

微服务调用

- 1 本地调用其他微服务配置
 - 1.1 服务提供方
- 2 服务消费方
 - 2.1 增加bootstrap.yml文件
 - 2.2 修改application.yml
 - 2.3 增加application-local.yml
 - 2.4 修改代码生成器模板

微服务之间的调用采用feign客户端，由服务提供方提供feign的api，服务消费方，将依赖加入本地项目中即可采用java api的方式调用其他微服务。

本地调用其他微服务配置

由于本地没有注册服务器环境，无法自动获取微服务的ip地址，无法进行跨服务的调用。

本章内容描述了如何解决上述问题。

原理为，采用feign客户端可以手工定制url的方法，在服务提供方提供的feign api类中预留url的配置项，在服务消费方的本地配置文件中配置固定的url值即可。

服务提供方

服务提供方修改feignClient的内容，修改如下：

```
@Api(value = "MasAirline", tags = "MasAirline")
@FeignClient(value = "mas-global", url="${feign.url.mas-global}")
public interface MasAirlineApi {

    /**
     * id.
     *
     * @param id id
     * @return MasAirlineVO
     */
    @ApiOperation("id")
    @ApiImplicitParam(name = "id", value = "MasAirline id", required =
true, dataType = "String")
    @GetMapping("/masAirline/id/{id}")
    MasAirlineVO findOne(@PathVariable String id);

}
```

上面例子中是一个行业级主文件的微服务api，需要修改FeignClient中的配置，增加url="`${feign.url.mas-global}`"的内容。对于服务提供方来说，只需要修改这里。

服务消费方

增加bootstrap.yml文件

此步的目的是为了将系统中重要的不变的配置项放在首要启动的配置文件中，其他可能会变化的配置项放在其他配置文件中。

```
application.yml
```

修改application.yml

```
spring:
  profiles:
    active: local

spring.main.allow-bean-definition-overriding: true
```

注意有些在bootstrap中配置不生效的项，仍然保留在application.yml中

增加application-local.yml

```
feign:
  url:
    mas-global: http://10.1.17.141/opra-api/mas-global
```

将本地调用其他模块的api的url地址写在local配置文件中。可以配置多个模块。

修改代码生成器模板

修改api.ftl模板文件，替换原来FeignClient为

```
@FeignClient(value = "${package.ModuleName}", url = "${r'${feign.url}'.${package.ModuleName}:}")
```