



《Spring Cloud Netflix》

第 06 节

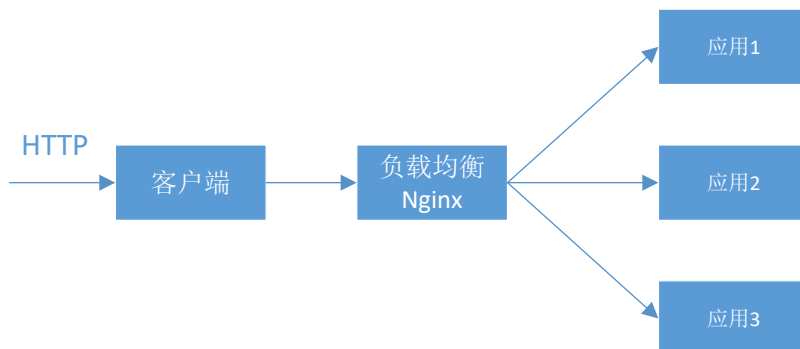
负载均衡-Ribbon 的使用

一、什么是负载均衡

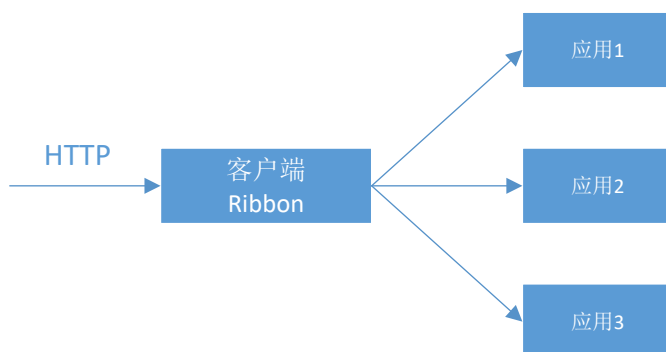
负载均衡就是分发请求流量到不同的服务器。

负载均衡一般分为两种：

1、服务器端负载均衡（nginx）



2、客户端负载均衡（Ribbon）



二、spring-cloud-provider（服务提供者）：

实体类：

```
package com.roncoo.education.bean;
```

```
import java.util.Date;
```

```
/**
```

```
 * 实体类
```

```
 *
```

```
 * @author wujing
```

```
 */
```

```
public class User {
```

```
    private int id;
```

```
    private String name;
```



```
private Date createTime;

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public Date getCreateTime() {
    return createTime;
}

public void setCreateTime(Date createTime) {
    this.createTime = createTime;
}

@Override
public String toString() {
    return "RoncooUser [id=" + id + ", name=" + name + ",\ncreateTime=" + createTime + "]\n";
}

}
```

接口类:

```
package com.roncoo.education.controller;
import java.util.Date;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
```



```
import org.springframework.web.bind.annotation.RestController;

import com.roncoo.education.bean.User;

/**
 * @author wujing
 */
@RestController
@RequestMapping(value = "/api/user")
public class ApiUserController {

    protected final Logger logger =
        LoggerFactory.getLogger(this.getClass());

    @RequestMapping(value =("/{id}", method =
        RequestMethod.GET)
    public User view(@PathVariable int id) {
        User user = new User();
        user.setId(id);
        user.setName("无境");
        user.setCreateTime(new Date());
        logger.info("请求接口返回: {}", user);
        return user;
    }
}
```

三、spring-cloud-consumer(服务消费者):

```
package com.roncoo.education.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

/**
 * @author wujing
 */
@RestController
@RequestMapping(value = "/user", method = RequestMethod.POST)
public class UserController {
```



```
private static final String
URL="http://localhost:7777/api/user/{id}";

@Autowired
private RestTemplate restTemplate;

@RequestMapping(value =("/{id}", method =
RequestMethod.GET)
public String get(@PathVariable(value = "id") int id) {
    return restTemplate.getForObject(URL, String.class, id);
}

}
```

使用RestTemplate进行调用，所以要先定义这个Bean

```
@Bean
protected RestTemplate restTemplate() {
    return new RestTemplate();
}
```

四、查看

provide: <http://localhost:7777/api/user/1>
consumer: <http://localhost:8888/user/1>

五、如何通过 Ribbon 进行调用

- 1、在 RestTemplate 中添加注解 @LoadBalanced
 - 2、修改调用的 URL=<http://spring-cloud-provider/api/user/{id}>
- 注意：控制台的应用名字为大写，我们统一为小写，更不能大小写都存在。

六、如何实现负载均衡

我们把提供者的应用，启动多个，进行测试。
为了区分，我们修改实体的名字。
注意：修改应用的时候，端口也要修改。

七、结论

- 1、Ribbon 通过@LoadBalanced 进行负载均衡。
- 2、默认的负载策略是轮询算法。

疑问：

- 1、Ribbon 的实现原理？
- 2、Ribbon 还有什么负载策略？