博学谷
www.boxuegu.com

第八章 Spring Cloud 用户认证与授权

一样的在线教育，不一样的教学品质

目录 Contents

OAuth2 原理

小节导学

OAuth2 是用来解决什么问题的?

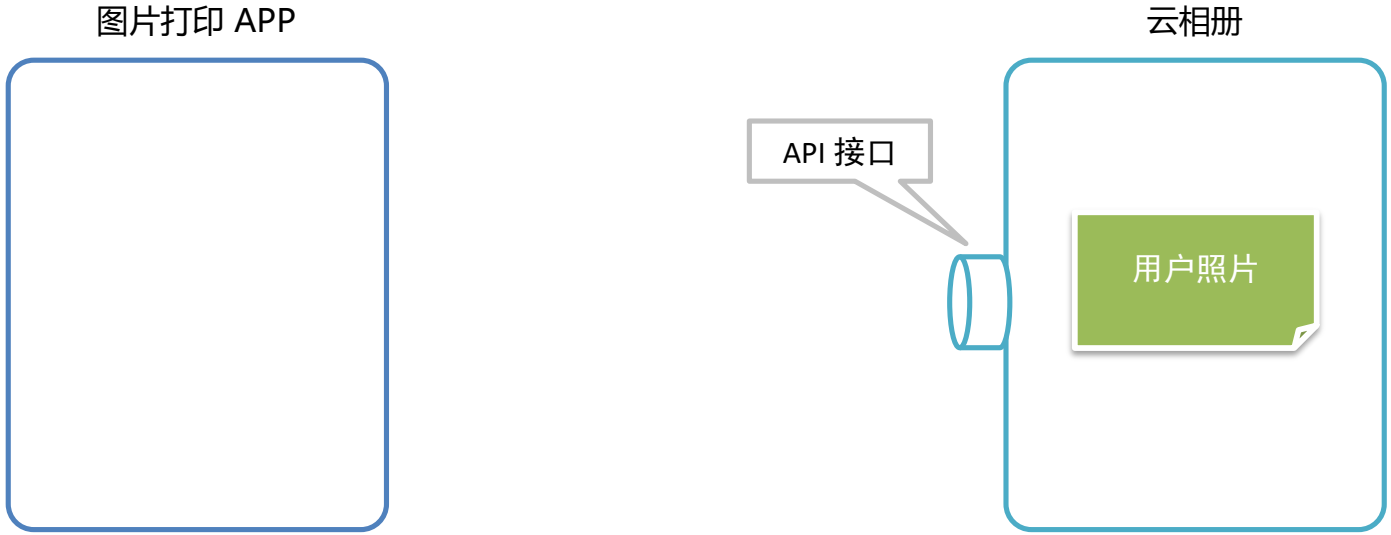
OAuth2 具体是怎么工作的?

明白了这些问题，才能做好实际开发，本节我们就要解决这些问题。

- OAuth2 工作流程
- OAuth2 核心概念
- OAuth2 经典模式

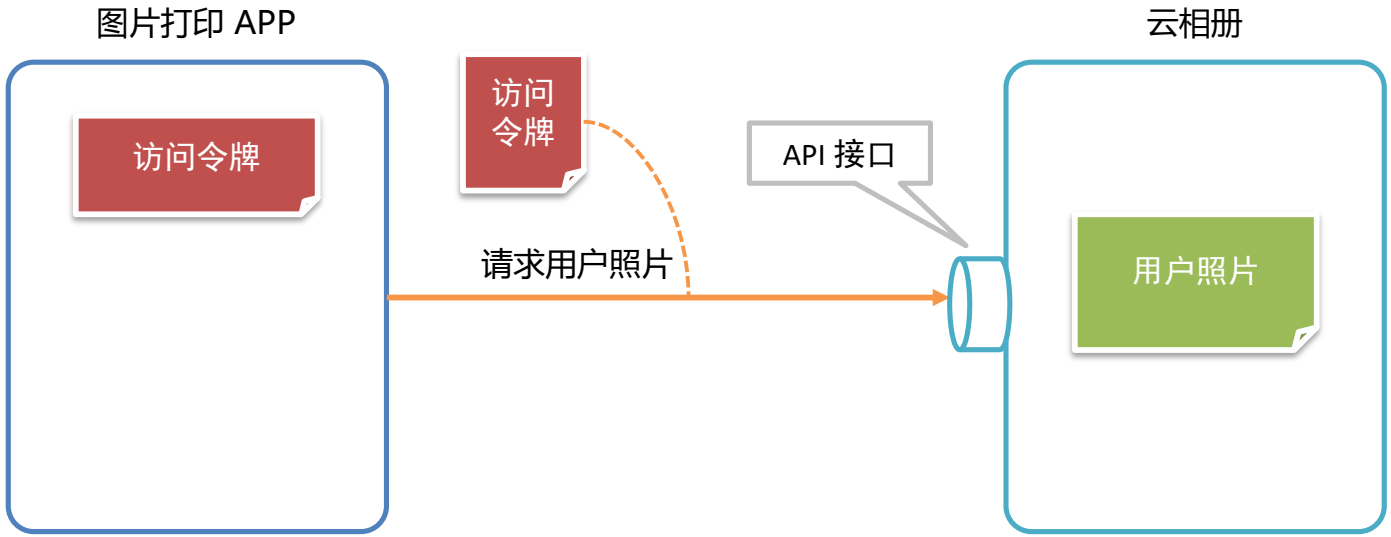
1. OAuth2 工作流程

图片打印 APP 如何拿到用户在云相册中的图片?



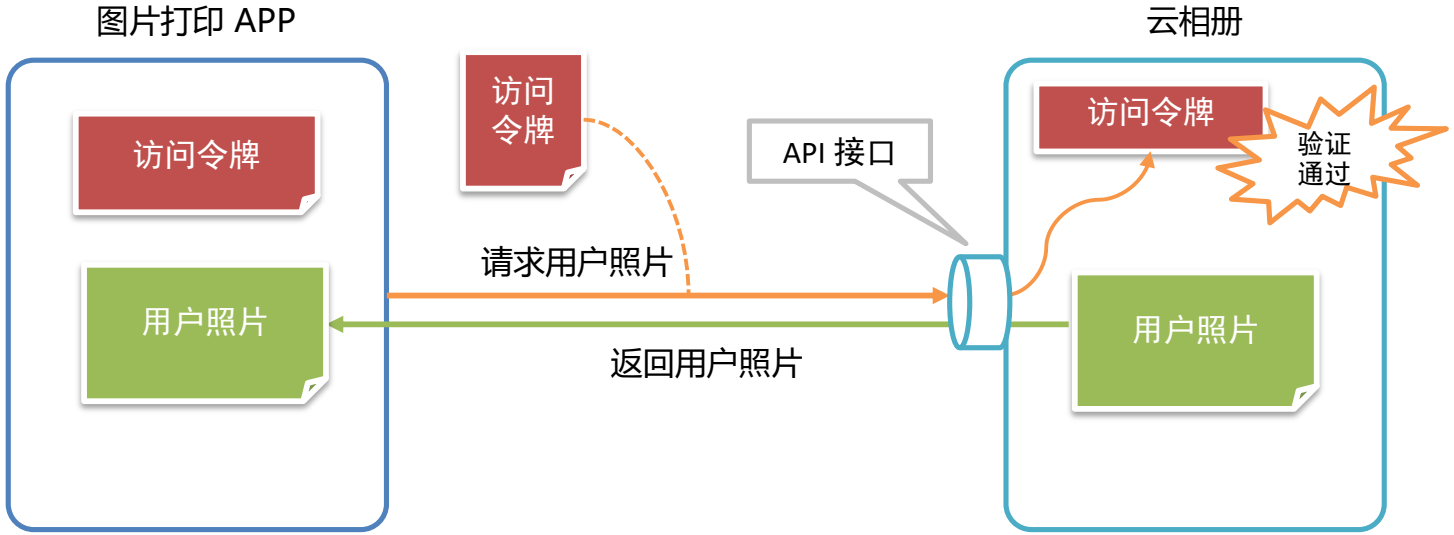
1. OAuth2 工作流程

最佳实践：**Token 访问令牌**，表示客户端已经被授予了访问用户数据的权限。



1. OAuth2 工作流程

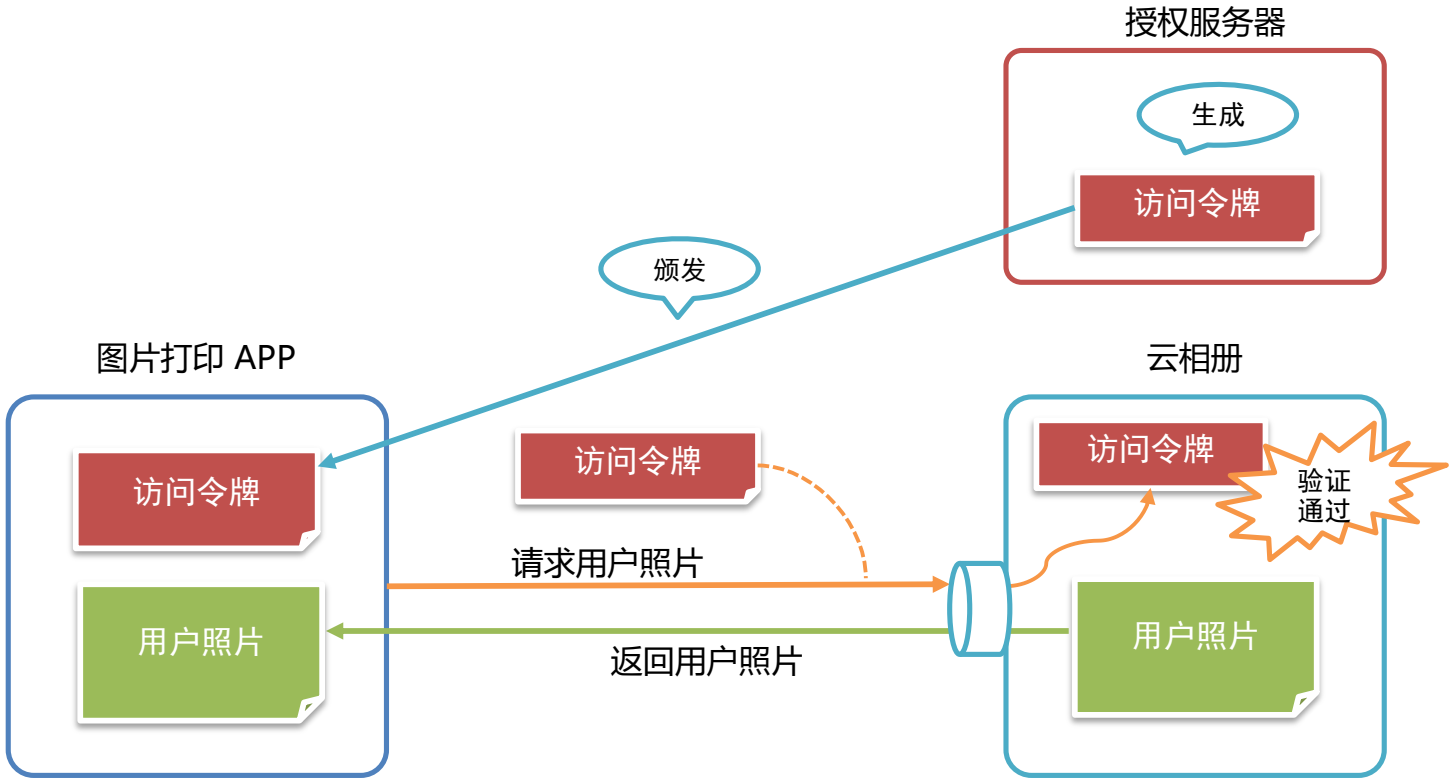
- 最佳实践：Token 表示客户端已经被授予了访问用户数据的权限。



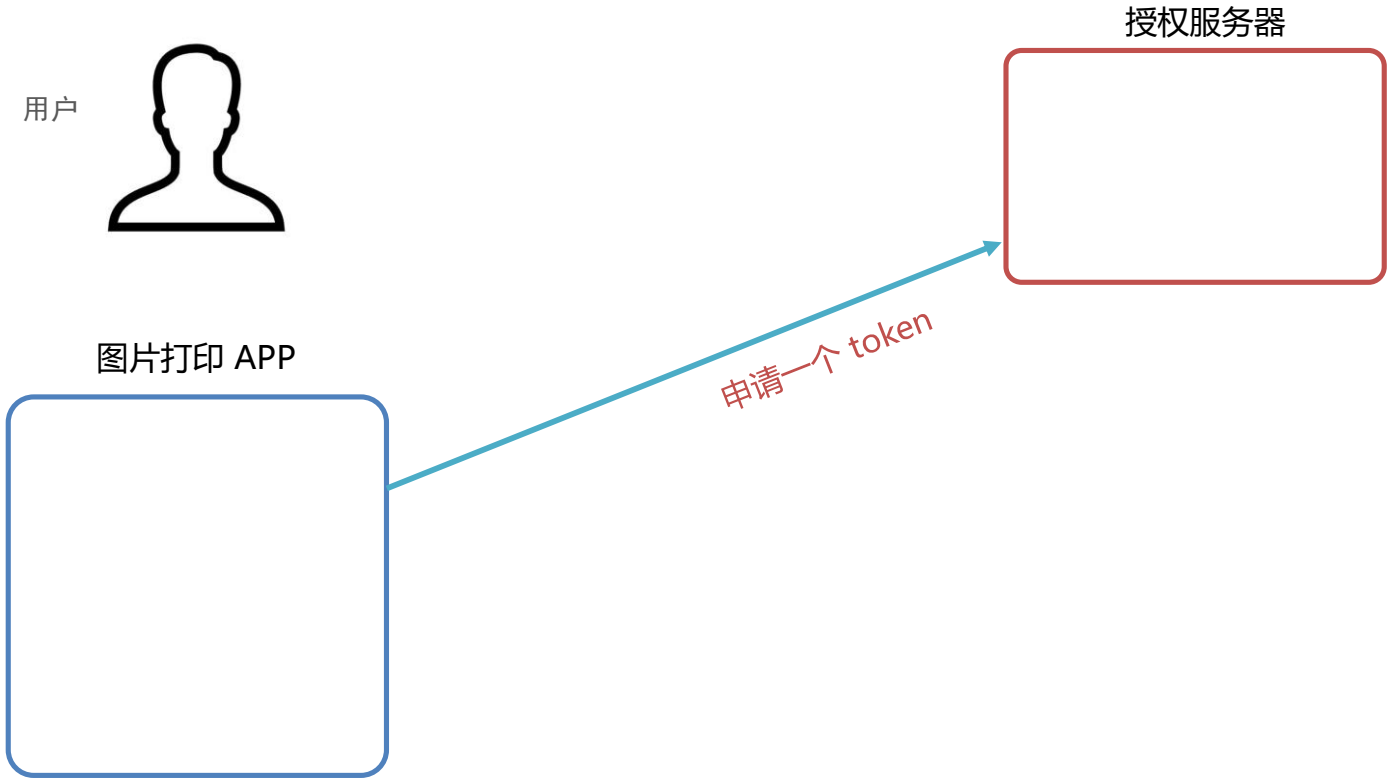
1. OAuth2 工作流程

有一个大问题

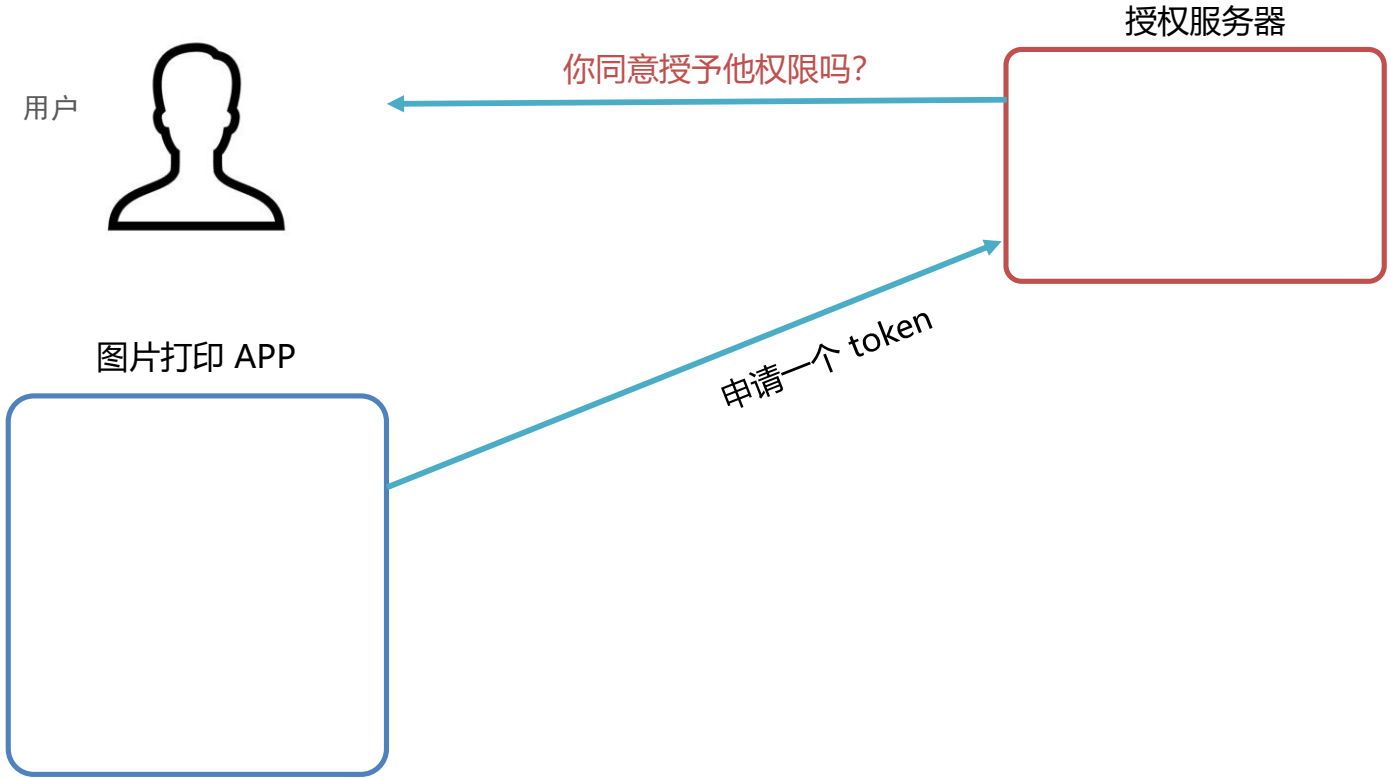
1. OAuth2 工作流程



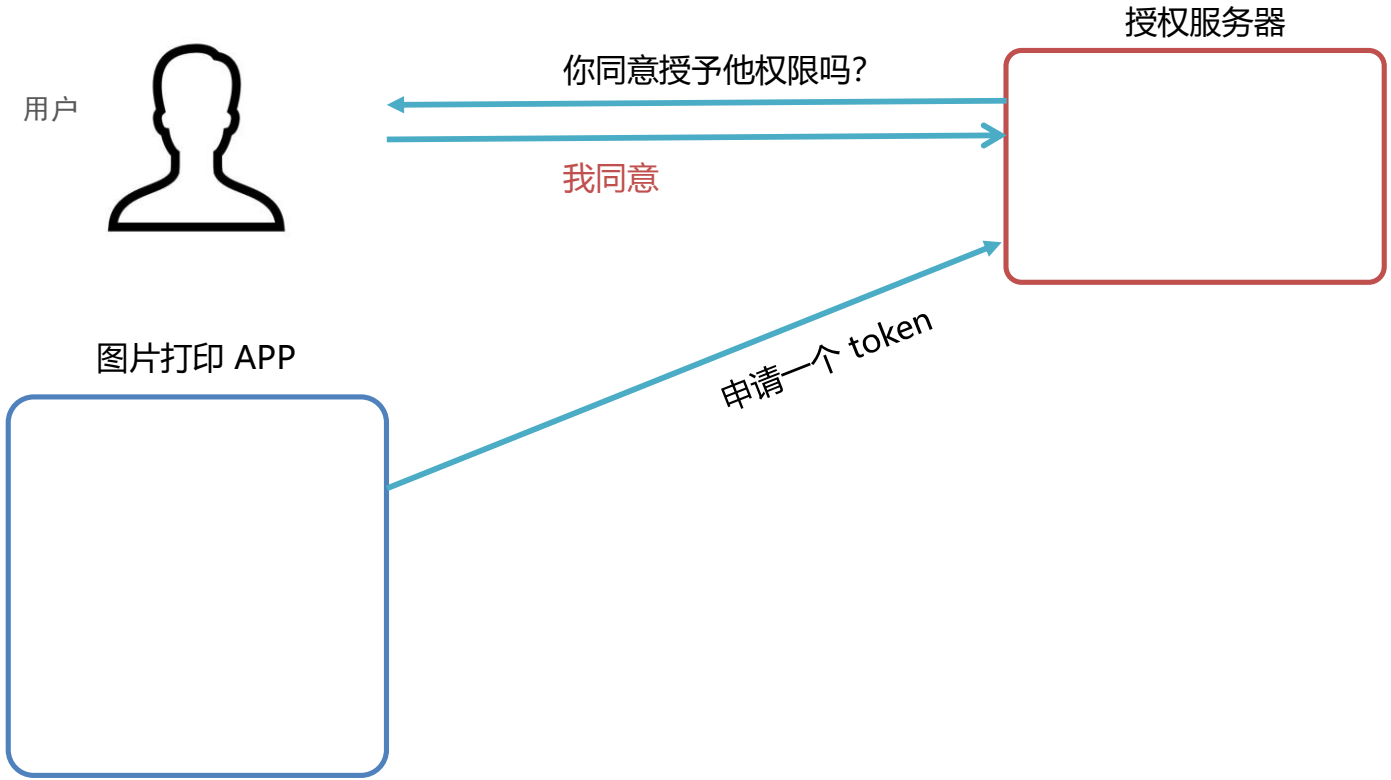
1. OAuth2 工作流程



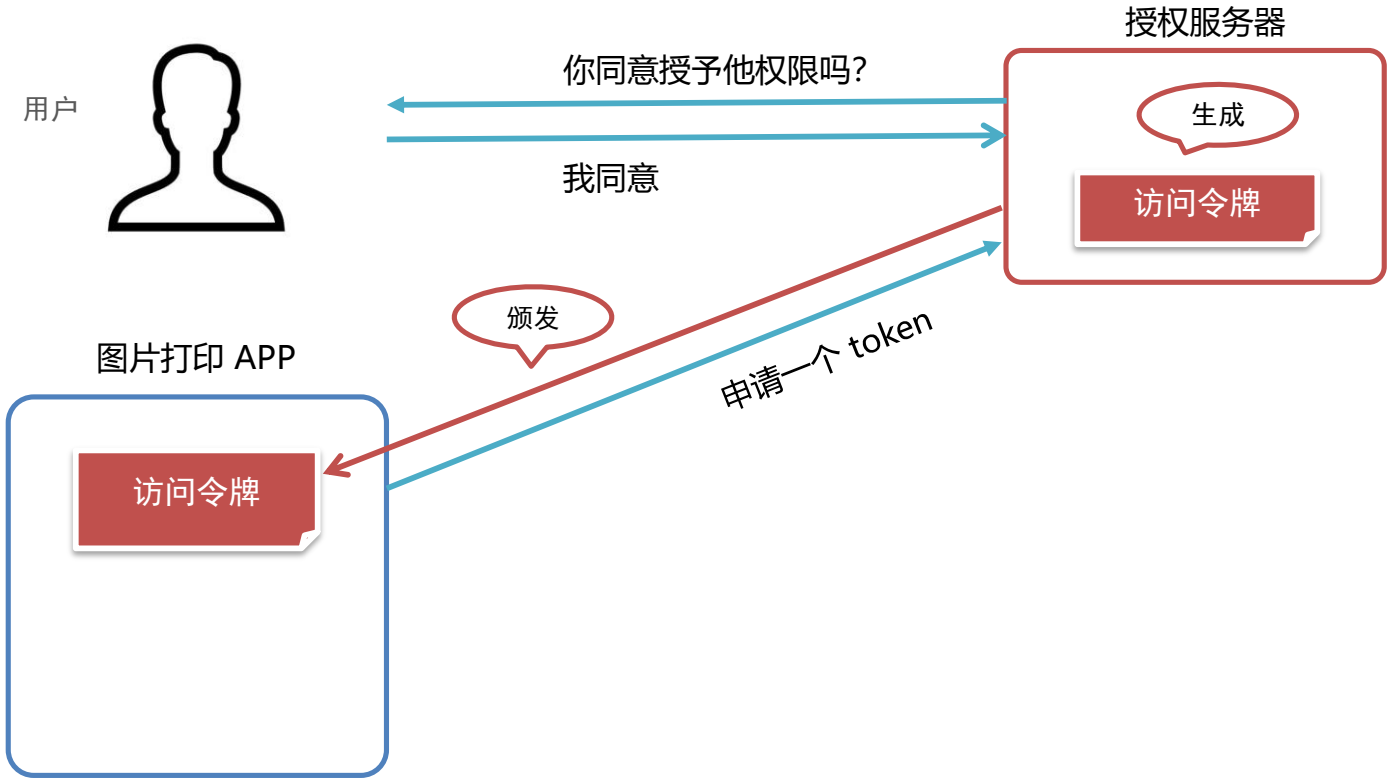
1. OAuth2 工作流程



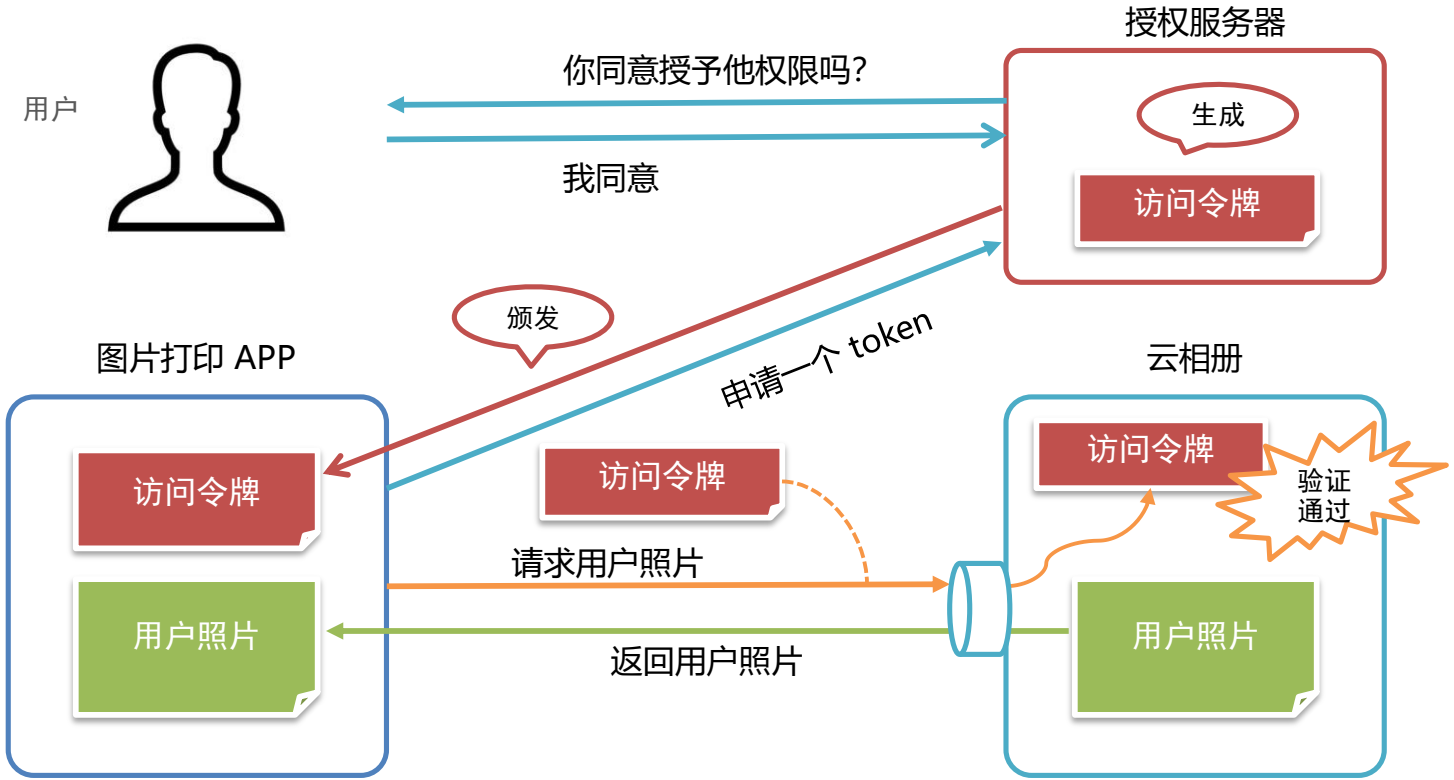
1. OAuth2 工作流程



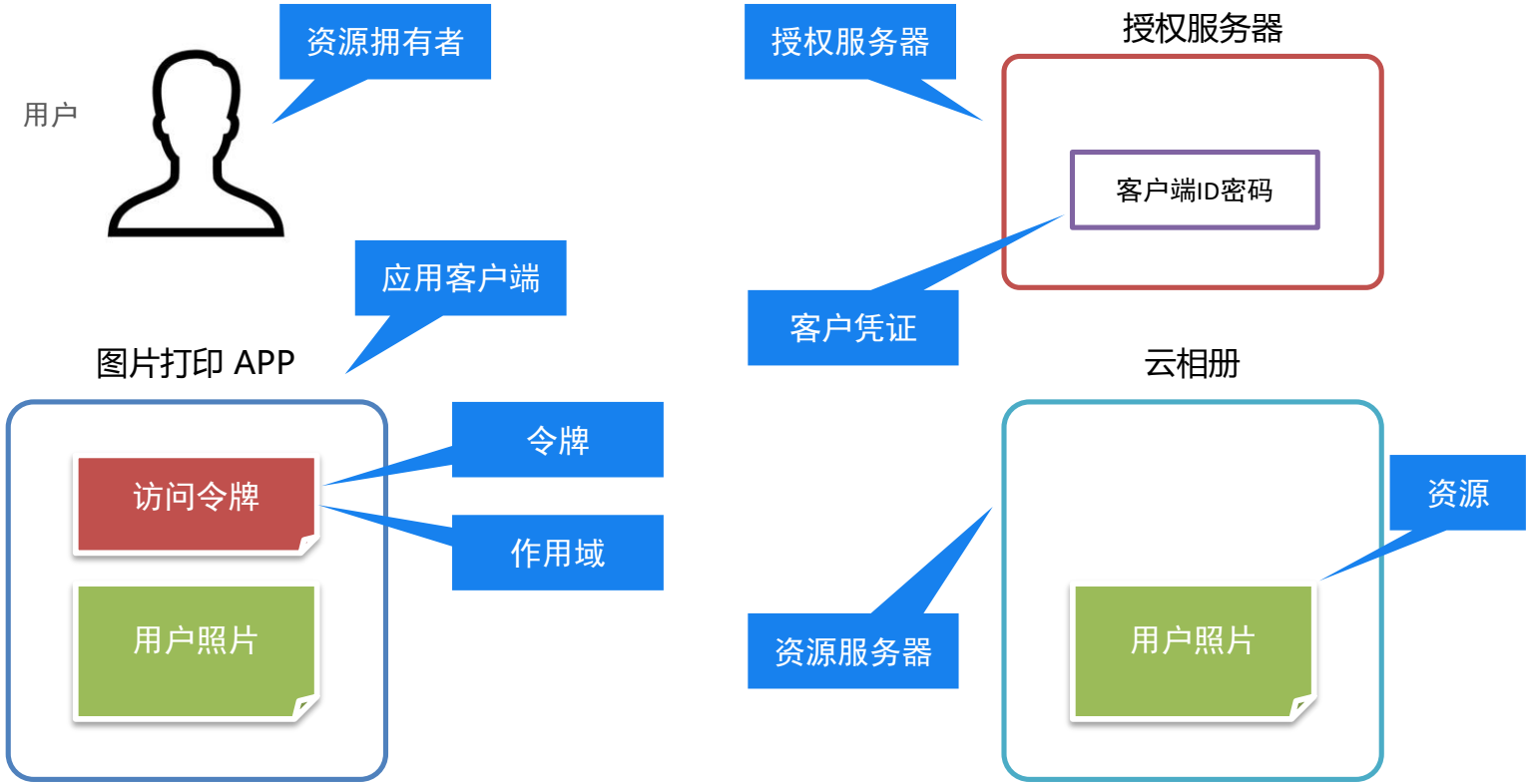
1. OAuth2 工作流程



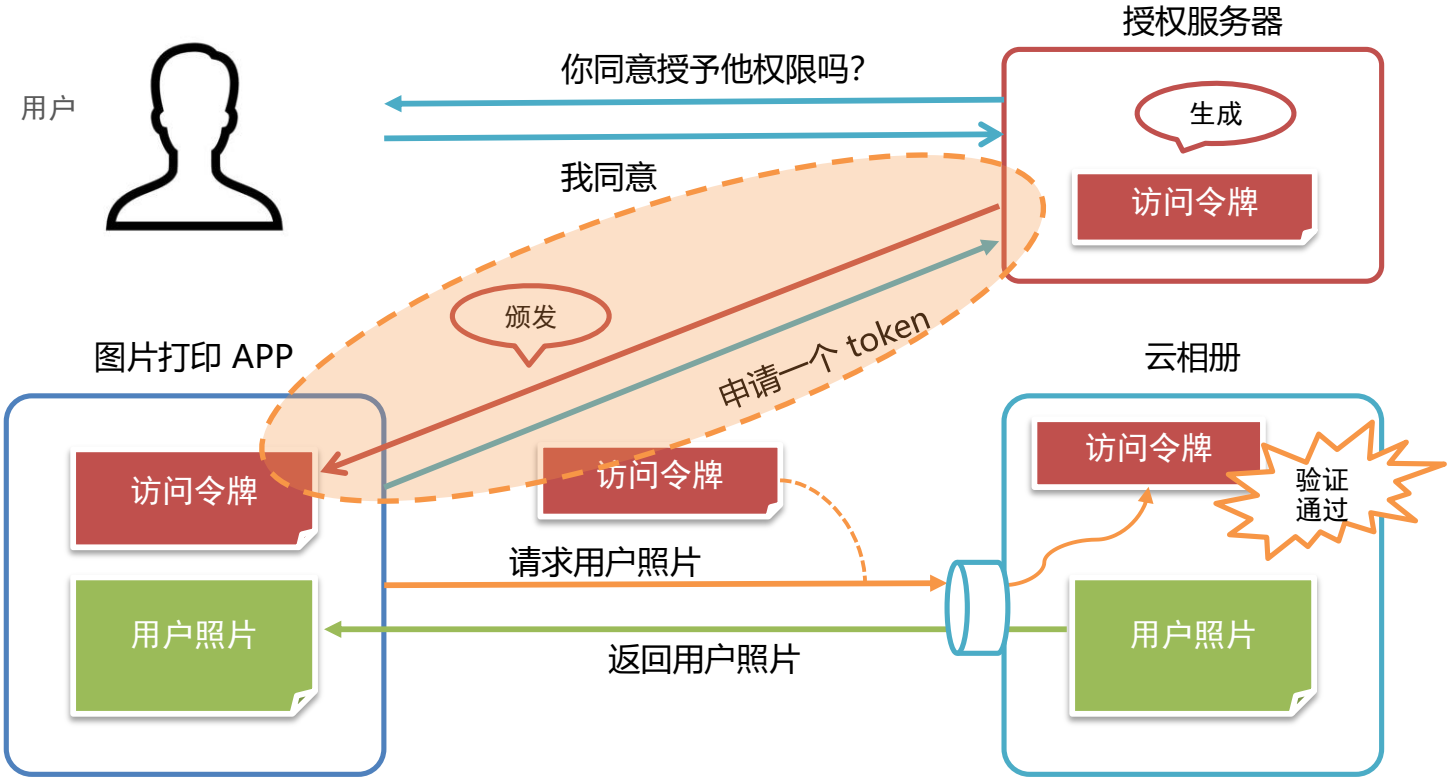
1. OAuth2 工作流程



2. OAuth2 核心概念



3. OAuth2 经典模式



3. OAuth2 经典模式

OAuth2 原理-总结

重难点

1. OAuth2 工作流程
2. OAuth2 核心概念
3. OAuth2 的4种经典模式

OAuth2 原理-总结

重难点

1. OAuth2 工作流程
2. OAuth2 核心概念
3. OAuth2 的4种经典模式

下节

挑战 **授权码模式**

应用最广，开发**复杂度最高**

博学谷
www.boxuegu.com

目录 Contents

OAuth2 授权码模式

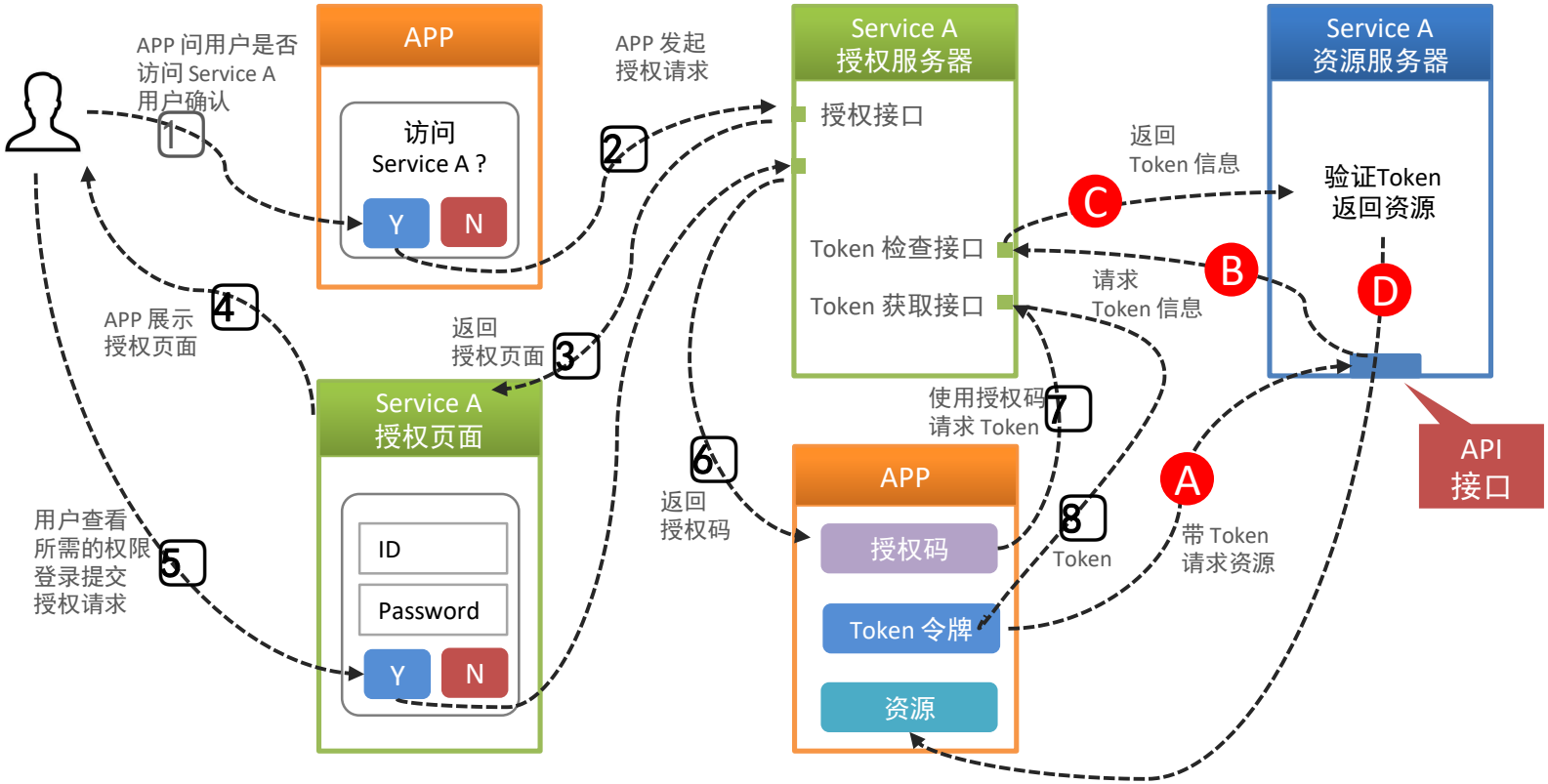
小节导学

授权码模式是4个模式中最复杂的，也是应用最为广泛的，我们先从这个模式学起，这个明白后其余3种模式就很好理解了。

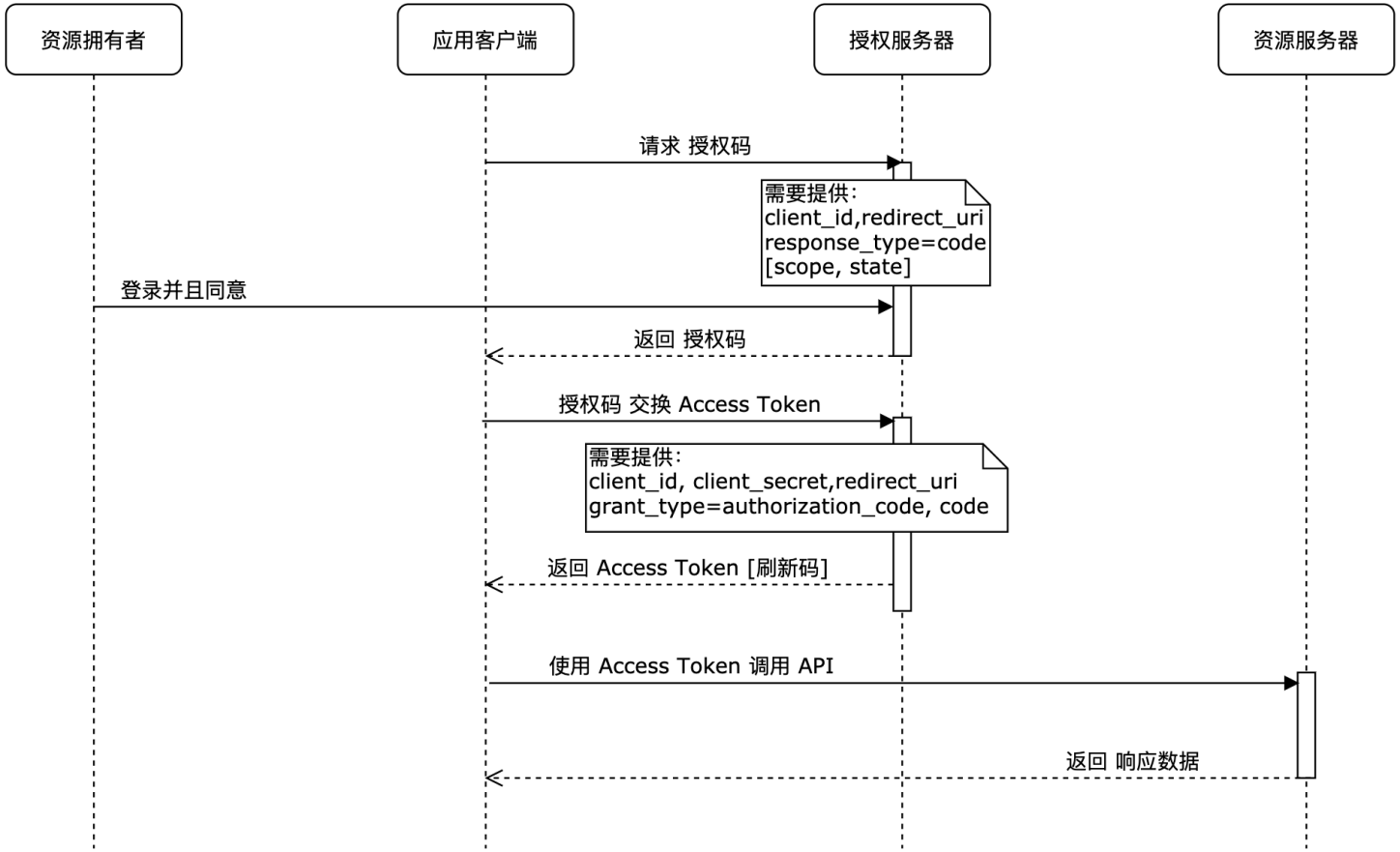
本节的目标就是弄明白“授权码模式”的工作流程，以及开发方法。

- 授权码模式的工作流程
- 授权码模式开发实践

1. 授权码模式的工作流程

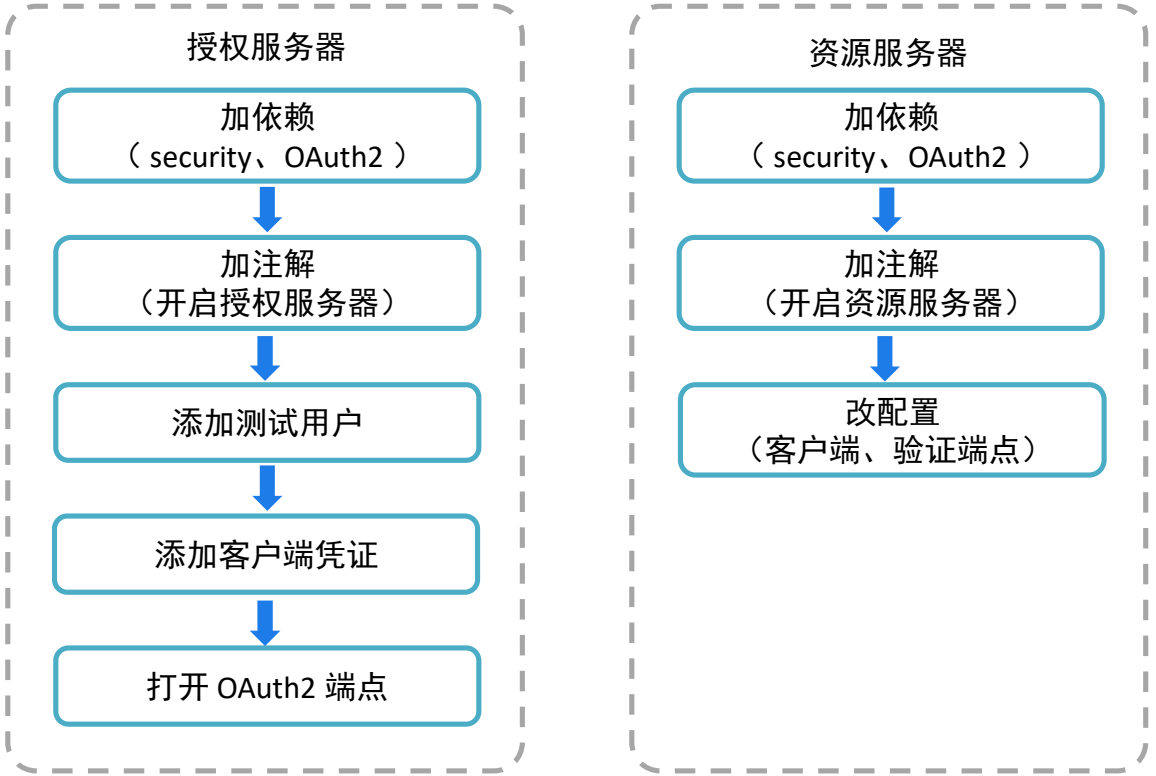


1. 授权码模式的工作流程



2. 授权码模式开发实践

开发流程



OAuth2 授权码模式-总结

重难点

1. 授权码模式的工作流程
2. 授权码模式的开发流程

OAuth2 授权码模式-总结

重难点

1. 授权码模式的工作流程
2. 授权码模式的开发流程

下节

搞懂**简化模式**，怎么简化的？和授权码模式有什么区别？

博学谷
www.boxuegu.com

目录 Contents

OAuth2 简化模式

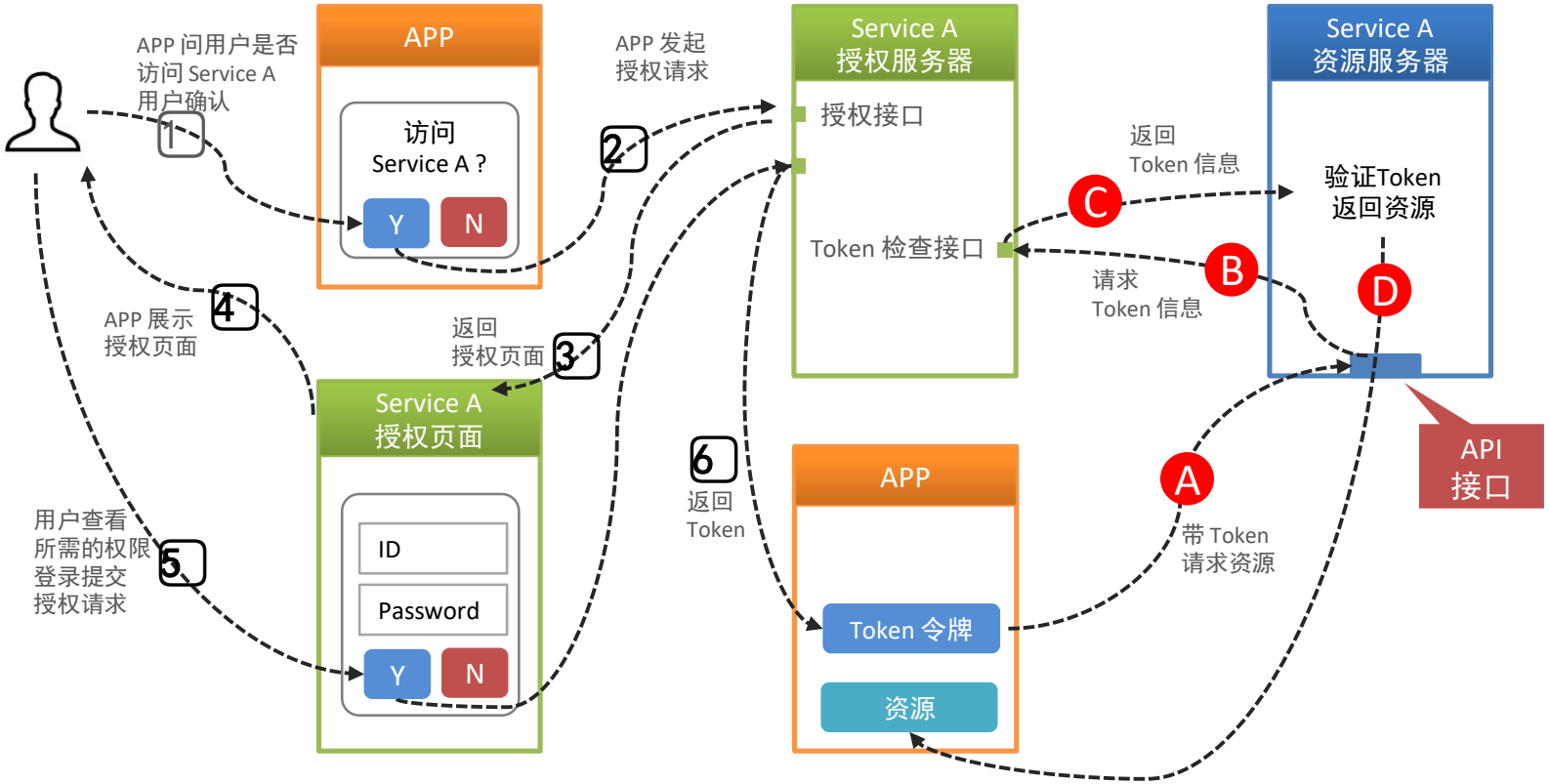
小节导学

简化模式是对授权码模式的简化，**把“授权码”去掉了。**

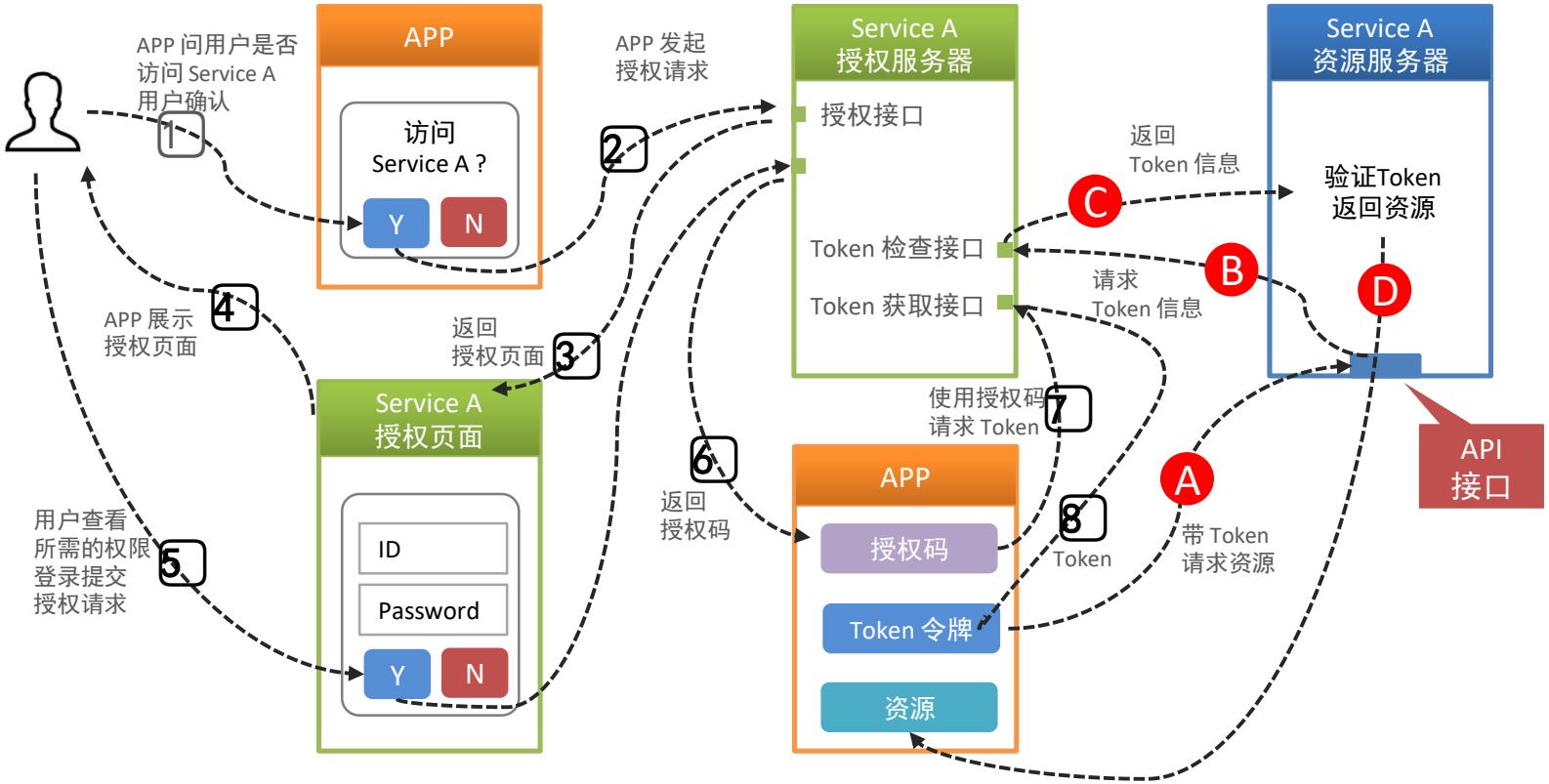
本节的目标就是弄明白“简化模式”的工作流程，以及开发方法。

- 简化模式的工作流程
- 简化模式开发实践

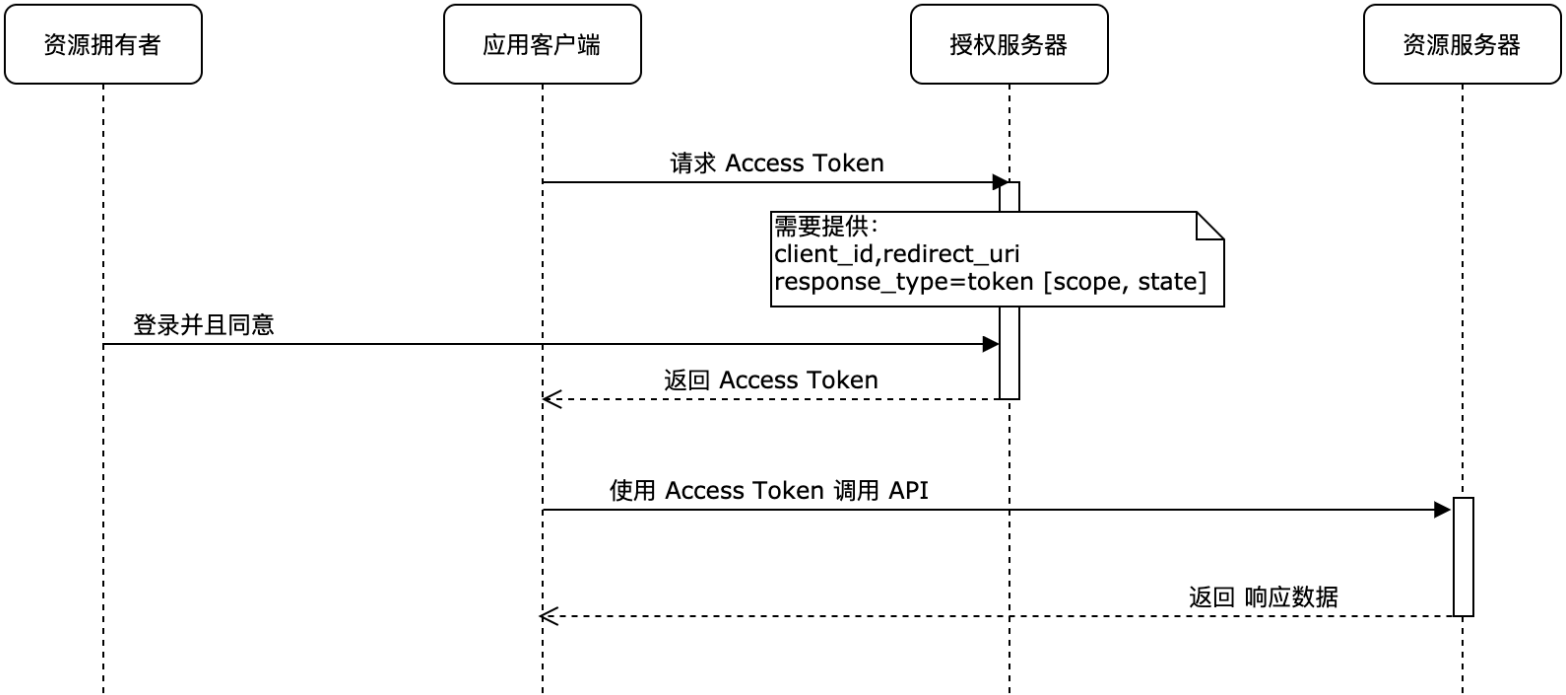
1. 简化模式的工作流程


APP 问用户是否
访问 Service A
用户确认
1
APP
访问
Service A ?
Y
N
APP 发起
授权请求
2
Service A
授权服务器
授权接口
Token 检查接口
返回
Token 信息
C
Service A
资源服务器
验证Token
返回资源
请求
Token 信息
B
D
API
接口
APP 展示
授权页面
4
返回
授权页面
3
Service A
授权页面
ID
Password
Y
N
用户查看
所需的权限
登录提交
授权请求
5
6
返回
Token
APP
Token 令牌
资源
A
带 Token
请求资源

授权码模式的工作流程（用于对照）

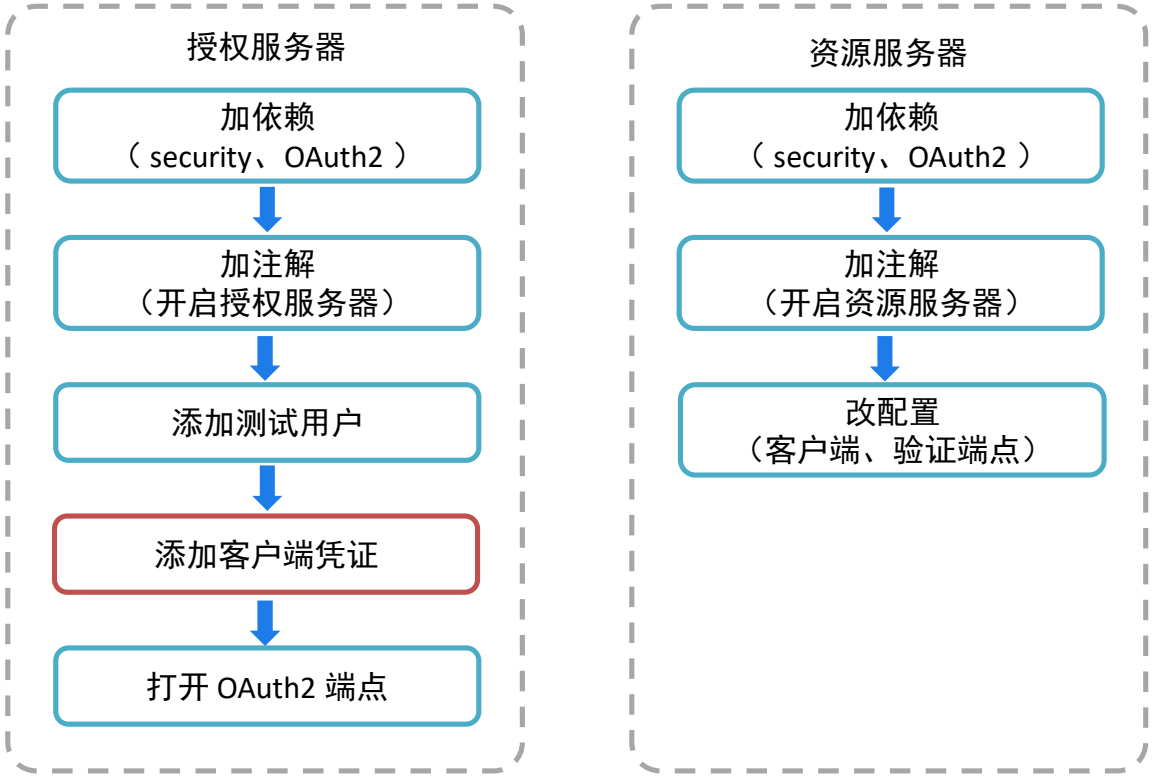


1. 简化模式的工作流程


资源拥有者
应用客户端
授权服务器
资源服务器
请求 Access Token
需要提供:
client_id,redirect_uri
response_type=token [scope, state]
登录并且同意
返回 Access Token
使用 Access Token 调用 API
返回 响应数据

2. 简化模式开发实践

开发流程



OAuth2 简化模式-总结

重难点

1. 简化模式的工作流程
2. 简化模式的开发流程

OAuth2 简化模式-总结

重难点

1. 简化模式的工作流程
2. 简化模式的开发流程

下节

密码模式，比简化模式更简单，可以轻松一下了

目录 Contents

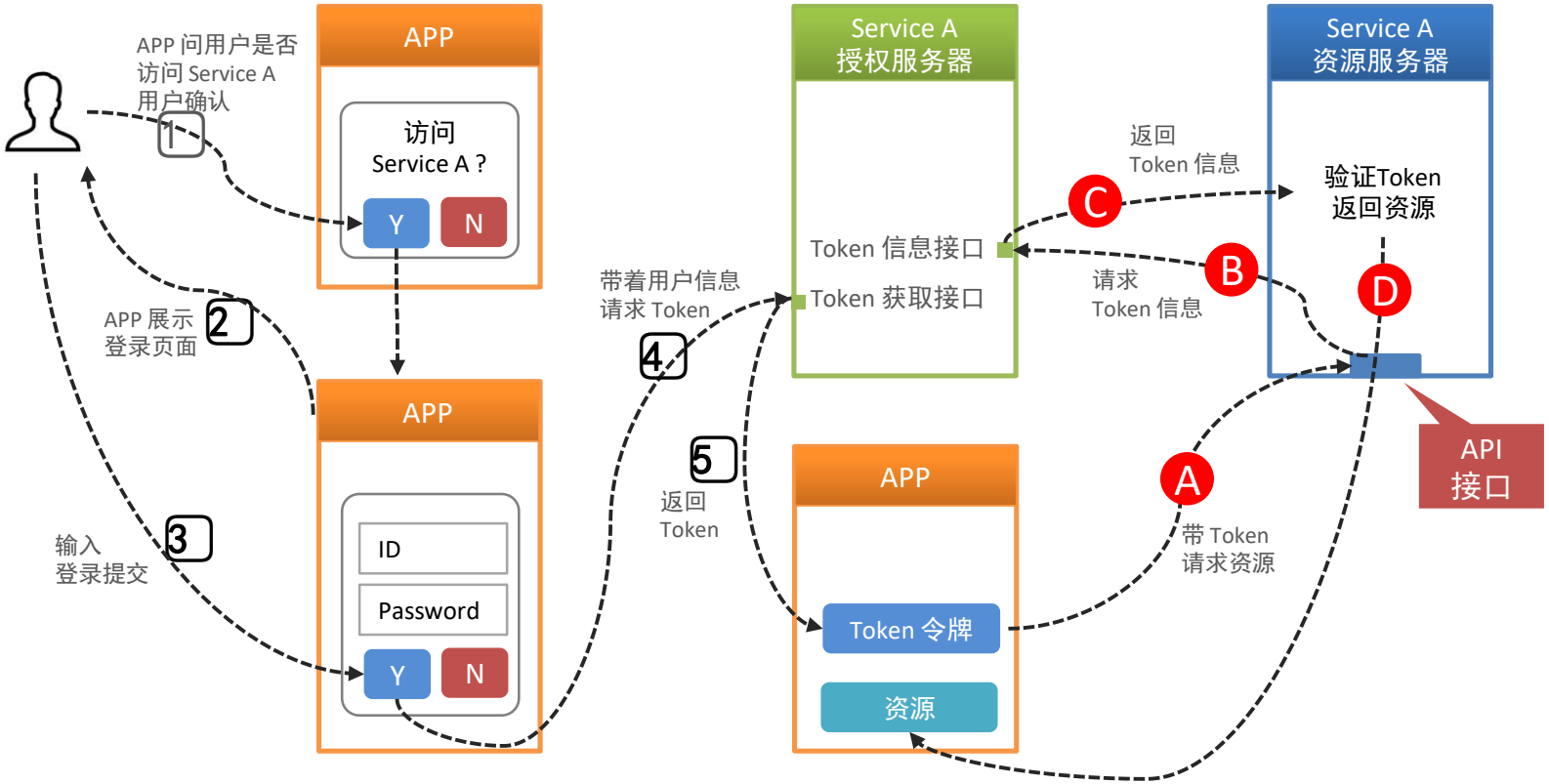
OAuth2 密码模式

小节导学

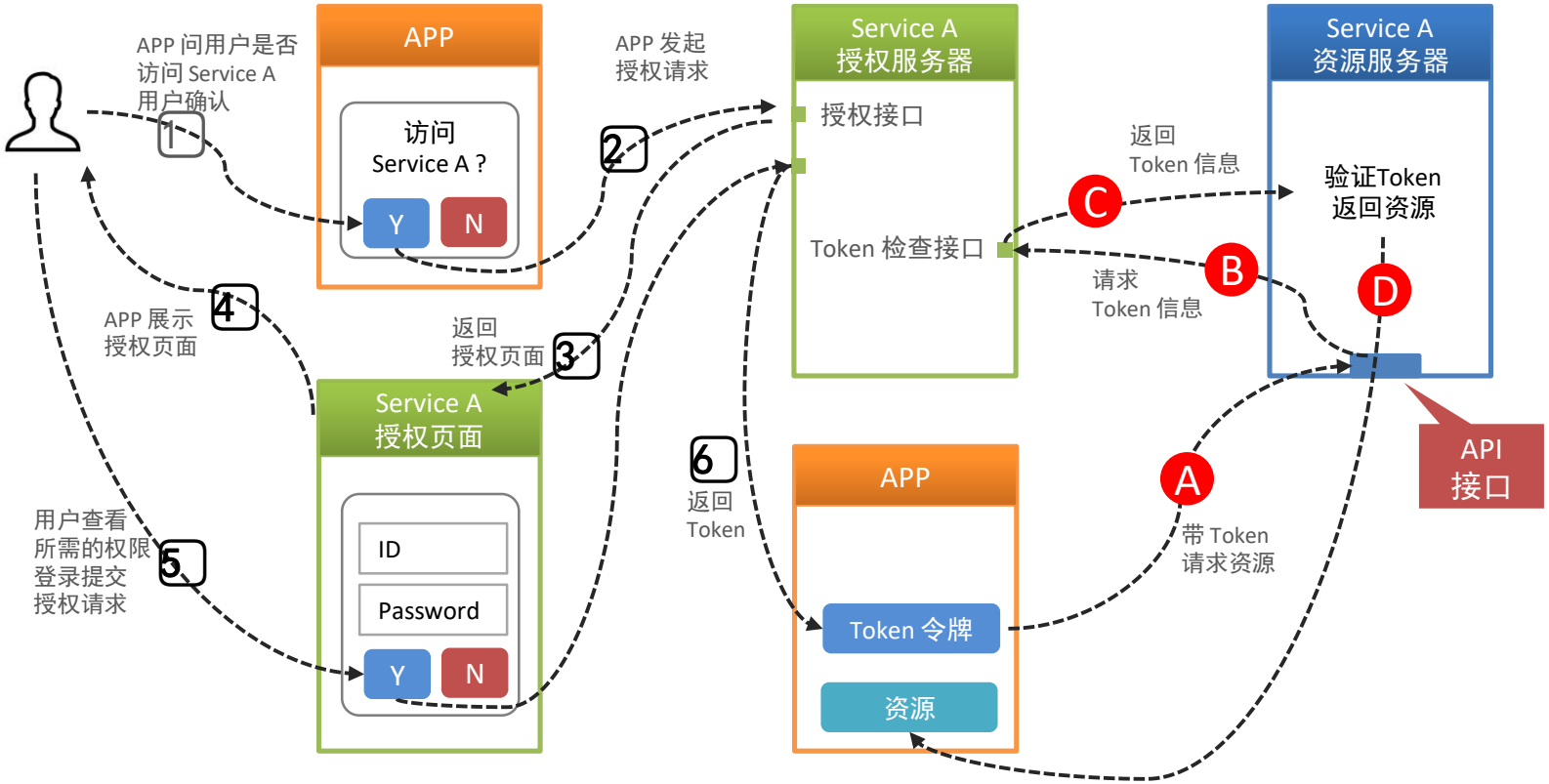
本节的目标就是弄明白“密码模式”的工作流程，以及开发方法。

- 密码模式的工作流程
- 密码模式开发实践

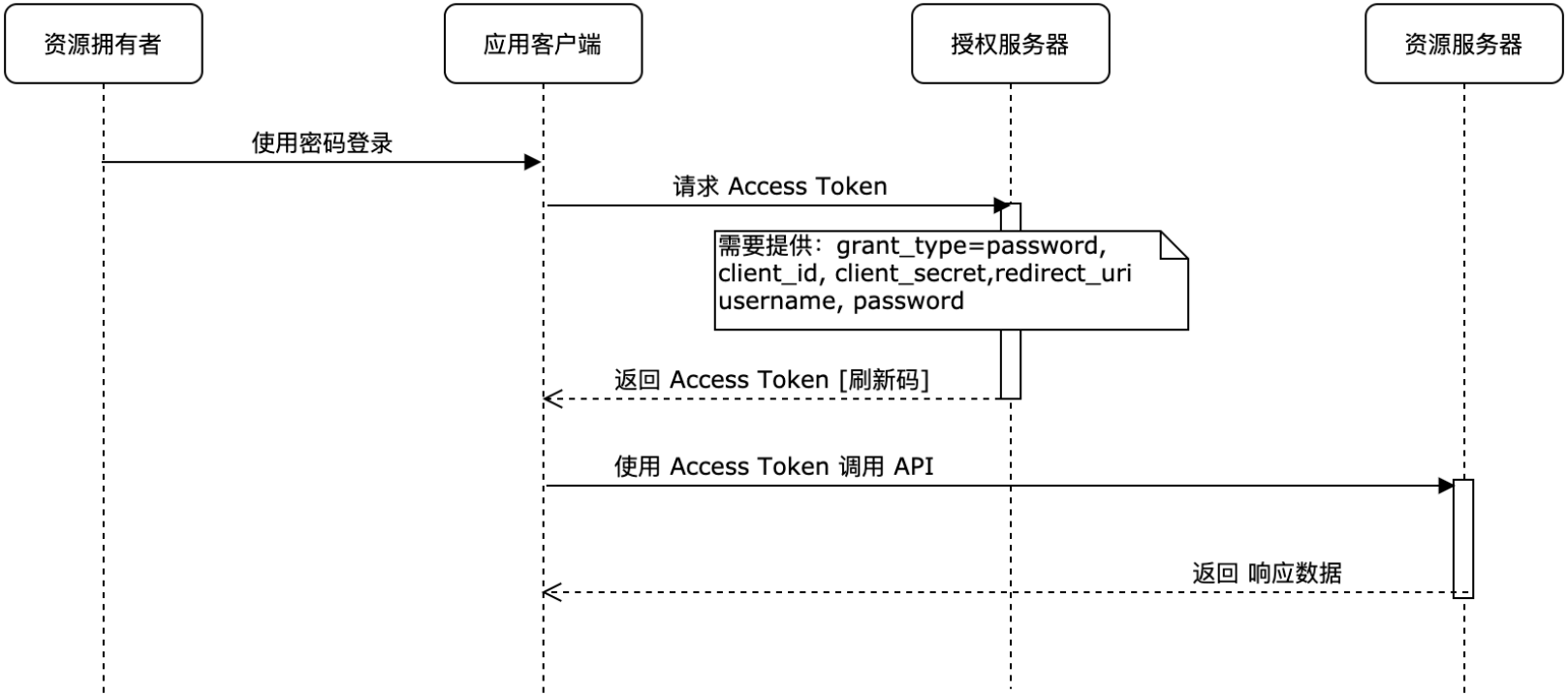
1. 密码模式的工作流程


APP 问用户是否
访问 Service A
用户确认
1
APP
访问
Service A ?
Y
N
APP 展示
登录页面
2
APP
ID
Password
Y
N
输入
登录提交
3
带着用户信息
请求 Token
4
返回
Token
5
Service A
授权服务器
Token 信息接口
Token 获取接口
APP
Token 令牌
资源
返回
Token 信息
C
请求
Token 信息
B
Service A
资源服务器
验证Token
返回资源
D
API
接口
A
带 Token
请求资源

简化模式的工作流程（用于对照）

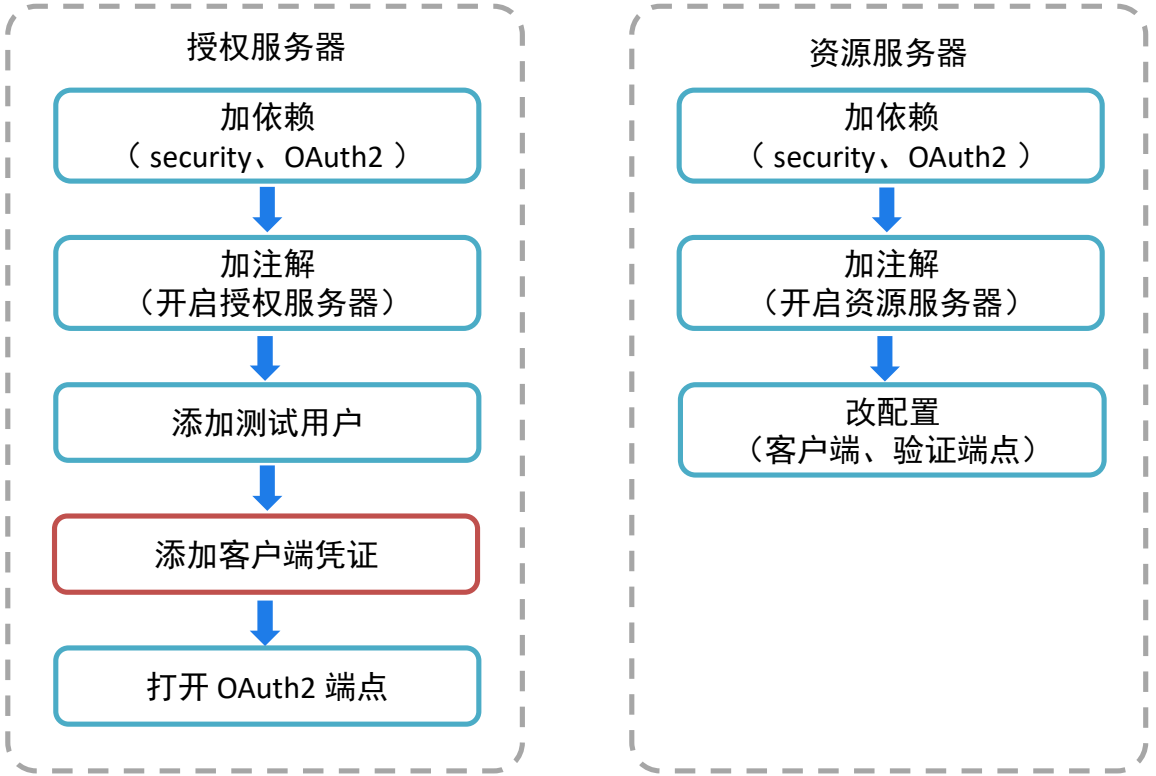


1. 密码模式的工作流程


资源拥有者
应用客户端
授权服务器
资源服务器
使用密码登录
请求 Access Token
需要提供：grant_type=password,
client_id, client_secret,redirect_uri
username, password
返回 Access Token [刷新码]
使用 Access Token 调用 API
返回 响应数据

2. 密码模式开发实践

开发流程



OAuth2 密码模式-总结

重难点

1. 密码模式的工作流程
2. 密码模式的开发流程

OAuth2 密码模式-总结

重难点

1. 密码模式的工作流程
2. 密码模式的开发流程

下节

客户端模式，流程非常**直接**

目录 Contents

OAuth2 客户端模式

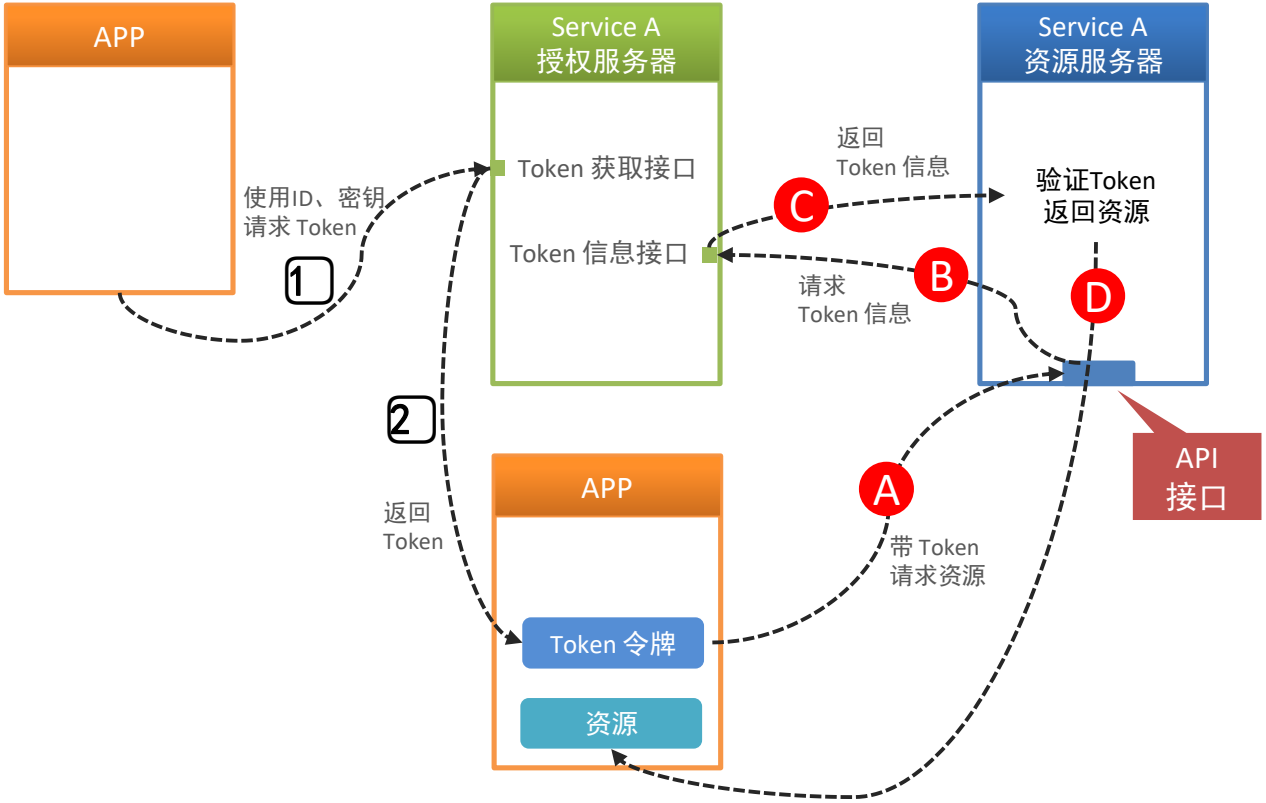
博学谷
www.boxuegu.com

小节导学

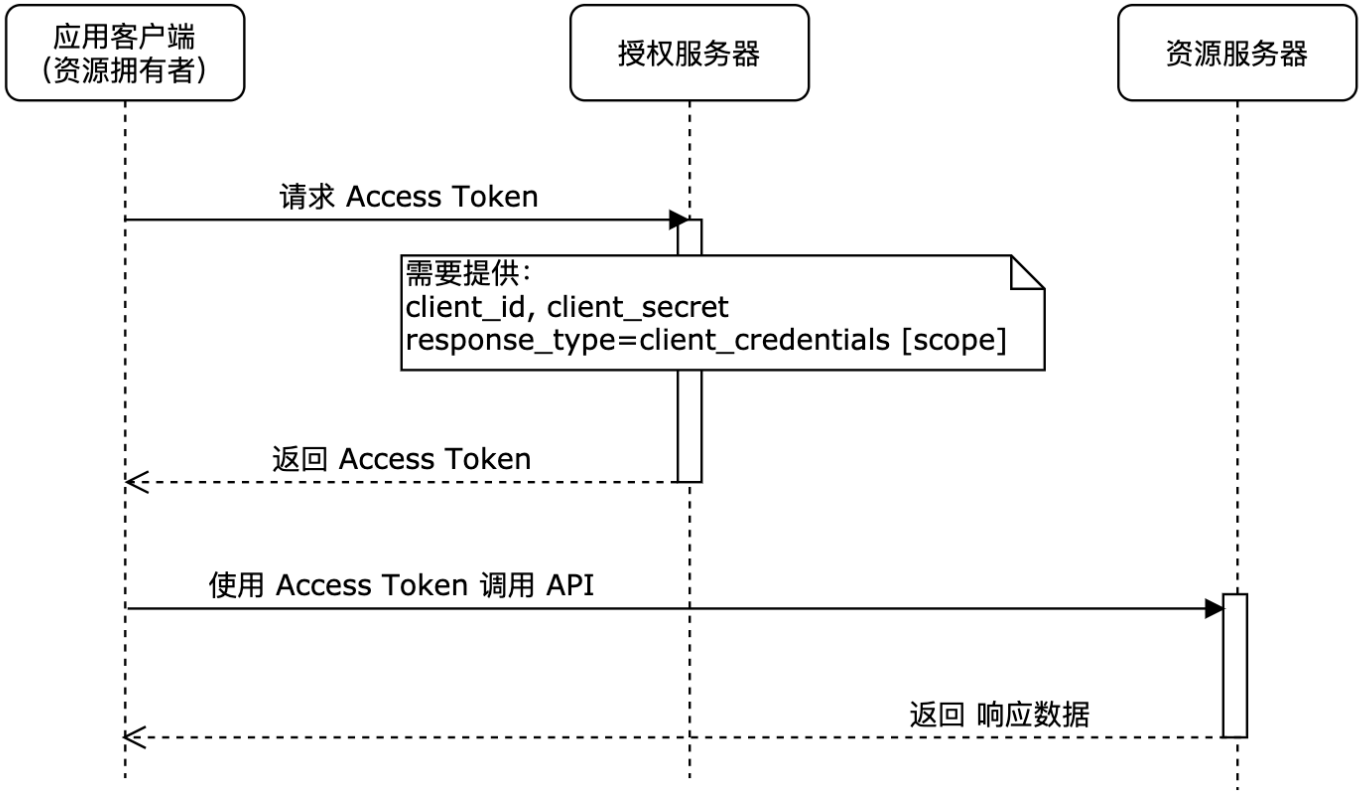
本节的目标就是弄明白“客户端模式”的工作流程，以及开发方法。

- 客户端模式的工作流程
- 客户端模式开发实践

1. 客户端模式的工作流程

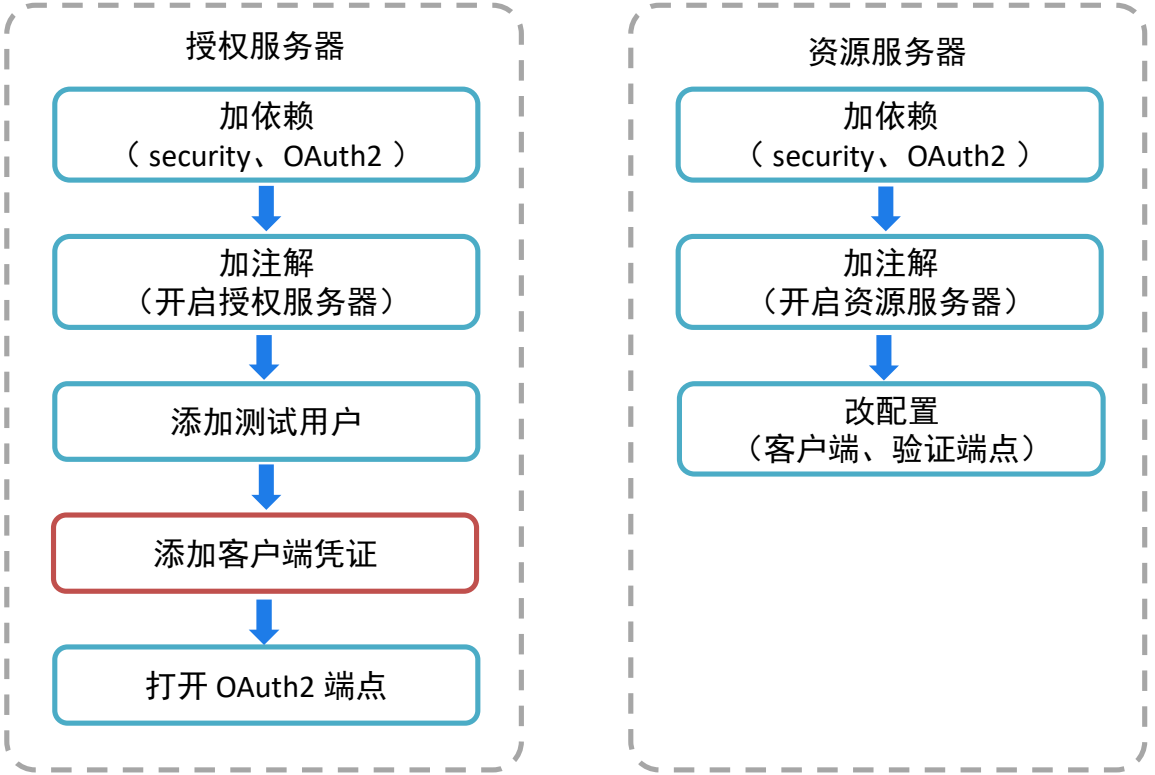


1. 客户端模式的工作流程



2. 客户端模式开发实践

开发流程



OAuth2 客户端模式-总结

重难点

1. 客户端模式的工作流程
2. 客户端模式的开发流程

OAuth2 客户端模式-总结

重难点

1. 客户端模式的工作流程
2. 客户端模式的开发流程

下节

实际开发中，这4种模式**应该怎么选?**

博学谷
www.boxuegu.com

目录 Contents

OAuth2 模式的选择策略

小节导学

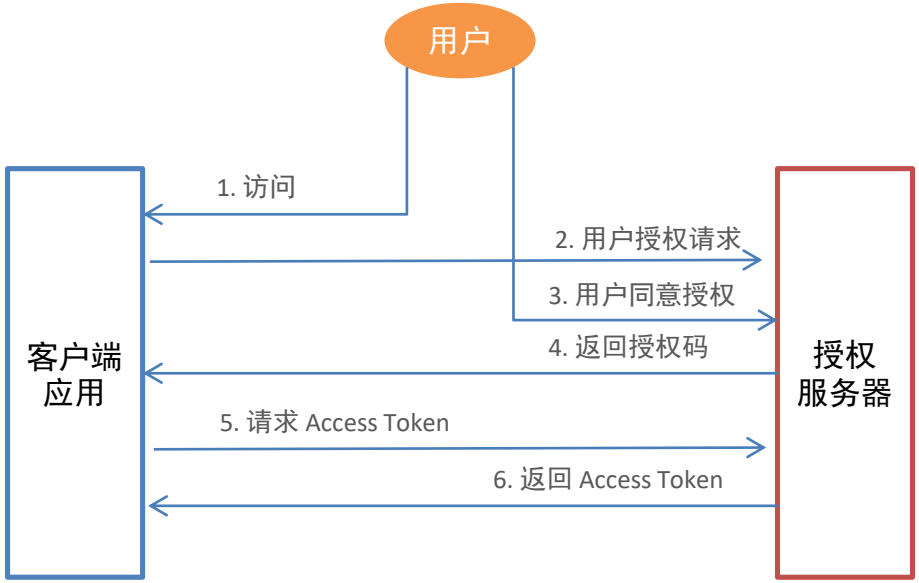
已经学习了 OAuth2 的4种经典模式，但在实际场景中具体如何选择呢？

本节就分析一下每种模式的使用场景，最后会总结出一个选择策略流程图。

- 授权码模式的使用场景
- 密码模式的使用场景
- 简化模式的使用场景
- 客户端模式的使用场景
- 模式选择策略总结

1. 授权码模式的使用场景

流程回顾

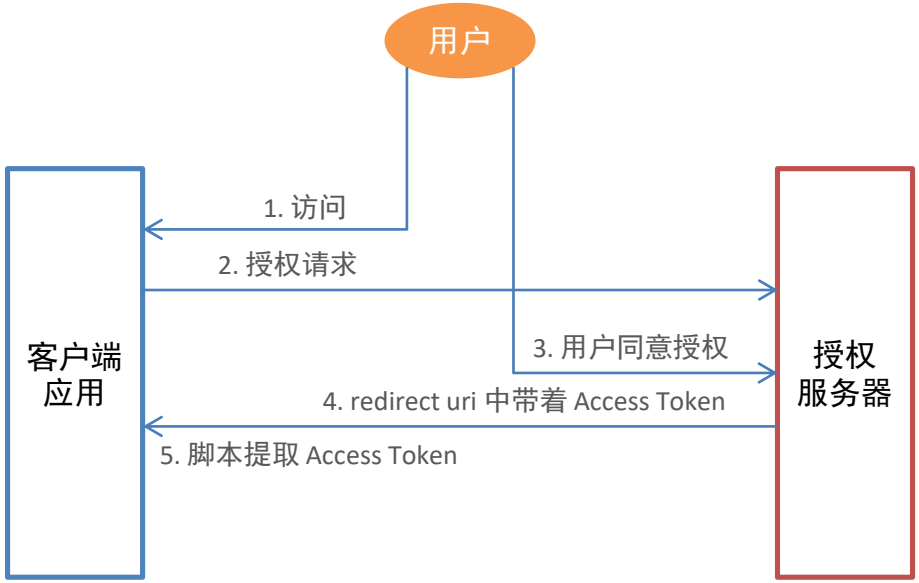


使用场景

- 客户端应有后端沟通能力，可以和授权服务器传递 token
- 对安全级别要求最高

2. 简化模式的使用场景

流程回顾



使用场景

- 应用客户端没有后端沟通能力，需要通过浏览器传递 token
- 对客户端权限可控

3. 密码模式的使用场景

流程回顾

使用场景

对客户端应用极其信任，自己人

4. 客户端模式的使用场景

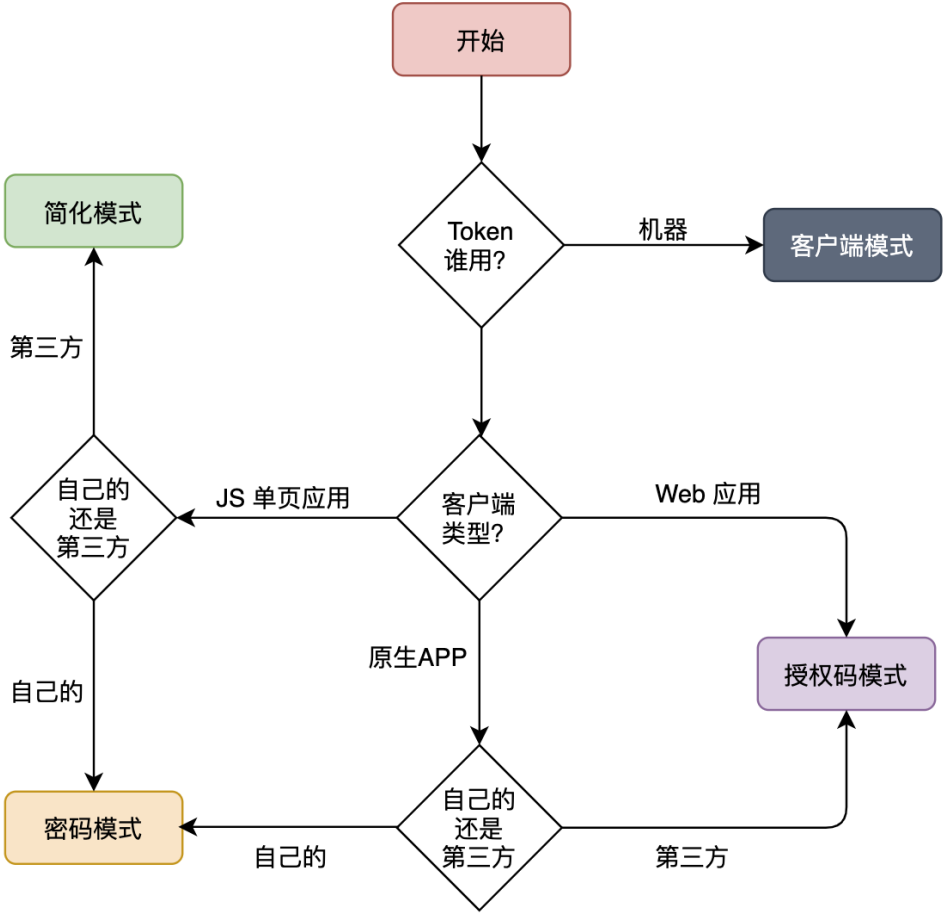
流程回顾

使用场景

没有用户的参与，不需要用户登录，只是

机器间的交互

5. 模式选择策略总结


开始
Token
谁用?
机器
客户端模式
简化模式
第三方
自己的
还是
第三方
JS 单页应用
客户端
类型?
Web 应用
自己的
原生APP
授权码模式
密码模式
自己的
自己的
还是
第三方
自己的
第三方

OAuth2 模式的选择策略-总结

重难点

1. 授权码模式的工作流程与使用场景
2. 简化模式的工作流程与使用场景
3. 密码模式的工作流程与使用场景
4. 客户端模式的工作流程与使用场景
5. 模式决策流程

OAuth2 模式的选择策略-总结

重难点

1. 授权码模式的工作流程与使用场景
2. 简化模式的工作流程与使用场景
3. 密码模式的工作流程与使用场景
4. 客户端模式的工作流程与使用场景
5. 模式决策流程

下节

JWT 是什么？怎么和 OAuth2 整合？

目录 Contents

OAuth2 整合 JWT

小节导学

在系统安全认证领域，JWT 是很有名的，那么 JWT 到底是什么？和 OAuth2 又是什么关系呢？

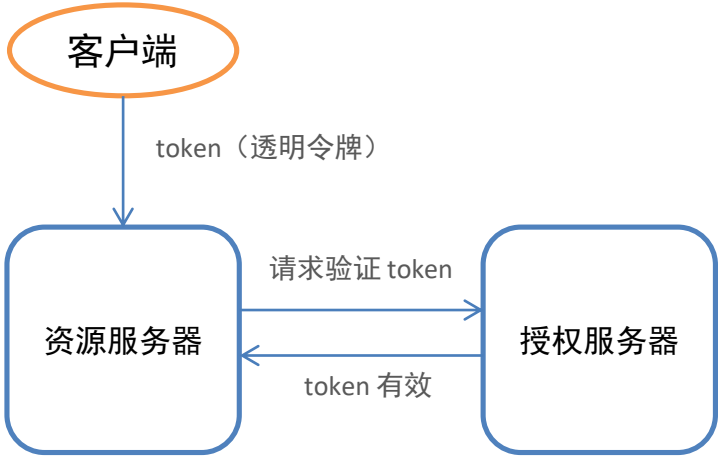
本节我们就认识一下 JWT，并实践一下 OAuth2 如何使用 JWT。

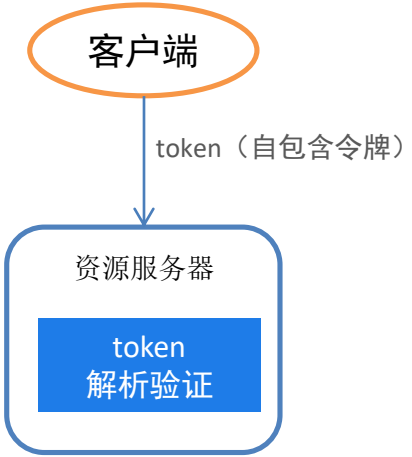
- JWT 简介
- OAuth2 整合 JWT 实践

1. JWT 简介

访问令牌的类型

- **透明令牌** - 随机生成的字符串，不包含任何信息，资源服务器需要请求授权服务器验证令牌有效性
- **自包含令牌** - 包含一些基本信息，例如谁颁发、颁发给谁、作用域，资源服务器可以本地校验令牌的有效性

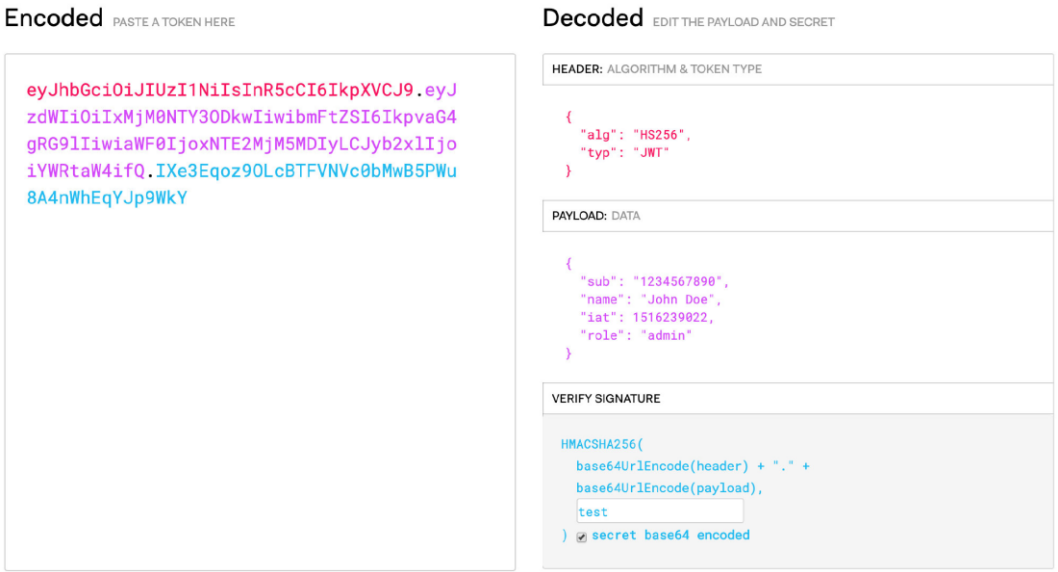




1. JWT 简介

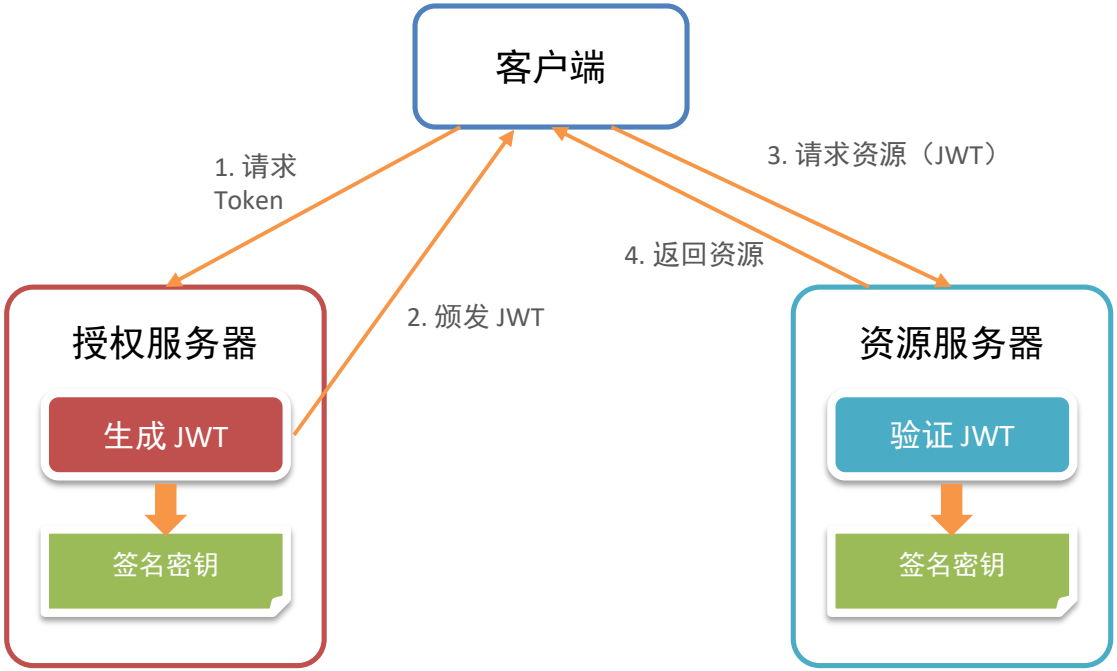
JWT 结构

JWT 的全称 **Json Web Token**，属于自包含令牌。其内部的数据使用 json 格式，由3部分组成。



2. OAuth2 整合 JWT 实践

结构



2. OAuth2 整合 JWT 实践

开发流程

OAuth2 整合 JWT -总结

重难点

1. JWT 的结构
2. 使用 JWT 后的验证流程
3. OAuth2 与 JWT 的整合方法

OAuth2 整合 JWT -总结

重难点

1. JWT 的结构
2. 使用 JWT 后的验证流程
3. OAuth2 与 JWT 的整合方法

下节

如何使用**数据库存储**客户端凭证 和 Token?

博学谷
www.boxuegu.com

目录 Contents

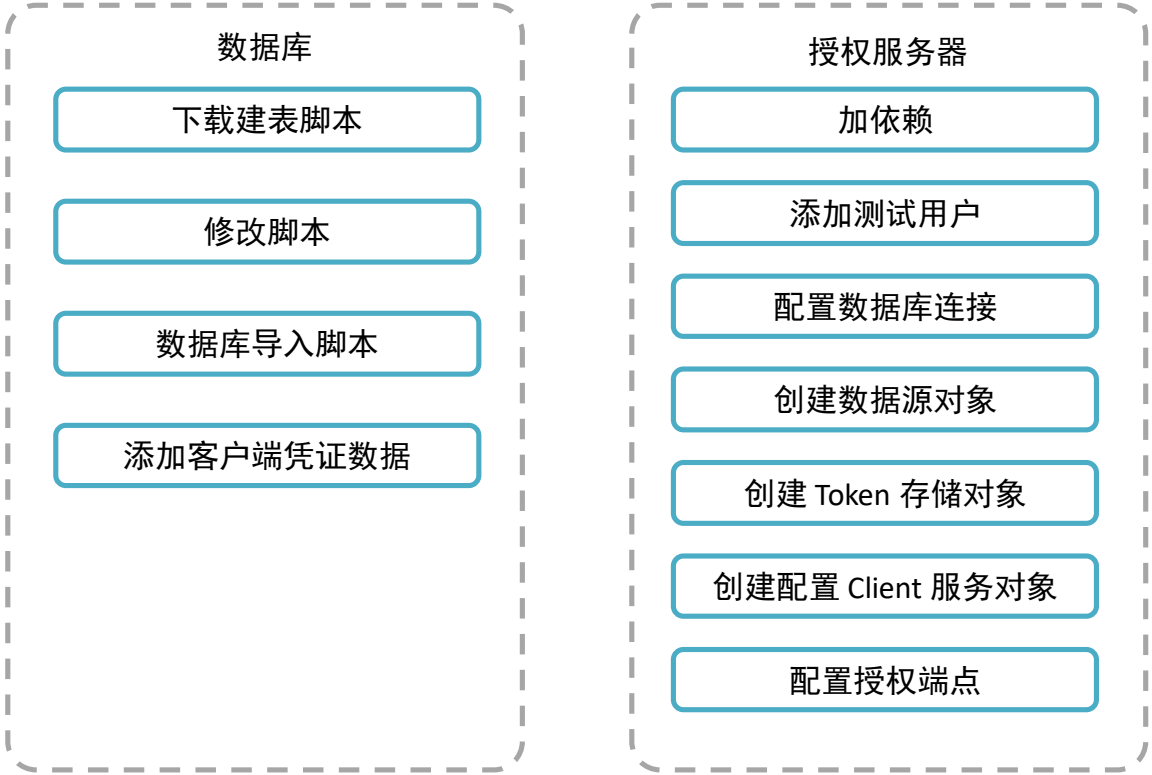
OAuth2 数据库存储令牌

小节导学

之前我们实践时，令牌是存放在内存中的，这只适用于开发测试，上线环境中，令牌需要存放在数据库中。
同样的，客户端凭证信息也要放在数据库中。
本节我们的目标就是进行数据库改造。

OAuth2 数据库存储令牌

实践流程



OAuth2 数据库存储令牌

数据库操作

- oauth2 表结构

项目地址：https://github.com/spring-projects/spring-security-oauth

脚本位置： spring-security-oauth2/src/test/resources/schema.sql

- 改脚本

“**LONGVARBINARY**”改为“**BLOB**”

- 导入脚本
- 添加客户端凭证数据

OAuth2 数据库存储令牌 -总结

重难点

1. OAuth2 数据库建表，各个表的结构

2. 数据库存储令牌的开发流程

OAuth2 数据库存储令牌 -总结

重难点

1. OAuth2 数据库建表，各个表的结构

2. 数据库存储令牌的开发流程

下节

什么是**基于角色的访问控制**？如何**整合 OAuth2**？

目录 Contents

基于角色的控制访问

小节导学

之前我们都是在代码中创建用户，例如这段代码：

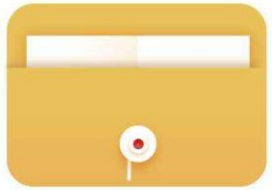
```
manager.createUser(User.withUsername("admin")
        .password(PasswordEncoderFactories.createDelegatingPasswordEncoder().encode("admin"))
        .authorities("USER").build());
```

虽然已经设置了角色“USER”，但并没有使用它。

这节我们就学习基于角色的权限控制（RBAC），把角色用起来。

- 认证授权的概念
- 基于角色的权限控制 RBAC
- OAuth2 整合角色实践

1. 认证授权的概念

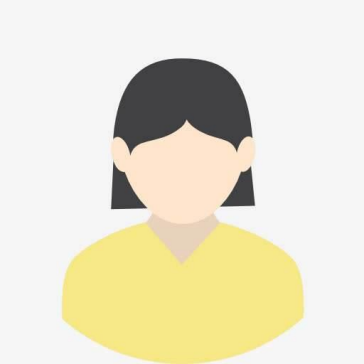


认证 - 验证身份，决定是否允许进入

授权 - 虽然你进来了，但有些东西你能看，有些你不能看

1. 认证授权的概念

授权的维度



人

系统

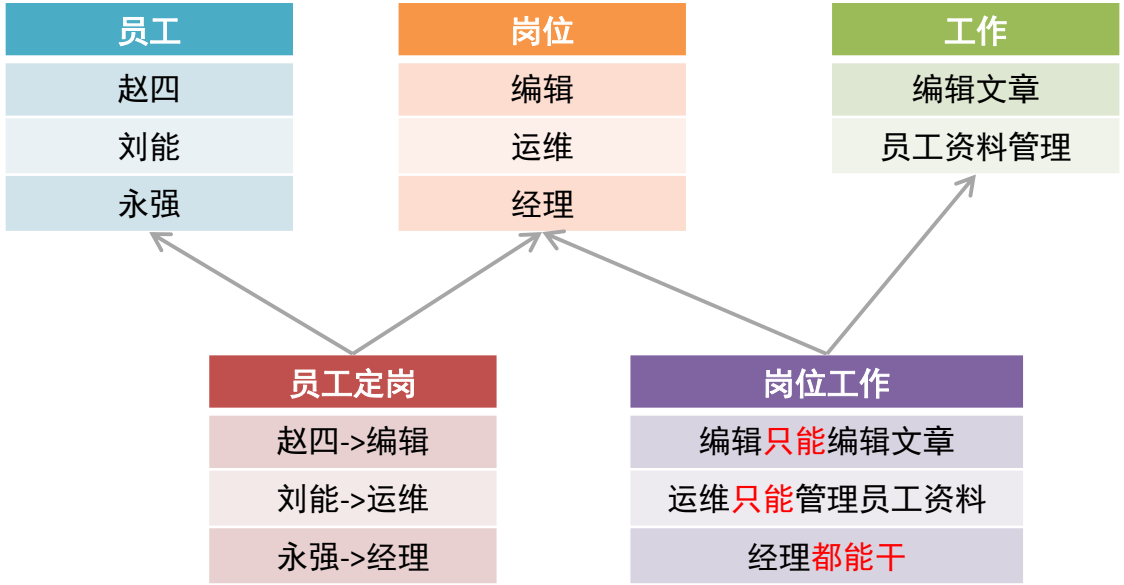
时间

2. 基于角色的权限控制 RBAC

基于角色的访问控制（Role-Based Access Control ）权限与角色相关联，为用户赋予某个角色，角色具有某些权限，从而使用户具有某些权限。

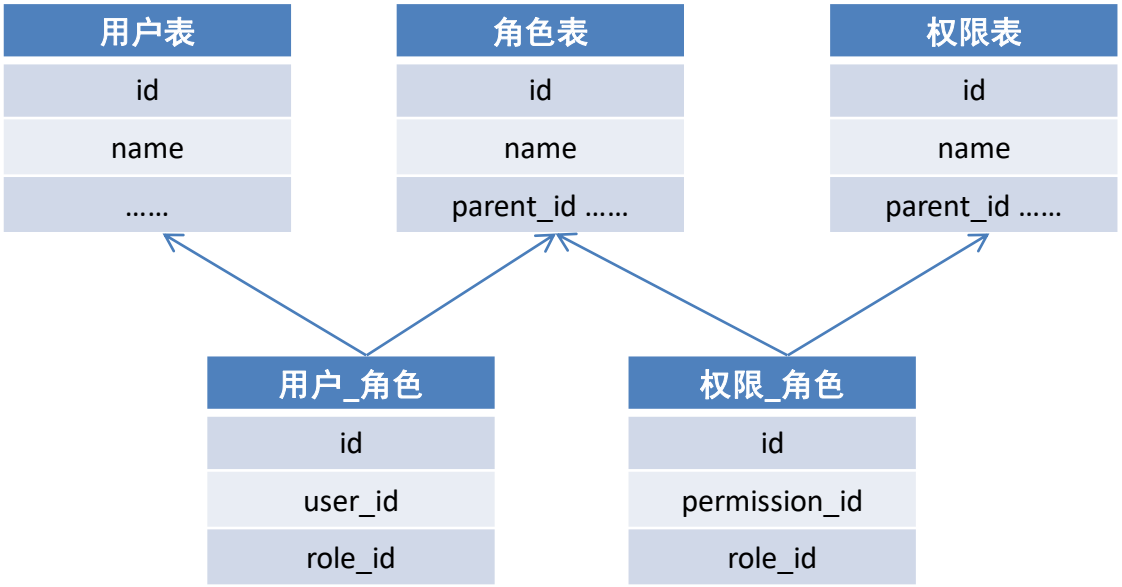
2. 基于角色的权限控制 RBAC

关系图



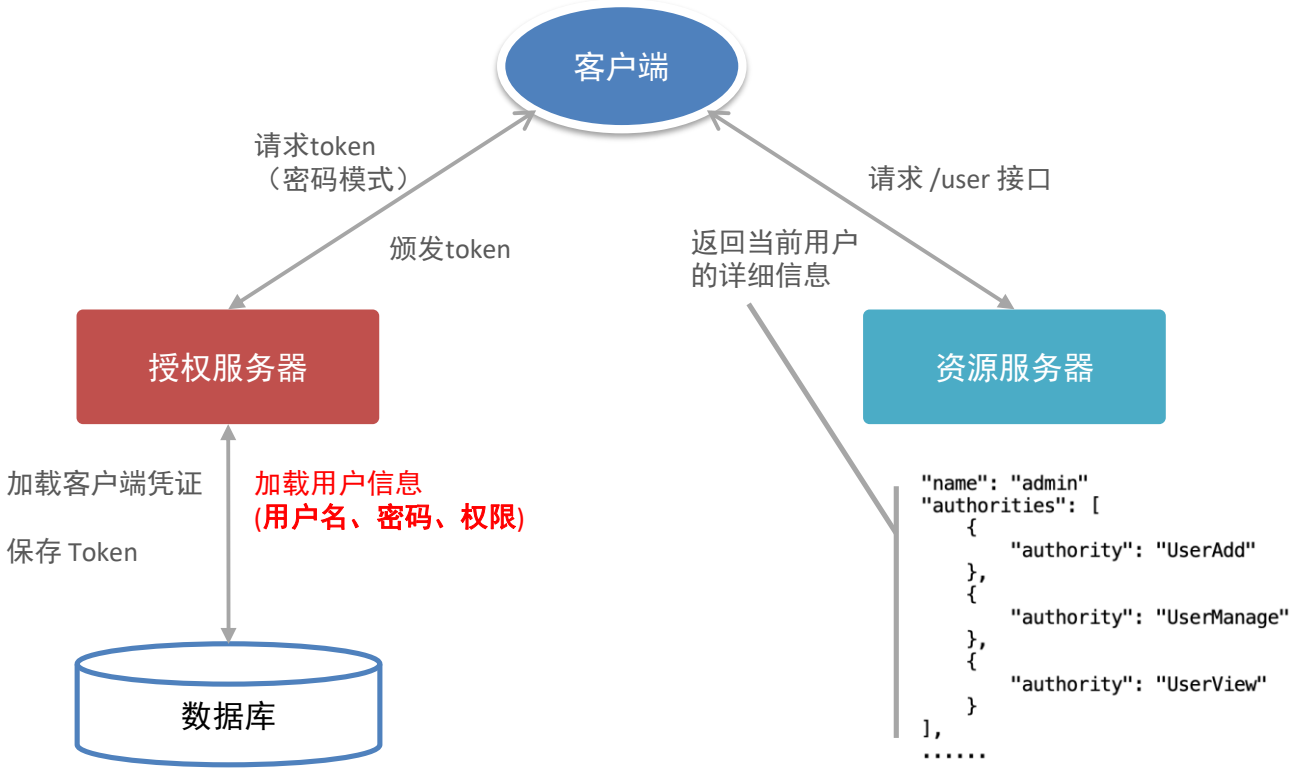
2. 基于角色的权限控制 RBAC

数据模型



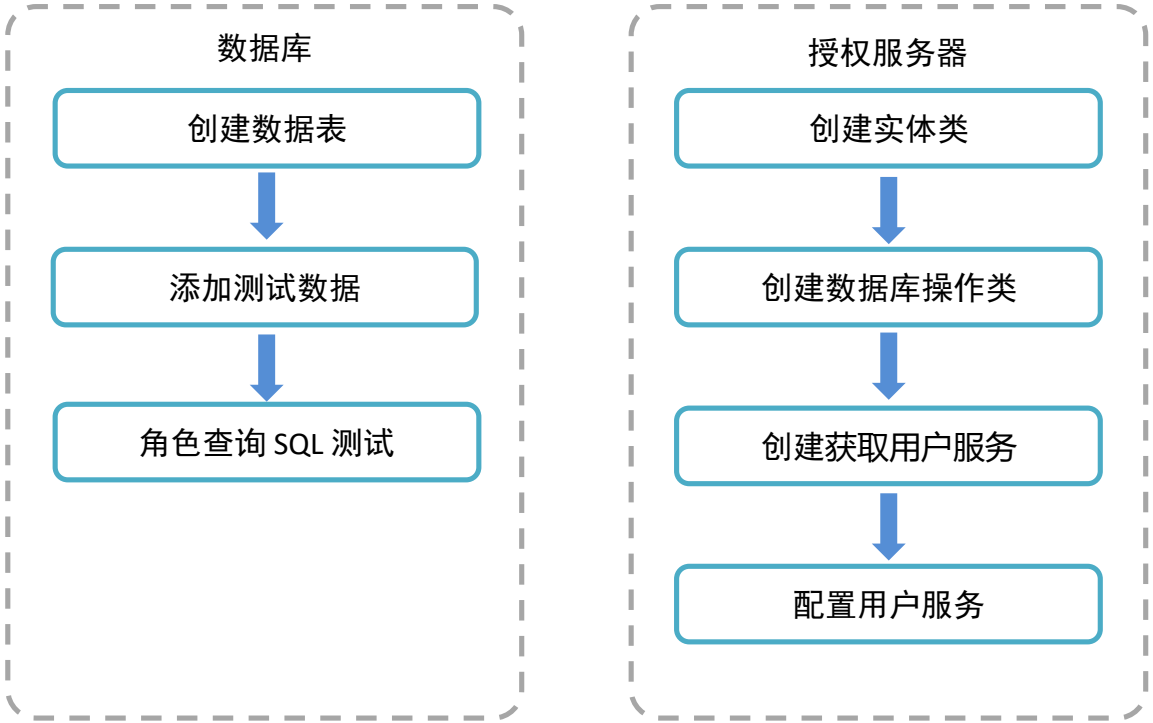
3. OAuth2 整合角色实践

实践目标



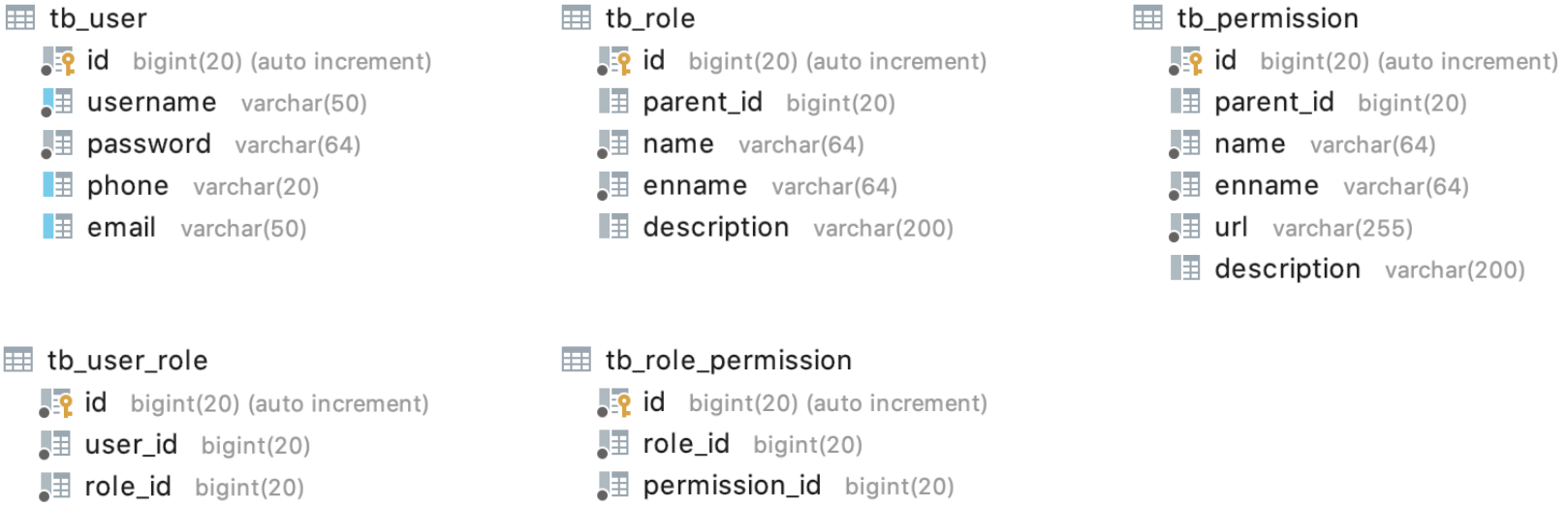
3. OAuth2 整合角色实践

实践流程



3. OAuth2 整合角色实践

表结构



基于角色的访问控制 -总结

重难点

1. 认证、授权的概念
2. RBAC 的设计思路
3. OAuth2 认证流程整合用户角色

一样的在线教育，不一样的教学品质