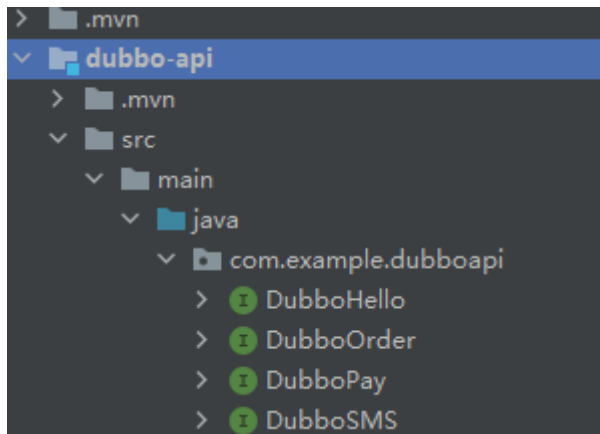


Springcloud Gateway 学习

网关直接转发后端服务，可以在前面加Nginx

一：hello world

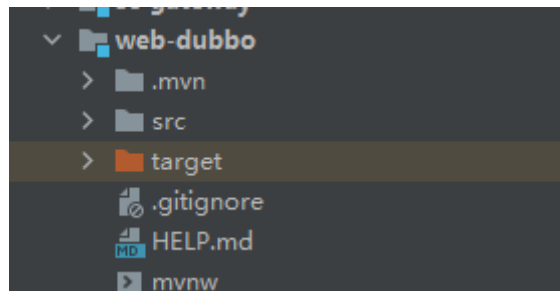
1.1:创建接口



创建DubboHello接口

```
public interface DubboHello {  
    String sayHello(String name);  
}
```

1.2:创建dubbo 服务提供者



1: pom:

```
<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
      http://maven.apache.org/xsd/maven-4.0.0.xsd">  
    <modelVersion>4.0.0</modelVersion>  
    <groupId>com.example</groupId>  
    <artifactId>web-dubbo</artifactId>  
    <version>0.0.1-SNAPSHOT</version>  
    <name>web-dubbo</name>  
    <description>web-dubbo</description>  
  
    <properties>  
        <java.version>1.8</java.version>
```

```

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
<spring-boot.version>2.3.7.RELEASE</spring-boot.version>
<spring-cloud-alibaba.version>2.2.2.RELEASE</spring-cloud-alibaba.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>com.alibaba.cloud</groupId>
    <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
  </dependency>
  <dependency>
    <groupId>com.alibaba.cloud</groupId>
    <artifactId>spring-cloud-starter-dubbo</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
    <exclusions>
      <exclusion>
        <groupId>org.junit.vintage</groupId>
        <artifactId>junit-vintage-engine</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
  <dependency>
    <groupId>com.example</groupId>
    <artifactId>dubbo-api</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <scope>compile</scope>
  </dependency>
  <dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger2</artifactId>
    <version>2.6.1</version>
  </dependency>

  <dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger-ui</artifactId>
    <version>2.6.1</version>
  </dependency>
</dependencies>
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-dependencies</artifactId>
      <version>${spring-boot.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>

```

```

        <dependency>
            <groupId>com.alibaba.cloud</groupId>
            <artifactId>spring-cloud-alibaba-dependencies</artifactId>
            <version>${spring-cloud-alibaba.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>

<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.1</version>
            <configuration>
                <source>1.8</source>
                <target>1.8</target>
                <encoding>UTF-8</encoding>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <version>2.3.7.RELEASE</version>
            <configuration>
                <mainClass>com.example.webdubbo.WebDubboApplication</mainClass>
            </configuration>
            <executions>
                <execution>
                    <id>repackage</id>
                    <goals>
                        <goal>repackage</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>

</project>

```

2: 配置文件:

```

dubbo:
  scan:
    base-packages: com.example.webdubbo
  protocols:
    dubbo:
      name: dubbo
      port: -1
  consumer:
    timeout: 5000
    check: false
    retries: 0
  cloud:

```

```
subscribed-services:
```

```
server:
  port: 8081

spring:
  application:
    name: web-dubbo
  main:
    allow-bean-definition-overriding: true
  cloud:
    nacos:
      server-addr: 127.0.0.1:8848
      username: nacos
      password: nacos
```

3: service

```
@DubboService(protocol = "dubbo")
public class DubboHelloImpl implements DubboHello {
    @Override
    public String sayHello(String name) {
        System.out.println("服务提供者被调用");
        return "dubbo service: " + name;
    }
}
```

4: controller

```
@RestController
public class WebController {
    @Autowired
    private DubboHello hello;

    @GetMapping("/test/{name}")
    public String test(@PathVariable("name") String name) {
        return hello.sayHello(name);
    }
}
```

5: application

```
@EnableDiscoveryClient
@SpringBootApplication
public class WebDubboApplication {

    public static void main(String[] args) {
        SpringApplication.run(WebDubboApplication.class, args);
    }

}
```

1.3: gateway 配置

1: pom

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>sc-gateway</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>sc-gateway</name>
  <description>sc-gateway</description>

  <properties>
    <java.version>1.8</java.version>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <spring-boot.version>2.3.7.RELEASE</spring-boot.version>
    <spring-cloud-alibaba.version>2.2.2.RELEASE</spring-cloud-alibaba.version>
    <spring-cloud.version>Hoxton.SR9</spring-cloud.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>com.alibaba.cloud</groupId>
      <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-gateway</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
      <exclusions>
        <exclusion>
          <groupId>org.junit.vintage</groupId>
          <artifactId>junit-vintage-engine</artifactId>
        </exclusion>
      </exclusions>
    </dependency>
  </dependencies>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-dependencies</artifactId>
        <version>${spring-cloud.version}</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>
</project>
```

```

        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-dependencies</artifactId>
        <version>${spring-boot.version}</version>
        <type>pom</type>
        <scope>import</scope>
    </dependency>
</dependencies>
</dependencyManagement>

<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.1</version>
            <configuration>
                <source>1.8</source>
                <target>1.8</target>
                <encoding>UTF-8</encoding>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <version>2.3.7.RELEASE</version>
            <configuration>
                <mainClass>com.example.scgateway.ScGatewayApplication</mainClass>
            </configuration>
            <executions>
                <execution>
                    <id>repackage</id>
                    <goals>
                        <goal>repackage</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>

</project>

```

2: 配置文件

application.yml

```

spring:
  cloud:
    gateway:
      discovery:

```

```

locator:
  lowerCaseServiceId: true
  enabled: true
# 访问链接http://127.0.0.1:9900/api-web/test/abc
routes:
- id: web
  uri: lb://web-dubbo
  predicates:
    - Path=/api-web/**
  filters:
    - StripPrefix=1

```

bootstrap.yml

```

server:
  port: 9900
spring:
  application:
    name: sc-gateway
  main:
    allow-bean-definition-overriding: true
  cloud:
    nacos:
      server-addr: 127.0.0.1:8848
      username: nacos
      password: nacos
      discovery:
        namespace: public

```

3: application

```

@EnableDiscoveryClient
@SpringBootApplication
public class ScGatewayApplication {

    public static void main(String[] args) {
        SpringApplication.run(ScGatewayApplication.class, args);
    }

}

```

启动访问: <http://localhost:9900/api-web/test/lktbz>

二：配置方式学习：

1: yaml 配置方式

```

spring:
  cloud:
    gateway:
      discovery:
        locator:
          lowerCaseServiceId: true
          enabled: true

```

```

# 访问链接http://localhost:8088/gateway/pay
routes:
  - id : web
#     uri: lb://web-dubbo
  predicates:
    - Path= /gateway/**      #在映射路径前追加路径 在pay 前追加gateway
  filters:
    - StripPrefix=1 #过滤匹配中的 /gateway/ 直接匹配后面的路径(匹配pay路径
uri: http://localhost:8083    #转发地址

```

```

spring:
  cloud:
    gateway:
      discovery:
        locator:
          lowerCaseServiceId: true
          enabled: true
      routes:
        - id : web_route
#          uri: lb://web-dubbo
        predicates:
          - Path= /gateway/**      #在映射路径前追加路径
        filters:
          - StripPrefix=1    #过滤匹配中的 /gateway/ 直接匹配后面的路径
uri: http://localhost:8083    #转发地址
        - id: cookie_route #在创建一个路由规则
        predicates: #对cookie 进行断言
          - Cookie=name,lktbz    #cookie 中的参数带有name
uri: http://www.baidu.com    #cookie 条件满足，则跳转到百度

```

其余的配置需要参考官网，给出的指南非常详细

<https://docs.spring.io/spring-cloud-gateway/docs/current/reference/html/#marking-an-exchange-as-routed>