在微服务系统中，我们把系统拆分成了很多小的服务，各个服务之间通过不同的方式进行依赖和调用，为了保证服务的高可用性，单个服务通常会集群部署，此时，许多服务由于各种问题可能会调用失败，比如超时、异常等。如何能够保证在一个依赖出问题的情况下，不会出现故障传播，导致整体服务失败，这个就是Hystrix需要做的事情。Hystrix提供了熔断、隔离、Fallback、cache、监控等功能，能够在一个、或多个依赖同时出现问题时保证系统依然可用。

如何结合负载均衡（Feign），使用Hystrix来实现熔断功能。

在feign-server项目的基础上添加熔断实现。

Feign中实现熔断功能非常简单，只需要在@FeignClient中指定fallback的实现即可，然后在接口的实现类中添加熔断逻辑。

service：

*/\*\**

*\* 通过注解来指定调用哪个服务*

*\* 如果开启了断路器，只需要添加fallback，在此接口实现类中添加熔断方法即可*

*\*/*

@FeignClient(value = "company-server",fallback = FeignServiceImpl.class)

public interface FeignService {

*/\*\**

*\* 调用company-server服务中的company/get接口*

*\**

*\** ***@param*** *id*

*\** ***@return***

*\*/*

@GetMapping(value = "company/get")

String getCompany(@RequestParam(value = "id") String id);

}

serviceImpl：

添加断路器的实现逻辑，实现熔断方法。

@Component

public class FeignServiceImpl implements FeignService {

*/\*\**

*\* 熔断方法*

*\**

*\** ***@param*** *id*

*\** ***@return***

*\*/*

@Override

public String getCompany(String id) {

return "服务出错，返回指定企业信息：华为" + id;

}

}

启动项目，测试

启动，访问接口<http://localhost:8015/getCompany?id=36>，可以看到服务会自动负载到两台服务上。

如果挂掉8011服务，可以看到访问结果如下：



