

# 基于 Spring 和 OAuth2.0 的第三方授权框架

刘 姚

(南京邮电大学 通信与信息工程学院,江苏 南京 210000)

**摘 要:** OAuth2.0 是一个开放标准的第三方授权协议,允许用户授权第三方平台获取在某一网站上存储的用户个人资源,而无需将用户名和密码提供给第三方平台。这个协议的主要作用就是定义了一个标准协议,允许一个 Web 或 APP 在用户授权下访问用户的隐私数据而无须了解用户的账号信息,这些数据可以存储在诸如微信、支付宝中。Spring Security For OAuth 2.0 为 OAuth2.0 的软件实现提供了一个开源 Java 库,广泛用于基于 Spring 框架的 Web 站点上,与 Spring Security 框架无缝衔接,易于 Web 后端服务器的升级,简化了基于 Web 的 OAuth2.0 协议的开发。文中分析了 OAuth2.0 协议细则以及关键流程,重点阐述了 Spring Security For OAuth 2.0 的优点、应用以及服务器配置,并以此为基础快速构建 OAuth2.0 服务器软件平台。该协议库具有易于使用、易于维护与使用安全等特点,目前已经广泛应用在互联网以及金融等领域。

**关键词:** Web 安全;Spring Security;OAuth2.0;开放平台认证

**中图分类号:** TP39

**文献标识码:** A

**文章编号:** 1673-629X(2017)03-0167-04

doi:10.3969/j.issn.1673-629X.2017.03.035

## Investigation on Third Party Authorization System Based on Spring Security and OAuth2.0

LIU Yao

(College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210000, China)

**Abstract:** OAuth2.0 is a third party authorization protocol of open standard, and allows the user to authorize a third party to obtain a user's personal resources stored on a Web site without having to provide the user name and password to a third party platform. The main role of this agreement is to define a standard protocol that allows a Web or APP access the private data of the user in the case of authorized, and the data can be stored in areas such as Alipay and WeChat. Spring Security For OAuth 2.0 provides an open source Java library for OAuth2.0 implementations widely used in Web sites based on the Spring framework and Spring Security framework for seamless, easy to upgrade for Web back-end, simplification of the development of Web-based OAuth 2.0. The OAuth2.0 protocol rules as well as the key process are analyzed, and the Spring Security For OAuth 2.0 advantages, applications and server configuration are described, and as a foundation to quickly build the OAuth2.0 server software platform. The protocol library has features of easy to use, easy to maintain and use security, now widely used in the Internet, as well as financial and other fields.

**Key words:** Web security; Spring Security; OAuth2.0; open platform certification

## 0 引 言

早期的互联网平台与应用系统是相互独立的,各个平台与系统之间不具有数据共享功能<sup>[1]</sup>。一个平台或者一个系统只能使用自己的系统资源,同时也不可以访问其他平台的资源。随着互联网的迅速发展,各个平台之间的联系日益密切,互联网平台之间的数据共享亟待解决。

OAuth 协议为资源服务器,第三方平台与用户之

间提供了一个安全、开放并且简易的协议标准<sup>[2]</sup>。任何第三方平台均可以使用 OAuth 认证服务在用户的许可下获取用户所允许的用户所属互联网资源。

Spring 是一个轻量级的 JAVAEE 框架,提供了一个标准的 IOC 容器<sup>[3]</sup>,全面支持 AOP 开发等特点,在企业应用开发与 Web 网站开发中占有重要地位。Spring Security 是 Spring 的一个子项目,在 Web 安全方面应用广泛。Spring Security 对 OAuth 协议提供了

收稿日期:2016-04-25

修回日期:2016-08-11

网络出版时间:2017-02-17

基金项目:国家自然科学基金资助项目(61271234)

作者简介:刘 姚(1992-),男,硕士研究生,研究方向为卫星通信技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20170217.1630.054.html>

完整的支持与整合<sup>[4]</sup>。考虑到 OAuth2.0 协议开发的复杂性,文中以 Spring Security 开源框架为基础开发第三方授权软件平台,具有开发周期短、代码量少、实用性强等特点。

1 OAuth2.0 协议简介

OAuth2.0 是 OAuth 的下一代开放平台授权协议, OAuth 旨在增加为开发 Web 应用程序、桌面应用程序、移动电话和客厅设备的简易性。并且作为标准协议入驻 IETF<sup>[5]</sup>。

OAuth 为客户端提供了一种代表资源所有者访问受保护资源的方法。在客户端访问受保护资源之前,它必须先从资源所有者获取授权(访问许可),然后用访问许可交换访问令牌(Access Token,包含许可的作用域、持续时间和其他属性等信息)。客户端通过向资源服务器出示访问令牌来访问受保护资源<sup>[6]</sup>。

在 OAuth2.0 协议流的定义下,第三方获取资源必须按照获取认证、获取访问资源令牌、通过令牌获取指定资源的顺序。其 workflow 顺序如图 1 所示<sup>[7]</sup>。

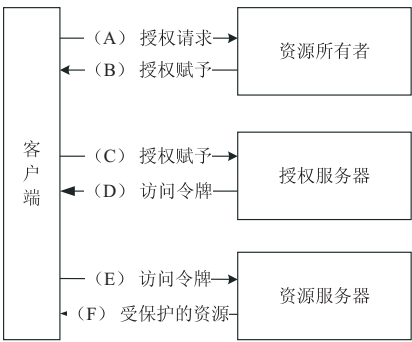


图 1 OAuth 抽象协议流

(A)客户端向资源服务器(可以是 OAuth 服务器)请求用户(资源所有者)授权。

(B)客户端获得授权,表现为资源所有者的授权凭据,可以是 OAuth2.0 所描述的返回类型,也可以是扩展类型。

(C)客户端通过使用授权服务器进行身份验证和提交授权授予请求访问令牌。

(D)授权服务器对客户端进行身份验证和验证授权授予,如果有效,会发出一个访问令牌。

(E)客户端从资源服务器请求受保护的资源,并使用访问令牌进行身份验证。

(F)资源服务器验证访问令牌,如果有效,则对请求进行服务。

OAuth2.0 中的资源所有者负责给客户端授权。客户端应用代表资源所有者获取受保护的资源。资源服务器负责提供这些受保护的资源。授权服务器负责为客户端应用提供 Access Token<sup>[8]</sup>。

OAuth2.0 包含四种授权类型:授权码(Web 应用使用)、隐式授权(基于浏览器或移动应用)、用户密码方式与客户端应用凭证(基于应用)授权方式<sup>[9]</sup>。

OAuth2.0 采用短期有效形式的令牌,可以通过刷新令牌来保持令牌的长期使用,其数据格式如下所示<sup>[10]</sup>:

```
{
  "access_token": "1CutcHUDNjksDkxIUvA",
  "token_type": "bearer",
  "expires_in": 3600,
  "refresh_token": "vAgd2IOpQ9XG0Dx1E2KWAE",
}
```

2 Spring 对 OAuth2.0 的支持

OAuth2.0 有多种实现方案,如 php、python、C++ 与 Java 等等。Spring 基于 Java 平台实现,对 OAuth2.0 进行二次适配。

OAuth2.0 包含众多 Java 语言实现框架,如 Jersey、Apache Oltu、Spring Security OAuth2、Google OAuth2 API 等等。Jersey 提供了对 Java EE 标准安全框架的集成,但只提供了客户端的解决方案。Apache Oltu 也是一个较为完善的解决方案,但参考文档较为复杂,学习成本高。

Spring Security OAuth 同时提供了对 OAuth1.0 与 OAuth2.0 的支持,支持 OAuth2.0 协议的所有特性与要求(Authorization Server, Resources Server, Client)。同时与 Spring 有着良好的集成,相比其他 OAuth2.0 的 Java 语言实现具有较大优势。基于 Spring 的注解与 xml 配置,使得 OAuth2.0 协议能够完全融合于 Spring 框架<sup>[11]</sup>。

即使 OAuth2.0 协议简化了开放平台授权,但仍然存在一定的工作量与复杂度。Spring 框架整合 OAuth2.0 协议,极大地简化了 OAuth2.0 协议实现的复杂度。Spring 框架中的 Spring Security 组件用于实现用户认证登陆,OAuth2.0 用于实现开放平台授权。两者结合简化了 OAuth2.0 服务器的开发。

OAuth2.0 服务端需要提供访问资源服务器的接口。共需要三个接口:客户端平台从资源所有者(用户)取得资源授权接口,客户端根据授权码向资源服务器获取 Access Token 接口,客户端根据 Access Token 获取资源接口。其中需要服务端提供两个页面:授权页面与登陆页面<sup>[12]</sup>。

Spring Security 组件集成了对 OAuth2.0 的支持,使得页面授权逻辑与 OAuth2.0 的协议逻辑相分离。Spring MVC 负责页面跳转与展示, Spring Security 负责安全认证, OAuth2.0 负责客户端、用户与服务器之间的关系处理<sup>[13]</sup>。

3 资源服务器数据分析

通过对个人诚信系统管理的分析,存在资源服务器上的信息包括用户注册信息、用户基本档案信息与诚信档案信息三个部分。其中,用户基本档案信息与用户诚信档案信息为该平台对外开放的信息。

用户注册信息包含用户 ID、用户名和密码等关键数据。用户基本档案包含用户的信用记录、信用状态以及信用得分等。用户基本档案记录着用户的实名认证等详细信息。用户的基本档案以及诚信档案信息必须在用户 ID、密码匹配的情况下才可以被访问。

4 Spring OAuth2.0 系统设计

Spring 采用插件化的方法来集成 Spring Security 与 OAuth2.0。构建该系统同时需要引入相应的 Spring Security 与 spring-security-oauth2 的 jar 包。Spring Security 通过在 xml 中配置 http 节点设置相应 URL 的访问权限,采用 Spring 框架的 OAuth2.0 的授权管理器。其原理如图 2 所示。

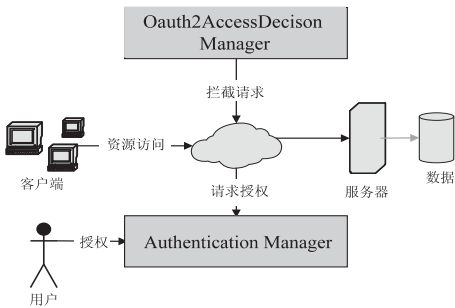


图 2 Spring OAuth2.0 授权原理

第三方平台首先尝试获取位于资源服务器的资源,若资源是该平台开放资源,OAuth2 Access Decision Manager 会判断当前第三方是否经过本平台用户授权。没有授权则转到登陆页面提示用户登陆,若已经授权则资源服务器将相应的数据返回给第三方平台。

同时, Spring 提供了基于 xml 配置的 OAuth: authorization-server,该节点用于自动解析创建 OAuth2.0 应用所需要创建的类与必须的配置项<sup>[14]</sup>。

OAuth2.0 核心流程所需要的数据均保存在认证服务器数据库中。每一个需要接入的第三方系统都需要在认证服务器注册用于标示该应用信息,认证服务器的第三方应用信息数据见表 1。

框架 Spring 为 OAuth2.0 提供了 Access Token 的实现类 JdbcTokenStore,用于实现令牌的存储。同时令牌拥有有效期, Spring Security 提供了令牌刷定时刷新方法。JdbcTokenStore 用于读写存在于数据库中的令牌数据。 Access Token 数据表包含 accesstoken、create-time 与 expiretime 三个字段,分别表示令牌、创建时间与过期时间等详细信息。

表 1 AppInfo 应用信息表

字段	说明
appid	客户端 id
appsceret	App 密钥
appname	App 名称
appowner	App 所属机构
appuri	App 回调安全 URI
appcreate-time	App 创建时间
appstatus	App 状态

在系统设计的最后阶段,需要向外部暴露自己的开放接口。该诚信系统只需要暴露两个开放接口,即调用基本信息接口与调用基本信息扩展的诚信信息接口。

SpringSecurity 集成 OAuth2.0 需要实现的步骤如下:

Spring Authorization Server 授权服务器端的配置。该配置用于创建授权服务器相关的支持类。 <authorization-server/>以及与其相关的注解配置,用于配置客户端详细服务信息,授权服务器令牌服务,以及授权服务器端点等。

Spring Resource Server 的服务端配置。该配置用于创建资源服务器相关的支持类。其作用是提供授权的 Servlet 过滤器用来保护 Web 资源。该项配置可以使用基于 XML 的配置<resource-server/>。

Spring OAuth2.0 Client 的配置,该配置用于创建可以存储当前请求和上下文环境的 Servlet 过滤器,用来管理 OAuth 授权 URI 的重定向。该项配置可以使用基于 XML 的配置<client/>。

5 系统测试

当第三方应用需要使用用户的个人信息档案或者个人基本信息档案时,就可以接入本诚信档案系统。接着页面跳转到授权服务器提供授权页面。用户输入有效的账户密码并同意授权。用户授权页面如图 3 所示。



图 3 用户授权页面

同意授权后,根据客户端应用提供的回调地址,将

当前页面跳转到客户端应用页面,此时客户端应用可以根据获取到的 code,再向授权服务器获取 Access Token。授权成功跳转页面如图 4 所示。



图 4 授权成功跳转页面

客户端应用根据 Access Token 可以得到的信息如图 5 所示。客户端可以得到用户所允许的个人基本信息及信用状态信息。由于篇幅,更多详细信息未列出。



图 5 获得用户信息页面

## 6 结束语

基于 Spring Security,配置了第三方授权平台的服务端。完成了基础的授权功能,实现了用户资源的第三方平台授权访问,保证了第三方授权的安全性。

在框架的支持下,Web 服务端开发人员仅仅需要

配置<authorization-server/>等几个参数就可以完成 OAuth2.0 服务器的相关协议开发。一定程度上简化了 OAuth2.0 协议的应用与开发,有利于 OAuth2.0 协议在互联网中的推广。

### 参考文献:

- [1] 刘大红,刘 明. 第三方应用与开放平台 OAuth 认证互连技术研究[J]. 电脑知识与技术,2012,8(8):5367-5369.
- [2] 张卫全,胡志远. 浅析作用于 Web2.0 安全防范的 OpenID 和 OAuth 机制[J]. 通信管理与技术,2011(2):15-18.
- [3] 王春枝,唐俊武. 关于 IoT 模式及轻量级容器的研究[J]. 湖北工业大学学报,2006,21(4):52-54.
- [4] 张 宇,王映辉,张翔南. 基于 Spring 的 MVC 框架设计与实现[J]. 计算机工程,2010,36(4):59-62.
- [5] Leiba B. OAuth web authorization protocol[J]. IEEE Internet Computing,2012,16(1):74-77.
- [6] Jones M, Hardt D. The OAuth 2.0 authorization framework: bearer token usage[R]. [s.l.]:[s.n.],2012.
- [7] Hardt D. The OAuth 2.0 authorization framework[R]. [s.l.]:[s.n.],2012.
- [8] 张 锐,张建林,孙国忠. 多业务系统的统一认证授权研究与设计[J]. 计算机工程与设计,2009,30(8):1826-1828.
- [9] 蒋 伟,马光思. Spring 与其他框架整合及流程分析[J]. 计算机工程,2007,33(14):79-81.
- [10] 庄少焯. 基于 Spring 的轻量级 Web 框架研究与实现[D]. 成都:电子科技大学,2009.
- [11] 黄道斌. 一种基于 SPRINGSECURITY 的访问控制方案[J]. 软件导刊,2011,10(8):122-123.
- [12] 丁振凡. 基于 Spring Security 的 Web 资源访问控制[J]. 宜春学院学报,2012,34(8):71-74.
- [13] 陈雄华. Spring 3.x 企业应用开发实战[M]. 北京:电子工业出版社,2012.
- [14] 肖 云. 基于 Spring Security 安全的 Web 应用开发[J]. 计算机与现代化,2011(6):158-159.

(上接第 166 页)

- [8] Zhou Xin, Zhong Zhangdui, Zhang Bei, et al. Experimental characterization and correlation analysis of indoor channels at 15GHz[J]. International Journal of Antennas and Propagation,2015(1):1-11.
- [9] 王 萍,勾天杭,李朋朋,等. 室内走廊环境高频段宽带无线信道测量与建模[J]. 电波科学学报,2012,27(3):496-500.
- [10] 秦 成,陈 豪. 无线信道大尺度传播效应的统计模型与统计方法[J]. 移动通信,2009,33(12):22-26.
- [11] Samimi M, Wang Kangping, Rappaport T S. 28 GHz angle of arrival and angle of departure analysis for outdoor cellular communications using steerable beam antennas in New York City[C]//77th vehicular technology conference. [s.l.]: IEEE,2013:1850-1859.

- [12] Azar Y, Wong G N, Wang Kangping, et al. 28 GHz propagation measurements for outdoor cellular communications using steerable beam antennas in New York city[C]//IEEE international conference on communications. [s.l.]:IEEE,2013:5143-5147.
- [13] Maccartney G R, Zhang Junhong, Nie Shuai, et al. Path loss models for 5G millimeter wave propagation channels in urban microcells[C]//Global communications conference. [s.l.]: IEEE,2013:3948-3953.
- [14] Rappaport T S, Gutierrez F. Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications[J]. IEEE Transactions on Antennas and Propagation,2013,61(4):1850-1859.