**简述：**

* 允许内存、数据库、JWT等方式存储令牌
* 允许 JWT 方式验证令牌
* 允许从内存、数据库中读取客户端详情
* 封装配置类，简化配置，通过注解方式定制使用何种令牌存储方式、客户端详情获取方式，可使用 JWT 方式存储令牌，从数据库中获取客户端详情
* 提供完整单元测试
* 较为详细的代码注释
* 允许从授权服务器（适用于 JDBC、内存存储令牌）验证令牌 **该代码尚未完善，仅供参考**
* 数据库 Schema : [https://github.com/spring-projects/spring-security-oauth/blob/master/spring-security-oauth2/src/test/resources/schema.sql](https://link.jianshu.com?t=https%3A%2F%2Fgithub.com%2Fspring-projects%2Fspring-security-oauth%2Fblob%2Fmaster%2Fspring-security-oauth2%2Fsrc%2Ftest%2Fresources%2Fschema.sql)
* Demo Git 地址：[https://gitee.com/LinYuanTongXue/OAuth2-Demo](https://link.jianshu.com?t=https%3A%2F%2Fgitee.com%2FLinYuanTongXue%2FOAuth2-Demo" \t "_blank)

**Demo 流程：**

* 使用 OAuth2 密码授权方式提供令牌
* 资源服务器1（也为客户端）提供登录接口，资源所有者（用户）通过将个人账号密码提供给 资源服务器1，资源服务器1 通过该信息向授权服务器获取令牌
* 资源服务器1（也为客户端）通过令牌（其中包含了客户端、用户等信息）访问自身受保护的资源（需要权限才能查看的资源）
* 资源服务器2（也可资源服务器）不包含登录接口，但其提供了某些受保护的资源（需要资源服务器1带着访问令牌才能访问）
* 资源服务器1（也为客户端）通过令牌向 资源服务器2（资源服务器） 请求其受保护的资源

**使用**

* 授权服务器通过继承 **AuthServerConfig** 抽象类来配置授权服务器
* 资源服务器、客户端通过继承 **ResServerConfig** 抽象类来配置资源服务器
* Web 权限配置可继承 **AbstractSecurityConfig** 抽象类来简化配置
* 授权服务器通过在启动类添加以下注解来设置令牌存储、客户端信息获取方式
  + **@EnableAuthJWTTokenStore**：使用 JWT 存储令牌
  + **@EnableDBClientDetailsService**：通过数据库获取客户端详情
  + **@EnableDBTokenStore**：通过数据库存储令牌
* 资源服务器通过在启动类添加以下注解来设置令牌解析方式
  + **@EnableResJWTTokenStore**：使用 JWT 解析令牌
  + **@EnableRemoteTokenService** **该代码尚未完善，仅供参考**

**项目结构**

* 下图是 Demo 项目结构，使用了 Maven 之间的继承关系，并添加了热部署，不了解的可以查看下 Git 上的 Demo 源码
  + oauth2-config：该包中定义了一些通用的类，例如授权服务器、资源服务器配置类，服务继承该类来简化配置
  + authentication-server：授权服务器
  + resource1-server：资源服务器1（也为客户端）
  + resource2-server：资源服务器2（也为资源服务器）

作者：林塬  
链接：https://www.jianshu.com/p/d80061e6d900  
来源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。