SpringCloud服务间的调用有两种方式：RestTemplate和FeignClient。不管是什么方式，他都是通过REST接口调用服务的http接口，参数和结果默认都是通过jackson序列化和反序列化。因为Spring MVC的RestController定义的接口，返回的数据都是通过Jackson序列化成JSON数据。

一、RestTemplate

使用这种方式，只需要定义一个RestTemplate的Bean，设置成LoadBalanced即可。

如下示例：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | @Configuration  public class SomeCloudConfiguration {  @LoadBalanced  @Bean  RestTemplate restTemplate() {      return new RestTemplate();  }  } |

这样我们就可以在需要用的地方注入这个bean使用：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | public class SomeServiceClass {  @Autowired  private RestTemplate restTemplate;  public String getUserById(Long userId) {      UserDTO results = restTemplate.getForObject("http://users/getUserDetail/" + userId, UserDTO.class);      return results;  }  } |

其它示例参考：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | @SpringBootApplication  public class SleuthClientApplication {        @Bean      public RestTemplate restTemplate() {          return new RestTemplate();      }        public static void main(String[] args) {          SpringApplication.run(SleuthClientApplication.class, args);      }  }    @RestController  class HomeController {        private static final Log log = LogFactory.getLog(HomeController.class);      @Autowired      private RestTemplate restTemplate;        private String url="http://localhost:9986";        @RequestMapping("/service1")      public String service1() throws Exception {          log.info("service1");          Thread.sleep(200L);          String s = this.restTemplate.getForObject(url + "/service2", String.class);          return s;      }  } |

二、FeignClient

 除了上面的方式，我们还可以用FeignClient。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | @FeignClient(value = "users", path = "/users")  public interface UserCompositeService {  @RequestMapping(value = "/getUserDetail/{id}",          method = RequestMethod.GET,          produces = MediaType.APPLICATION\_JSON\_VALUE)  UserDTO getUserById(@PathVariable Long id);  } |

  我们只需要使用@FeignClient定义一个接口，Spring Cloud Feign会帮我们生成一个它的实现，从相应的users服务获取数据。

  其中，@FeignClient(value = “users”， path = “/users/getUserDetail”)里面的value是服务ID，path是这一组接口的path前缀。在下面的方法定义里，就好像设置Spring MVC的接口一样，对于这个方法，它对应的URL是/users/getUserDetail/{id}。然后，在使用它的时候，就像注入一个一般的服务一样注入后使用即可：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | public class SomeOtherServiceClass {  @Autowired  private UserCompositeService userService;  public void doSomething() {      // .....      UserDTO results = userService.getUserById(userId);      // other operation...  }  } |