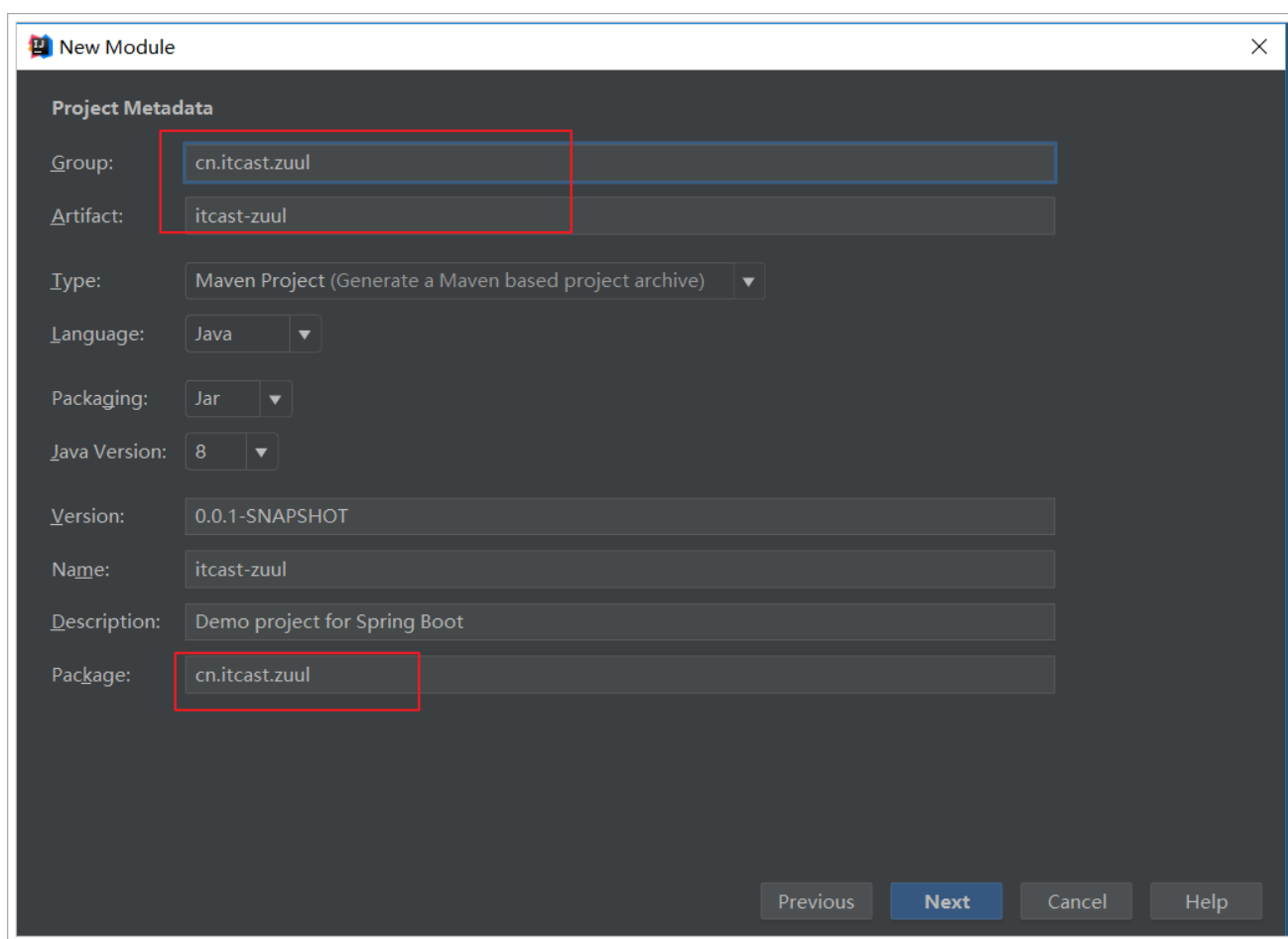


## 3.3.快速入门

### 3.3.1.新建工程

填写基本信息：

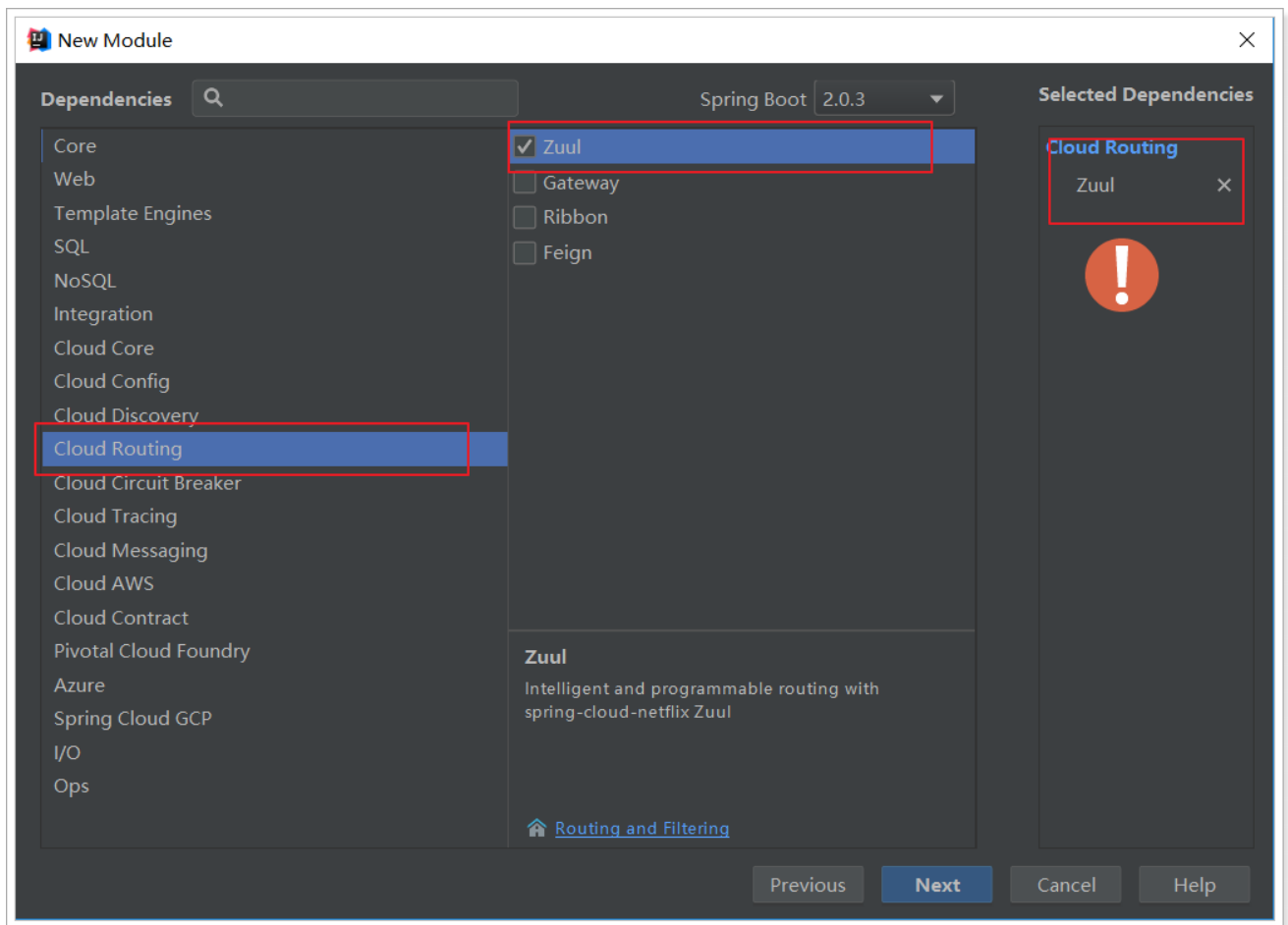


The screenshot shows the 'New Module' dialog box with the following fields and values:

Field	Value
Group:	cn.itcast.zuul
Artifact:	itcast-zuul
Type:	Maven Project (Generate a Maven based project archive)
Language:	Java
Packaging:	Jar
Java Version:	8
Version:	0.0.1-SNAPSHOT
Name:	itcast-zuul
Description:	Demo project for Spring Boot
Package:	cn.itcast.zuul

At the bottom of the dialog, there are four buttons: 'Previous', 'Next' (highlighted in blue), 'Cancel', and 'Help'.

添加Zuul依赖：



### 3.3.2.编写配置

```
server:
  port: 10010 #服务端口
spring:
  application:
    name: api-gateway #指定服务名
```

### 3.3.3.编写引导类

通过 `@EnableZuulProxy` 注解开启Zuul的功能:

```
@SpringBootApplication
@EnableZuulProxy // 开启网关功能
public class ItcastZuulApplication {

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

### 3.3.4.编写路由规则

我们需要用Zuul来代理service-provider服务，先看一下控制面板中的服务状态：

Application	AMIs	Availability Zones	Status
ITCAST-EUREKA	n/a (1)	(1)	UP (1) - localhost:itcast-eureka:10086
ITCAST-ZUUL	n/a (1)	(1)	UP (1) - localhost:itcast-zuul:10010
SERVICE-CONSUMER	n/a (1)	(1)	UP (1) - localhost:service-consumer:80
SERVICE-PROVIDER	n/a (1)	(1)	UP (1) - localhost:service-provider:8081

- ip为：127.0.0.1
- 端口为：8081

映射规则：

```
server:
  port: 10010 #服务端口
spring:
  application:
    name: api-gateway #指定服务名
zuul:
  routes:
    service-provider: # 这里是路由id，随意写
      path: /service-provider/** # 这里是映射路径
      url: http://127.0.0.1:8081 # 映射路径对应的实际url地址
```

我们将符合 `path` 规则的一切请求，都代理到 `url` 参数指定的地址

本例中，我们将 `/service-provider/**` 开头的请求，代理到<http://127.0.0.1:8081>

### 3.3.5.启动测试

访问的路径中需要加上配置规则的映射路径，我们访问：<http://127.0.0.1:10010/service-provider/user/1>

