RestTemplate

通过 RestTemplate 可以实现不同微服务之间的调用。

RestTemplate 是 Spring 框架提供的一种基于 RESTful 的服务组件，底层对 HTTP 请求及响应进行了封装，提供了很多访问远程 REST 服务的方法，可以简化代码的开发。

如何使用 RestTemplate

1、创建 Maven 工程，pom.xml

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.0.7.RELEASE</version>
</parent>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
  </dependency>
</dependencies>
```

2、创建 User 类

```
package com.southwind.entity;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class User {
    private Integer id;
    private String name;
}
```

3、创建 UserRepository 接口及实现类

```java
package com.southwind.repository;

import com.southwind.entity.User;

import java.util.Collection;

public interface UserRepository {
    public Collection<User> findAll();
    public User findById(Integer id);
    public void saveOrUpdate(User user);
    public void deleteById(Integer id);
}
```

```java
package com.southwind.repository.impl;

import com.southwind.entity.User;
import com.southwind.repository.UserRepository;
import org.springframework.stereotype.Repository;

import java.util.Collection;
import java.util.HashMap;
import java.util.Map;

@Repository
public class UserRepositoryImpl implements UserRepository {

    private static Map<Integer,User> map;

    static {
        map = new HashMap<>();
        map.put(1,new User(1,"张三"));
        map.put(2,new User(2,"李四"));
        map.put(3,new User(3,"王五"));
    }

    @Override
    public Collection<User> findAll() {
        return map.values();
    }

    @Override
    public User findById(Integer id) {
        return map.get(id);
    }

    @Override
    public void saveOrUpdate(User user) {
        map.put(user.getId(),user);
```

```
    }

    @Override
    public void deleteById(Integer id) {
        map.remove(id);
    }
}
```

4、创建 UserHandler

```
package com.southwind.controller;

import com.southwind.entity.User;
import com.southwind.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.Collection;

@RestController
@RequestMapping("/user")
public class UserHandler {
    @Autowired
    private UserRepository userRepository;

    @GetMapping("/findAll")
    public Collection<User> findAll(){
        return userRepository.findAll();
    }

    @GetMapping("/findById/{id}")
    public User findById(@PathVariable("id") Integer id){
        return userRepository.findById(id);
    }

    @PostMapping("/save")
    public void save(@RequestBody User user){
        userRepository.saveOrUpdate(user);
    }

    @PutMapping("/update")
    public void update(@RequestBody User user){
        userRepository.saveOrUpdate(user);
    }

    @DeleteMapping("/deleteById/{id}")
    public void deleteById(@PathVariable("id") Integer id){
        userRepository.deleteById(id);
    }
```

```
}
```

使用 RestTemplate 访问 REST 服务

1、将 RestTemplate 的实例化对象通过 @Bean 注入 IoC。

```
package com.southwind.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.client.RestTemplate;

@Configuration
public class MyConfig {

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

2、创建 RestHandler

```
package com.southwind.controller;

import com.southwind.entity.User;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.client.RestTemplate;

import java.util.Collection;

@RestController
@RequestMapping("/rest")
public class RestHandler {

    @Autowired
    private RestTemplate restTemplate;

    @GetMapping("/findAll")
    public Collection<User> findAll(){
        return
restTemplate.getForObject("http://localhost:8080/user/findAll",Collection.clas
s);
    }

    @GetMapping("/findById/{id}")
    public User findById(@PathVariable("id") Integer id){
```

```
        return
restTemplate.getForObject("http://localhost:8080/user/findById/{id}",User.clas
s,id);
    }

    @PostMapping("/save")
    public void save(@RequestBody User user){

 restTemplate.postForObject("http://localhost:8080/user/save",user,Collection.
class);
    }

    @PutMapping("/update")
    public void update(@RequestBody User user){
        restTemplate.put("http://localhost:8080/user/update",user);
    }

    @DeleteMapping("/deleteById/{id}")
    public void deleteById(@PathVariable("id") Integer id){
        restTemplate.delete("http://localhost:8080/user/deleteById/{id}",id);
    }
}
```

RestTemplate 底层对 HTTP 请求及响应进行了封装，提供了很多访问远程 REST 服务的方法，基于它的这个特性，我们可以实现不同微服务之间的调用。

服务消费者

1、创建 Module，pom.xml

```
<dependencies>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    </dependency>
</dependencies>
```

2、application.yml

```yaml
server:
  port: 8020
spring:
  application:
    name: consumer
eureka:
  client:
    service-url:
      defaultZone: http://localhost:8761/eureka/
  instance:
    prefer-ip-address: true
```

3、启动类

```java
package com.southwind;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

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

4、实体类

```java
package com.southwind.entity;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class Student {
    private Integer id;
    private String name;
}
```

5、注入 RestTemplate

```java
package com.southwind;

import org.springframework.boot.SpringApplication;
```

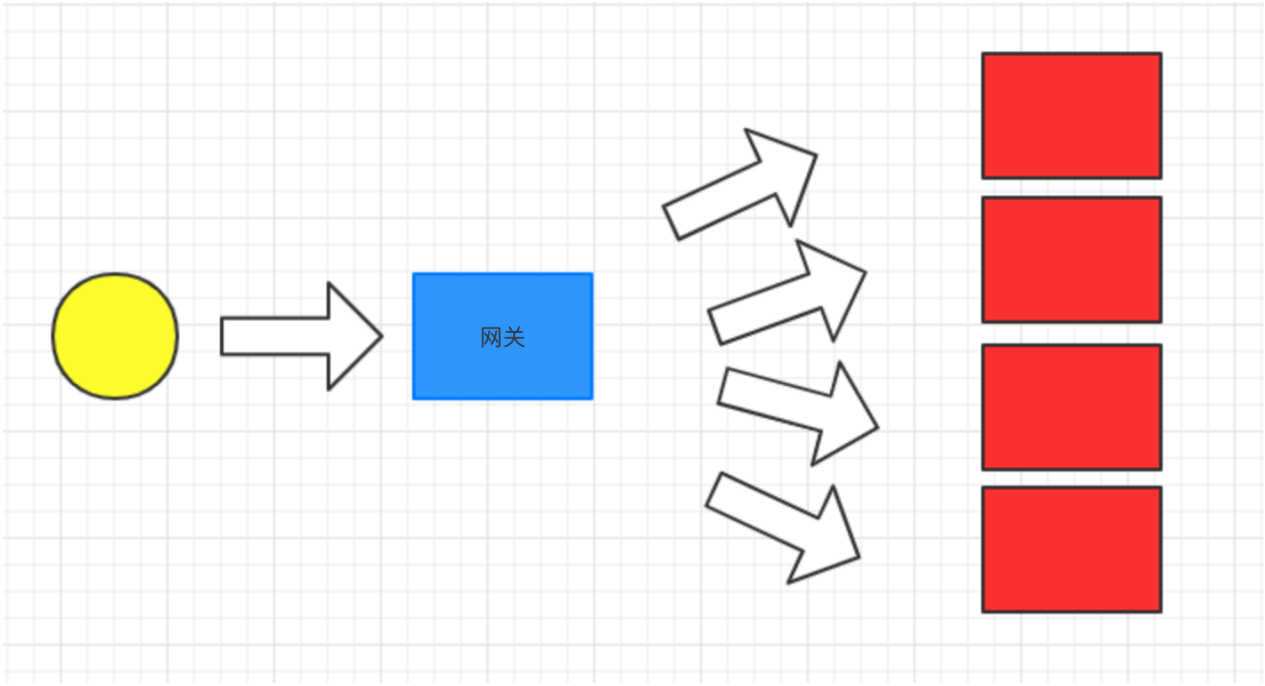
```
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.web.client.RestTemplate;

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

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

6、StudentHandler

```
package com.southwind.controller;

import com.southwind.entity.Student;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.client.RestTemplate;

import java.util.Collection;

@RestController
@RequestMapping("/consumer")
public class StudentHandler {

    @Autowired
    private RestTemplate restTemplate;

    @GetMapping("/findAll")
    public Collection<Student> findAll(){
        return
restTemplate.getForObject("http://localhost:8010/provider/findAll",Collection.
class);
    }

    @GetMapping("/findById/{id}")
    public Student findById(@PathVariable("id") Integer id){
        return
restTemplate.getForObject("http://localhost:8010/provider/findById/{id}",Stude
nt.class,id);
    }

    @PostMapping("/save")
```

```
    public void save(@RequestBody Student student){

 restTemplate.postForObject("http://localhost:8010/provider/save",student,Stud
ent.class);
    }

    @PutMapping("/update")
    public void update(@RequestBody Student student){
        restTemplate.put("http://localhost:8010/provider/update",student);
    }

    @DeleteMapping("/deleteById/{id}")
    public void deleteById(@PathVariable("id") Integer id){

 restTemplate.delete("http://localhost:8010/provider/deleteById/{id}",id);
    }
}
```

服务网关



API 网关可以对所有的 API 请求进行统一的管理维护，相当于为系统开放出一个统一的接口，所有的外部请求只需要统一访问这个外部入口即可，系统内部再通过高 API 网关去映射不同的微服务。

对于开发者而言就不需要关注具体的微服务 URL，直接访问网关即可。

Spring Cloud Zuul

1、创建 Module，pom.xml

```
<dependencies>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-netflix-zuul</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    </dependency>
</dependencies>
```

2、application.yml

```
server:
  port: 8030
spring:
  application:
    name: gateway
eureka:
  client:
    service-url:
      defaultZone: http://localhost:8761/eureka/
zuul:
  routes:
    provider: /p/**
```

3、创建启动类

```
package com.southwind;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.cloud.netflix.zuul.EnableZuulProxy;

@EnableZuulProxy
@EnableAutoConfiguration
public class GateWayApplication {
    public static void main(String[] args) {
        SpringApplication.run(GateWayApplication.class,args);
    }
}
```

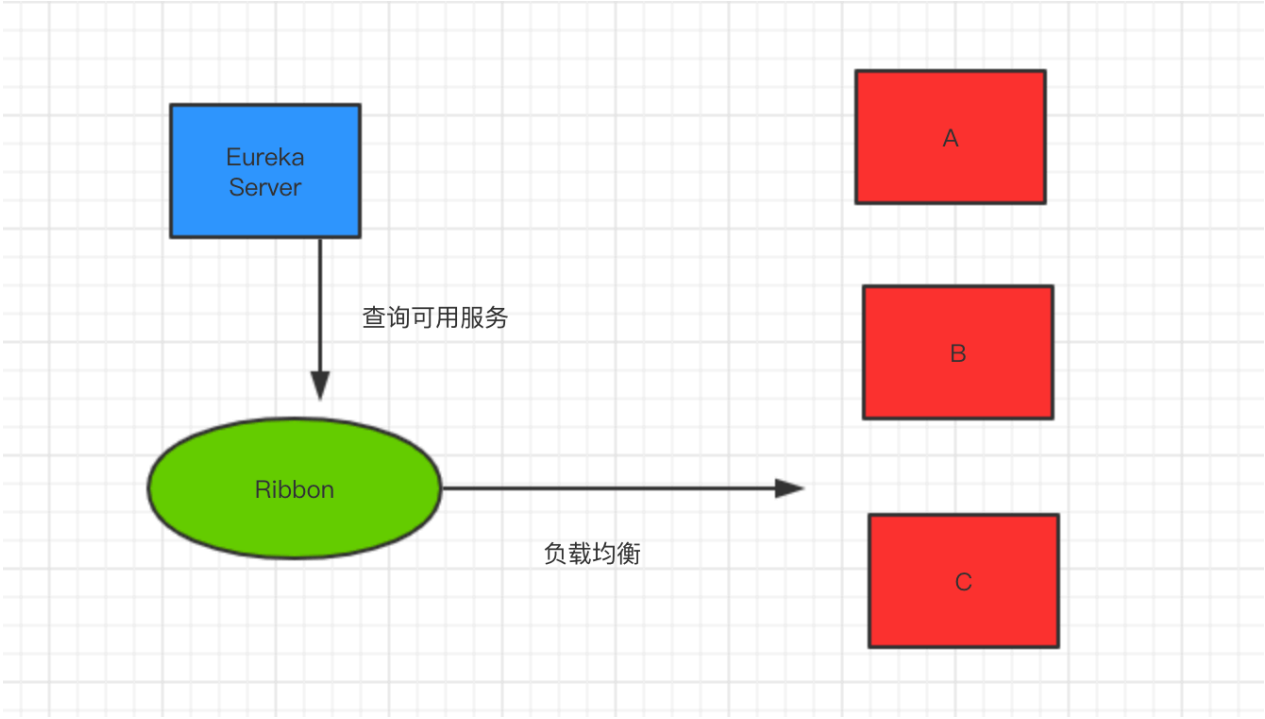
Zuul 处理路由映射之外，还自带了负载均衡功能。

Ribbon 负载均衡

Spring Cloud 提供的一种负载均衡解决方案，Ribbon 是 Netflix 发布的负载均衡器，Spring Cloud Ribbon。

Ribbon 的使用同样需要结合 Eureka Server

负载均衡算法：轮询、随机、加权轮询、加权随机



1、创建 Module，pom.xml

```
<dependencies>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    </dependency>
</dependencies>
```

2、application.yml

```
server:
  port: 8040
spring:
  application:
    name: ribbon
eureka:
  client:
    service-url:
      defaultZone: http://localhost:8761/eureka/
  instance:
    prefer-ip-address: true
```

3、创建启动类

```
package com.southwind;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.context.annotation.Bean;
import org.springframework.web.client.RestTemplate;

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

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

4、实体类

```
package com.southwind.entity;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class Student {
    private Integer id;
    private String name;
}
```

5、创建 Handler

```
package com.southwind.controller;

import com.southwind.entity.Student;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;
```

```java
import java.util.Collection;

@RestController
@RequestMapping("/ribbon")
public class RibbonHandler {

    @Autowired
    private RestTemplate restTemplate;

    @GetMapping("/findAll")
    public Collection<Student> findAll(){
        return
restTemplate.getForObject("http://provider/provider/findAll",Collection.class)
;
    }

    @GetMapping("/index")
    public String index(){
        return
restTemplate.getForObject("http://consumer/consumer/index",String.class);
    }
}
```