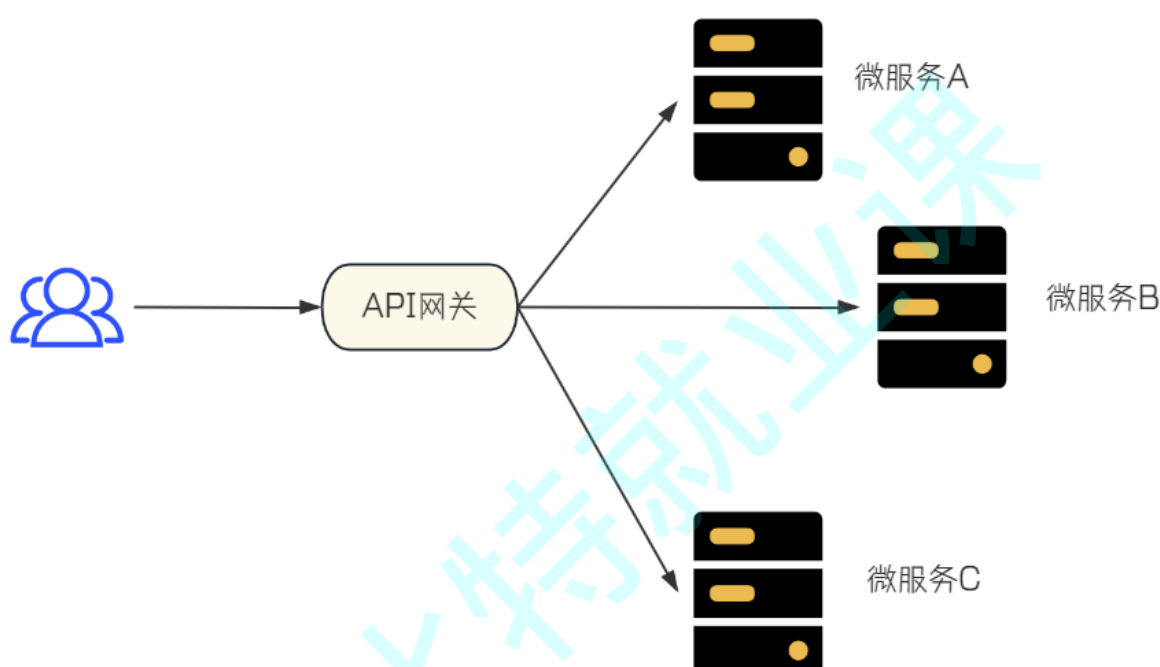


# 在线oj项目-API网关

## API网关

### 简介

API网关(简称网关)也是一个服务,通常是后端服务的唯一入口. 它的定义类似设计模式中的Facade模式(门面模式,也称外观模式). 它就类似整个微服务架构的门面,所有的外部客户端访问,都需要经过它来进行调度和过滤.



### 网关的核心作用:

- **权限控制:** 作为微服务的入口, 对用户进行权限校验, 如果校验失败则进行拦截
- **动态路由:** 一切请求先经过网关, 但网关不处理业务, 而是根据某种规则, 把请求转发到某个微服务
- **负载均衡:** 当路由的目标服务有多个时, 还需要做负载均衡
- **限流:** 请求流量过高时, 按照网关中配置微服务能够接受的流量进行放行, 避免服务压力过大.

我们将使用springcloud gateway作为我们项目的api网关。

## 引入网关

- 引入网关依赖

```
1 <!-- SpringCloud Gateway -->
2 <dependency>
3     <groupId>org.springframework.cloud</groupId>
4     <artifactId>spring-cloud-starter-gateway</artifactId>
5 </dependency>
6
7 <!-- SpringCloud Loadbalancer -->
8 <dependency>
9     <groupId>org.springframework.cloud</groupId>
10    <artifactId>spring-cloud-starter-loadbalancer</artifactId>
11 </dependency>
12
13 <!-- SpringCloud Alibaba Nacos -->
14 <dependency>
15     <groupId>com.alibaba.cloud</groupId>
16     <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
17 </dependency>
18
19 <!-- SpringCloud Alibaba Nacos Config -->
20 <dependency>
21     <groupId>com.alibaba.cloud</groupId>
22     <artifactId>spring-cloud-starter-alibaba-nacos-config</artifactId>
23 </dependency>
```

- bootstrap.yml

```
1 # Spring
2 spring:
3     application:
4         # 应用名称
5         name: oj-gateway
6     profiles:
7         # 环境配置
8         active: local
9     cloud:
10         nacos:
11             discovery:
12                 namespace: biteoj
13                 server-addr: http://localhost:8848
14             config:
15                 namespace: biteoj
16                 server-addr: http://localhost:8848
17                 file-extension: yaml
```

- **nacos配置**

```
1 server:
2   port: 19090
3 spring:
4   application:
5     # 应用名称
6     name: oj-gateway
7   cloud:
8     gateway:
9       routes:
10        # 管理模块
11        - id: oj-system
12          uri: lb://oj-system
13          predicates:
14            - Path=/system/**
15          filters:
16            - StripPrefix=1
```

#### 配置字段说明:

- id: 自定义路由ID, 保持唯一
- uri: 目标服务地址, 支持普通URI 及 `lb://应用注册服务名称`. lb表示负载均衡, 使用 `lb://` 方式表示从注册中心获取服务地址.
- predicates: 路由条件, 根据匹配结果决定是否执行该请求路由, 上述代码中, 我们把符合Path规则的一切请求, 都代理到uri参数指定的地址.
- filters: 于定义在请求转发到目标地址之前或之后执行的过滤器。
  - StripPrefix: 这是一个StripPrefix过滤器, 它的作用是从请求的路径中去除一部分前缀。  
`StripPrefix=1` 表示去除1个路径段。例如, 如果原始请求的路径是 `/system/test`, 那么经过这个过滤器处理后, 转发到目标服务的路径就变成了 `/test`。