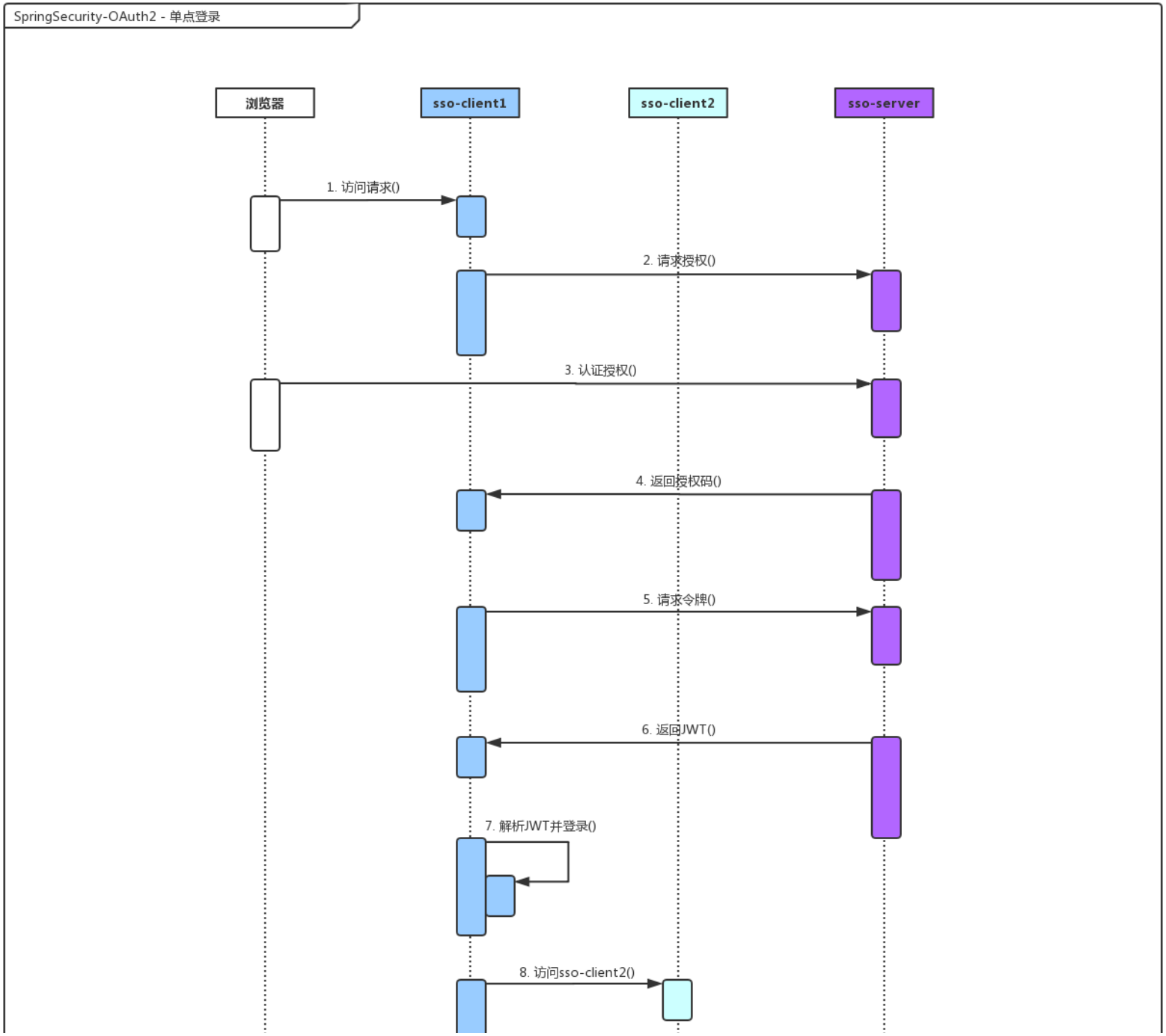
spring-security-oauth2 使用 AuthorizationEndpoint 自定义授权页面

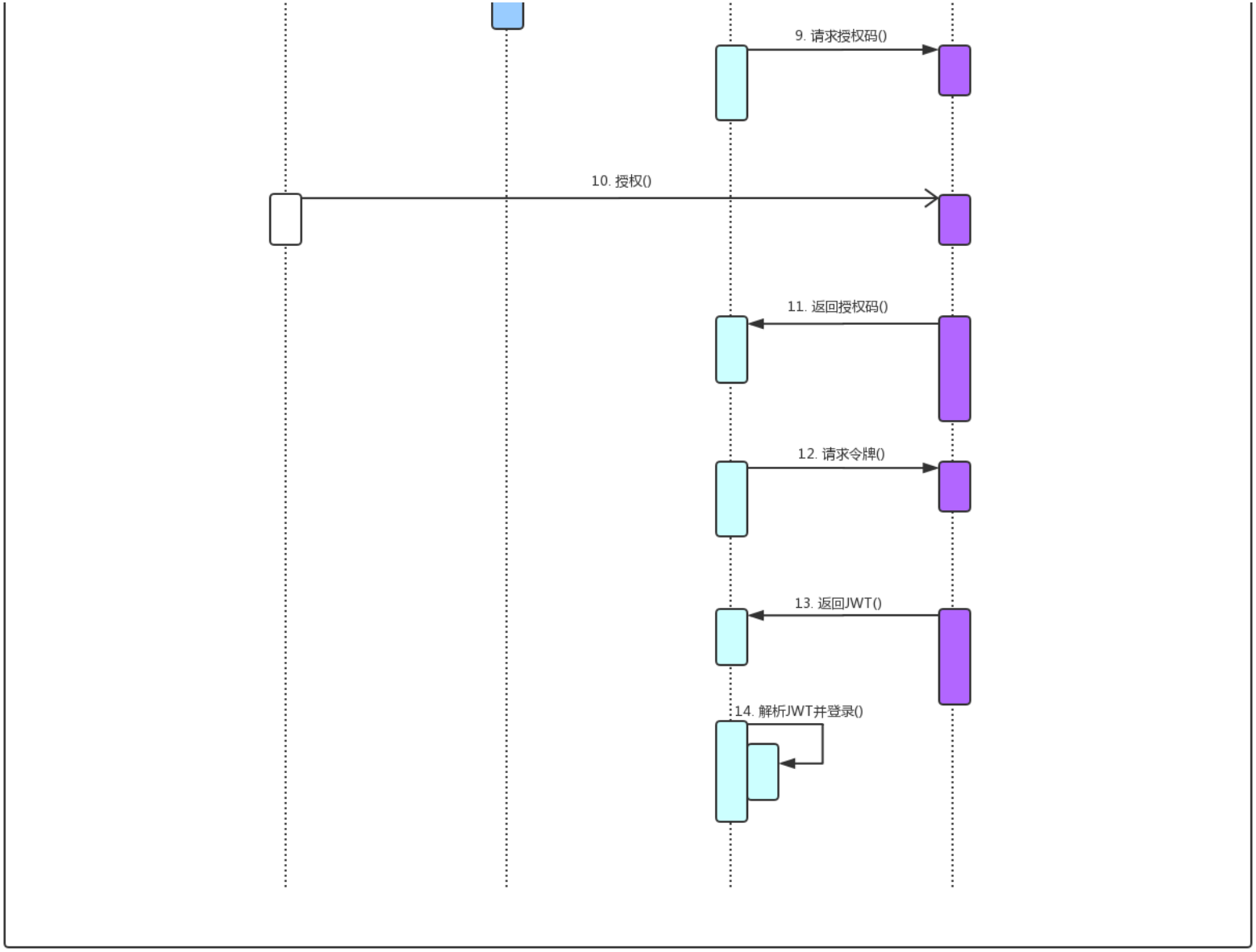
https://blog.csdn.net/yucaifu1989/article/details/85053669 <
https://github.com/deadzq/oauth-boot <待 (可用,按照demo中readme完全可操作)
oauth-boot的使用无障碍, 但没有整体一个逻辑,可以自己做.
oauth-boot-up 的暂时还没看.

https://projects.spring.io/spring-security-oauth/docs/Home.html < OAuth官网doc

https://github.com/longfeizheng < 大神郑龙飞的一些demo
https://github.com/longfeizheng/sso-merryyou < 使用spring security 和 OAuth2 实现单点登录
springboot2.0-oauth2 ,运行SpringBoot2Oauth2Application 报错

sso-merryyou


SpringSecurity-OAuth2 - 单点登录
浏览器
sso-client1
sso-client2
sso-server
1. 访问请求()
2. 请求授权()
3. 认证授权()
4. 返回授权码()
5. 请求令牌()
6. 返回JWT()
7. 解析JWT并登录()
8. 访问sso-client2()



*虽然不清楚为什么需要在client1跳转到client2,但是整个逻辑还算比较顺的.

项目分包:

sso-client1 : context-path:/client1
sso-client2 : context-path:/client2
sso-resouce : context-path:/resource
sso-server : context-path:/uaa

---

sso-server 内部有几个关键的类:
SsoAuthorizationServerConfig 它实现的是AuthorizationServerConfigurerAdapter (<属于 org.springframework.security.oauth2.config….configuration包中).
重写了三个重载方法:

```
public void configure(ClientDetailsServiceConfigurer clients) throws Exception {}

public void configure(AuthorizationServerEndpointsConfigurer endpoints) throws Exceptio
n {}

public void configure(AuthorizationServerSecurityConfigurer security) throws Exception
{}
```

Spring-Security-Core : 封装了验收, 授权流程核心基础类
Spring-Security-Web : 是基于HTTP实现对安全验证的封装.
Spring-Security-Config : 实现了Spring Security命名空间的配置
Spring-Security-Oauth2 : 通过Spring Security框架,实现Oauth2标准验证流程.

Spring-Security-Core 分析:

org.springframework.security.access

权限访问处理层,包含了访问权限annotation的配置注解, 访问验证的是event, 访问权限配置的表达式解析 expression, 对访问权限的方法拦截器 intercept , 权限方法注解的实现 method , 验证的post过程 prepost , 还有核心的 授权方法与管理 vote .

核心类访问决策器AccessDecisionVoter

AbstractAclVoter : 提供编写域对象ACL选项的帮助方法,没有绑定到任何特定的ACL系统.

AclEntryVoter : 给定一个作为方法参数传递的域对象实例, 确保类要绑定合适的权限AclService.

AuthenticatedVoter : 对IS_AUTHENTICATED_FULLY或IS_AUTHENTICATED_REMEMBERED或IS_AUTHENTICATED_ANONYMOUSLY检查与验证.

ClientScopeVoter : [Oauth2]检查是否在客户端的权限范围内

Jsr250Voter : 通过JSR-250配置的注解进行授权

PreInvocationAuthorizationAdviceVoter : 使用@PreFilter和@PreAuthorize注释生成的PreInvocationAuthorizationAdvice来授权

RoleVoter : 匹配默认的前缀字符串是ROLE_的ConfigAttribute, 如果匹配到则授权, 针对角色的授权.

RoleHierarchyVoter : 扩展RoleVoter使用RoleHierarchy定义的来确定在授权给当前用户的角色

ScopeVoter : 匹配默认的前缀字符串是SCOPE_的ConfigAttribute,如果匹配到则授权, 如: 授权范围 SCOPE_READ, SCOPE_WRITE

```
int ACCESS_GRANTED = 1; //决策结果-允许
int ACCESS_ABSTAIN = 0; //决策结果-放弃
int ACCESS_DENIED = -1; //决策结果-拒绝
//决策方法
int vote(Authentication authentication,S object,Collection<ConfigAttribute> attributes)
;
```

核心类访问控制决策管理AccessDecisionManager

AffirmativeBased实现类轮询所有配置AccessDecisionVoter的,并且如果有的话AccessDecisionManager肯定的授权访问权限.

ConSensusBased实现类轮询所有配置AccessDecisionVoter的,并且如果AccessDecisionVoter肯定大于否定的数量的话就授权访问权限,相等的话就看AbstractAccessDecisionManager.isAllowIfAllAbstainDecisions()方法(默认为false).

UnanimousBased实现类轮询所有配置AccessDecisionVoter的每一个配置,并且如果AccessDecisionVoter全部是肯定才能授予访问权限.

```
//决策方法,如果决策不通过就抛出异常
void decide(Authentication authentication,Object object,Collection<ConfigAttribute> con
figAttributes) throws AccessDeniedException,InsufficientAuthenticationException;
```

org.springframework.security.authentication

权限认证处理层,实现了对权限的管理和认证,dao封装了用户信息的访问,encoding封装了对用户密码加密的处理,event实现了对授权认证的成功/失败等事件,jaas是对Java Jaas授权API的封装,rcp提供了远程授权验证与管理.