REST Client with Spring Cloud Feign



Spring Cloud

- Spring Cloud Contract: supports and facilitates contract testing
- adapted for Spring:
 - Hystrix, Eureka, Ribbon, Feign, Zuul
- Spring Cloud Config Server: configuration as a service
- Spring Cloud Sleuth: distributed tracing

• Spring Cloud Netflix: umbrella project offering a number of Netflix services and libraries



Spring Cloud Feign

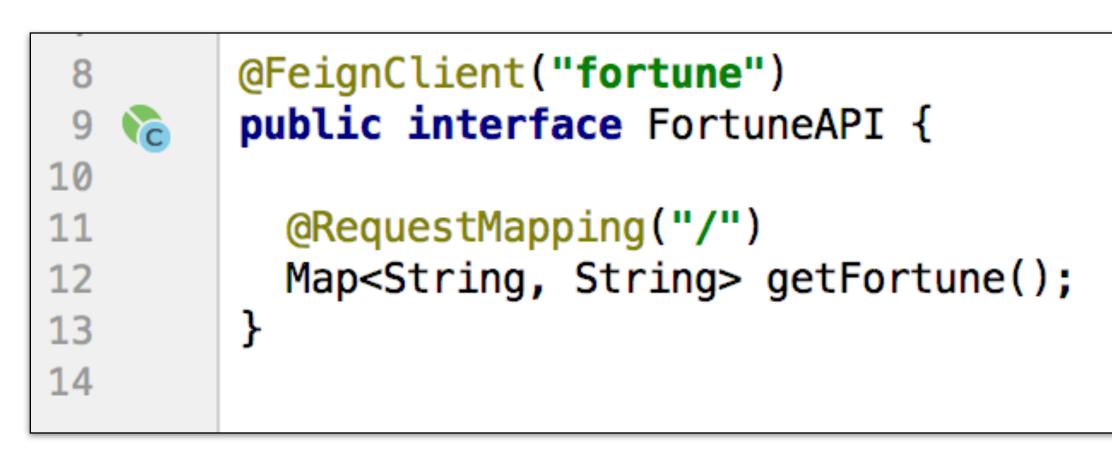
- Encapsulates the details of REST API calls behind an interface
- Integrated with Eureka and Ribbon

- Add dependency: spring-cloud-starter-feign 1.
- 2. Annotate Spring Boot application class with @EnableFeignClients
- 3. Define the interface

Procedure

Feign Interface: Simple Example

- Annotate class with @FeignClient and indicate the service id of the backing service this interface represents
- b. Map REST API calls to interface methods
- c. Annotate each method with the familiar @RequestMapping annotation

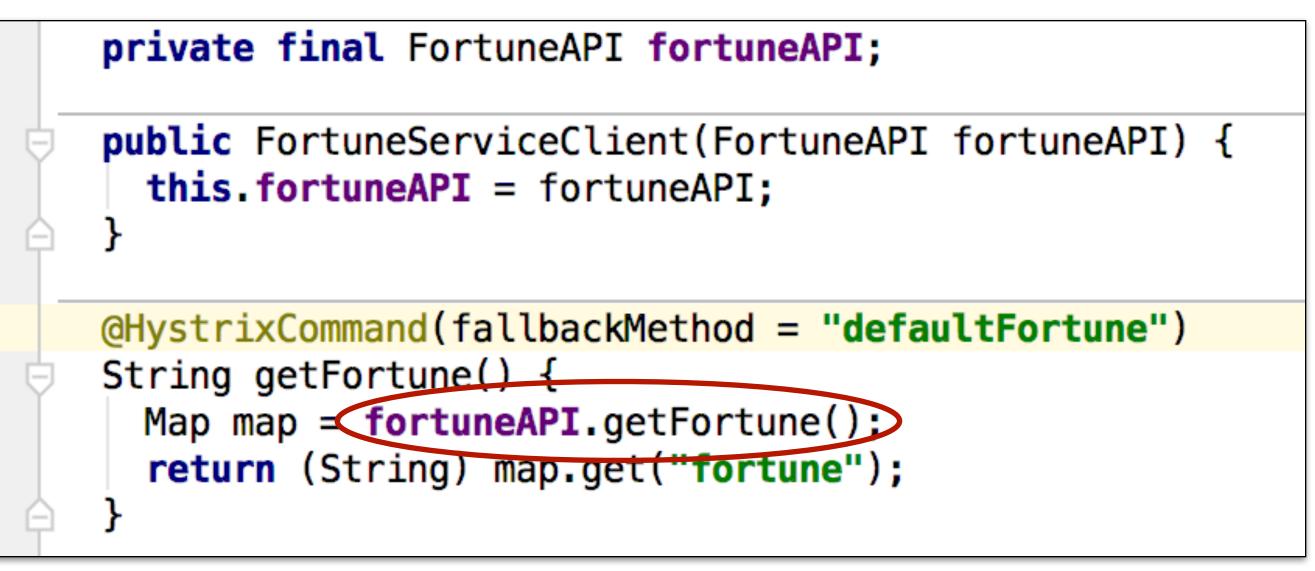




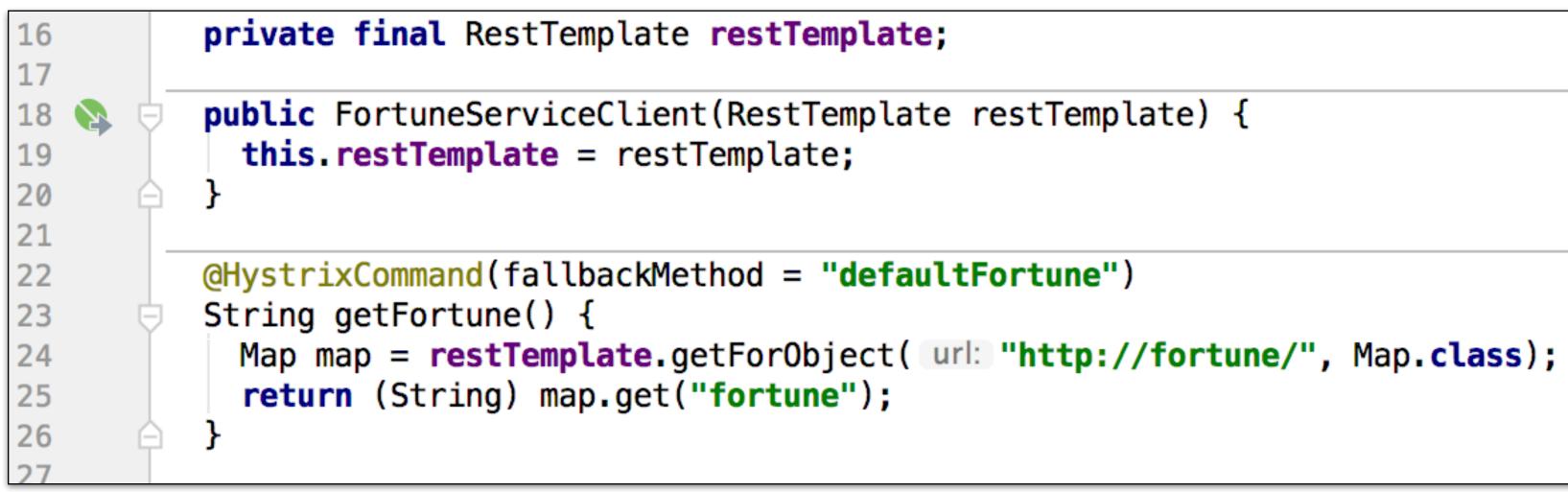
- Auto-wire the interface into any client class
- Anywhere a REST API call needs to be made against the backing service, just invoke the corresponding interface method instead
- Eureka url lookup and Ribbon load balancing still take place
- In addition we've neatly encapsulated a set of related REST API calls behind an interface



Usage



Contrast with RestTemplate





private final FortuneAPI fortuneAPI;

```
public FortuneServiceClient(FortuneAPI fortuneAPI) {
 this.fortuneAPI = fortuneAPI;
```

```
@HystrixCommand(fallbackMethod = "defaultFortune")
String getFortune() {
 Map map = fortuneAPI.getFortune();
  return (String) map.get("fortune");
```

