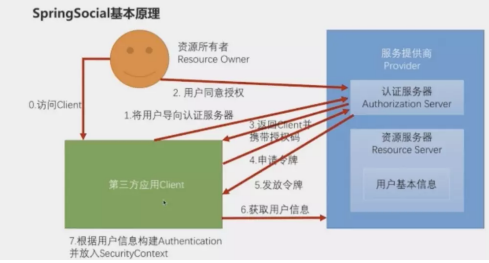
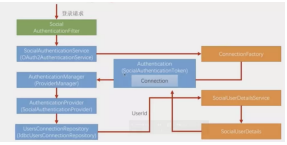
https://docs.spring.io/spring-social/docs/2.0.0.M4/reference/htmlsingle/
spring social 文档地址如上



图一


图二


图三


图四

图一为OAuth2认证流程图
图二为spring security中 spring用于处理social的功能的过滤器的位置（SocialAuthenticationFilter）

图三为需要自己自定义的类的信息，来进行相应的社交登录（详细讲）
1. AbstractOAuth2ApiBinding： 由用户继承，实现相应的获取社交（qq用户）用户的具体信息。实现类 QQImpl.
2. OAuth2Template 实现OAuth2Operations: 处理图一中认证流程的前5步，实现获取accessTokenUrl等信息. 具体的流程如果不同需要自己写一个子类复写相应的方法来满足服务提供商的要求.
3.AbstractOAuth2ServiceProvider : 封装1和2的实例.提供给ConnectionFactory来使用
4.OAuth2ConnectionFactory （extends ConnectionFactory）： 目的是创建出Connection的实例用于和数据库中的用户进行比较.
5.Connection : 根据整个流程返回的第三方用户的信息，封装成的一个类.其实例就对应着这个第三方账号对应的用户信息.

6.ApiAdapter:用于把AbstractOAuth2ApiBinding中获取的用户信息的封装类实例（QQUserInfo实例），适配成标准的connection信息,connection会被封装成一个SocialAuthenticaionToken。

图四：完成整个获取获取第三方信息后，再用JdbcUsersConnectionRepository来从数据库中获取相应的用户信息（userDetails），与SocialAuthenticaionToken进行校验，完成，则把userDetails封装到token中. 没有则作相应的处理.

最后需要一个配置类来集成SocialAutoConfigurerAdapter，进行创建相应的ConnectionFactory.并且加上@Configuration ， 最好加上@ConditionalOnProperty(prefix ="cn.kkcoder.security.social.qq" ,name = "app-id" )来进行条件设置,即如果没有配置cn.kkcoder.security.social.qq.app-id 则该配置不生效.