## 17. Declarative REST Client: Feign

[Feign](https://github.com/Netflix/feign)是一个声明式的Web服务客户端。它让使用Web服务更简单。要使用Feign，需要创建一个接口并添加注解。它提供插拔式的Feign注解和JAX-RS注解支持。Feign还支持可插拔编码器和解码器。Spring Cloud增加了对Spring MVC注解的支持，以及在Spring Web项目中默认使用同样的HttpMessageConverters。Spring Cloud集成Ribbon和Eureka以在使用Feign时提供负载均衡的http客户端。

## 17.1 How to Include Feign

要在您的项目中使用Feign，需要引入group org.springframework.cloud artifact ID spring-cloud-starter-feign。

*@Configuration*

*@ComponentScan*

*@EnableAutoConfiguration*

*@EnableFeignClients*

**public** **class** Application {

**public** **static** **void** main(String[] args) {

SpringApplication.run(Application.**class**, args);

}

}

**StoreClient.java.**

*@FeignClient("stores")*

**public** **interface** StoreClient {

*@RequestMapping(method = RequestMethod.GET, value = "/stores")*

List<Store> getStores();

*@RequestMapping(method = RequestMethod.POST, value = "/stores/{storeId}", consumes = "application/json")*

Store update(*@PathVariable("storeId")* Long storeId, Store store);

}

@FeignClient注解的value（上文中的“stores”）是客户端名字（微服务application name），这个名字被用于创建Ribbon负载平衡器（[有关Ribbon支持的详细信息，](https://springcloud.cc/spring-cloud-dalston.html#spring-cloud-ribbon)请参阅[下文](https://springcloud.cc/spring-cloud-dalston.html#spring-cloud-ribbon)））。您还可以使用注解的url属性（值可为绝对路径或主机名 http://ip:port/api）指定URL。这些接口在Application context中的实例化的bean名称是该接口的完全限定名称。要指定您自己的别名值，您可以使用@FeignClient注解的qualifier属性。

Ribbon客户端将会寻找“stores”服务的物理地址。如果您的应用程序是Eureka客户端，那么它将在Eureka服务注册中解析该服务。如果您不想使用Eureka，你可以配置 stores的服务列表（细节参考16.6）。

## 17.2 Overriding Feign Defaults

Spring Cloud的Feign支持的核心概念是声明式客户端。每个feign客户端都是feign全体组件的一部分，它们一起工作以根据需要联系远程服务器，并且该集合有一个名字用在@FeignClient注解中。Spring cloud 每一个使用 FeignClientsConfiguration 生成的声明式客户端创建一个application context。这里面包含 一个 feign.Decoder, 一个 feign.Encoder 和一个 feign.Contract.