

```

1 package com.itmayiedu.fallback;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import org.springframework.stereotype.Component;
7
8 import com.itmayiedu.service.MemberFeign;
9
10 @Component
11 public class MemberFallBack implements MemberFeign {
12
13     public List<String> getToOrderMemberAll() {
14         //服务降级处理
15         List<String> list = new ArrayList<String>();
16         list.add("服务发生异常...");
17         return list;
18     }
19
20 }

```

EurekaApp [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (2018年2月28日 下午9:10:01)
 2018-02-28 22:30:17.830 WARN 23000 --- [eerNodesUpdater] c.n
 2018-02-28 22:30:18.629 INFO 23000 --- [a-EvictionTimer] c.n
 2018-02-28 22:31:18.646 INFO 23000 --- [a-EvictionTimer] c.n
 2018-02-28 22:32:18.647 INFO 23000 --- [a-EvictionTimer] c.n

什么是服务降级

所有的RPC技术里面服务降级是一个最为重要的话题，所谓的降级指可使用的时候，程序不会出现异常，而会出现本地的操作调用。

service-order 工程新增 Maven 依赖

```

<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-hystrix</artifactId>
</dependency>

```

Rest 方式使用断路器

(每特学院&蚂蚁课堂) 上海每特教育科技有限公司 www.itmayiedu.com

(每特学院&蚂蚁课堂) 上海每特教育科技有限公司 www.itmayiedu.com

1. 断路器机制。

断路器很好理解，当 Hystrix Command 请求后端服务失败数量超过路器会切换到开路状态(Open)。这时所有请求会直接失败而不会发保持在开路状态一段时间后(默认5秒)，自动切换到半开路状态(半闭合)。如果下一次请求的返回情况，如果请求成功，断路器切回闭路状态(CLOSED)。Hystrix的断路器就像我们家庭电路中的保险丝，断路器会直接切断请求链，避免发送大量无效请求影响系统吞吐量测并恢复的能力。

2. Fallback。

Fallback相当于是降级操作。对于查询操作，我们可以实现一个方法，在服务出现异常的时候，可以使用 fallback 方法返回的值。fallback 设置的默认值或者来自缓存。

3. 资源隔离。

在 Hystrix 中，主要通过线程池来实现资源隔离。通常在使用的线程服务划分出多个线程池，例如调用产品服务的 Command 放入 A 线程池，调用支付服务的 Command 放入 B 线程池。这样做的主要优点是运行环境被隔离开来了。如果存在 bug 或者由于其他原因导致自己所在线程池被耗尽时，不会影响。但是带来的代价就是维护多个线程池会对系统带来额外的性能要求而且确保自己调用服务的客户端代码不会出问题的话，信号模式(Semaphores)来隔离资源。

1. 服务雪崩产生服务堆积，导致其他服务接口无法。
2. 如何解决服务雪崩效应
3. 超时机制--服务降级处理
4. 服务降级--服务接口发生错误，不去调用接口，调用本地方法 fallback
5. 垮架机制--类似保险丝

什么是熔断机制。

熔断机制，就是下游服务出现问题后，为保证整个系统正常运行下的机制，通过返回缓存数据或者既定数据，避免出现系统整体雪崩中，该功能可通过配置的方式加入到项目中。

Hystrix 作用。

(每特学院&蚂蚁课堂) 上海每特教育科技有限公司 www.itmayiedu.com

1. 断路器机制。

断路器很好理解，当 **Hystrix Command** 请求后端服务失败数量超过路器会切换到开路状态 (**Open**)。这时所有请求会直接失败而不会发保持在开路状态一段时间后(默认 5 秒)，自动切换到半开路状态 (**H**) 下一次请求的返回情况，如果请求成功，**断路器**切回闭路状态 (**CLOSED**)。开路状态 (**OPEN**)。Hystrix 的断路器就像我们家庭电路中的保险丝，

2. 解决或缓解服务雪崩的方案。

一般情况对于服务依赖的保护主要有 3 中解决方案：

(1) 熔断模式：这种模式主要是参考电路熔断，如果一条线路电压防止火灾。放到我们的系统中，如果某个目标服务调用慢或者有大量务的调用，对于后续调用请求，不在继续调用目标服务，直接返回，标服务情况好转则恢复调用。

(2) 隔离模式：这种模式就像对系统请求按类型划分成一个个小岛火少光了，不会影响到其他的小岛。例如可以对不同类型的请求使用种类型的请求互不影响，如果一种类型的请求线程资源耗尽，则对后回，不再调用后续资源。这种模式使用场景非常多，例如将一个服务使用单独服务器来部署，再或者公司最近推广的多中心。

(3) 限流模式：上述的熔断模式和隔离模式都属于出错后的容错处可以称为预防模式。限流模式主要是提前对各个类型的请求设置最高置的阀值则对该请求直接返回，不再调用后续资源。这种模式不能解能解决系统整体资源分配问题，因为没有被限流的请求依然有可能让

3. 熔断设计

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "Java EE - itmayedu-order-fgin".
- Editor:** Displays the `MemberFeign.java` file with its corresponding `application.yml` configuration. The configuration includes Eureka client and server settings, and a Spring application definition.
- Outline:** Shows the class hierarchy and member definitions.
- Console:** Displays log output from the application, showing INFO messages related to Eureka registry and eviction timers.
- Task List:** Shows a list of tasks or errors.

```
1 eureka:
2   client:
3     serviceUrl:
4       defaultZone: http://localhost:8888/eureka/
5 server:
6   port: 8765
7   tomcat:
8     max-threads: 50
9
10 spring:
11   application:
12     name: service-order-feign
13
```

```
2018-02-28 21:58:59.954 INFO 23000 --- [io-8888-exec-10] c.n.e.registry.AbstractEurekaServer : 2018-02-28 21:59:18.561 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer : 2018-02-28 22:00:17.782 WARN 23000 --- [eerNodesUpdater] c.n.eureka.cluster.PeerNodesUpdater : 2018-02-28 22:00:18.562 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer : 2018-02-28 22:01:18.562 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer : 2018-02-28 22:02:18.563 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer : 2018-02-28 22:03:18.563 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer : 2018-02-28 22:04:18.565 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer : 2018-02-28 22:05:18.565 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer :
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "Java EE - service-member".
- Editor:** Displays the `MemberController.java` file, which contains Java code for a REST controller. It includes annotations like `@RestController` and `@RequestMapping`.
- Console:** Displays log output from the application, showing INFO messages related to Eureka registry and eviction timers.
- Task List:** Shows a list of tasks or errors.

```
1 package com.itmayedu.controller;
2
3 import java.util.ArrayList;
4
5 @RestController
6 public class MemberController {
7   @Value("${server.port}")
8   private String serverPort;
9   private static int count = 0;
10
11   @RequestMapping("/getMemberAll")
12   public List<String> getMemberAll() {
13     try {
14       Thread.sleep(3000);
15     } catch (Exception e) {
16       // TODO: handle exception
17     }
18     count++;
19     List<String> listUser = new ArrayList<String>();
20     listUser.add("zhangsan");
21     listUser.add("lisi");
22   }
23 }
```

```
2018-02-28 21:35:18.497 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer : 2018-02-28 21:36:18.498 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer : 2018-02-28 21:37:18.498 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer : 2018-02-28 21:38:18.498 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer : 2018-02-28 21:39:18.499 INFO 23000 --- [a-EvictionTimer] c.n.e.registry.AbstractEurekaServer :
```

订单功能需要订单会员工
tomcat服务器

什么是服务雪崩效应?

客户端 → Tomcat最大线程数50个
order/getOrder 依然服务
order/addOrder

会员月
每次3秒进
user/get
user/login

雪崩效应，所有请求在处理一个服务，不能访问其他服务接口。

1. 使用超时机制、服务降级()
2. 服务降级服务调用接口，如果发生错误或者超时，不让调用接口，调用本地fallback

流量控制的具体措施包括：

- 网关限流
- 用户交互限流

断路器(Hystrix)

为什么需要 Hystrix?

在微服务架构中，我们将业务拆分成一个个的服务，服务与服务之间可以相互调用 (RPC)。为了保证其高可用，单个服务又必须集群部署。由于网络原因或者自身的原因，服务并不能保证服务的100%可用，如果单个服务出现问题，调用这个服务就会出现网络延迟，此时如果有大量的网络涌入，会形成任务堆积，导致服务瘫痪，甚至导致服务“雪崩”。为了解决这个问题，就出现断路器模型。

Hystrix 是一个帮助解决分布式系统交互时超时处理和容错的类库，它同样拥有保护系统的能力。什么是服务雪崩。

分布式系统中经常会出现某个基础服务不可用造成整个系统不可用的情况，这种现象称为服务雪崩效应。为了应对服务雪崩，一种常见的做法是手动服务降级，而 **Hystrix** 的实现，给我们提供了另一种选择。

服务雪崩应对策略

针对造成服务雪崩的不同原因，可以使用不同的应对策略：

1. 流量控制
2. 改进缓存模式
3. 服务自动扩容
4. 服务调用者降级服务

1.服务雪崩的原因

(1) 某几个机器故障：例如机器的硬驱动引起的错误，或者一个bug（如，内存中断或者死锁）。

(2) 服务器负载发生变化：某些时候服务会因为用户行为造成雪崩，例如阿里的双十一活动，若没有提前增加机器预估流量则会挂掉。

(3) 人为因素：比如代码中的路径在某个时候出现bug。

2.解决或缓解服务雪崩的方案

一般情况对于服务依赖的保护主要有3中解决方案：

(1) 熔断模式：这种模式主要是参考电路熔断，如果一条线路

·断路器(Hystrix)

为什么需要 Hystrix?

在微服务架构中，我们将业务拆分成一个个的服务，服务与服务为了保证其高可用，单个服务又必须集群部署。由于网络原因或保证服务的100%可用，如果单个服务出现问题，调用这个服务有大量的网络涌入，会形成任务累计，导致服务瘫痪，甚至导致整个问题，就出现断路器模型。

Hystrix是一个帮助解决分布式系统交互时超时处理和容错的能力。什么是服务雪崩。

分布式系统中经常会出现某个基础服务不可用造成整个系统不可用为服务雪崩效应。为了应对服务雪崩，一种常见的做法是手动服务，给我们提供了另一种选择。

·服务雪崩应对策略

针对造成服务雪崩的不同原因，可以使用不同的应对策略；

服务消费者（Feign）

什么是 Feign

Feign 是一个声明式的伪 Http 客户端，它使得写 Http 客户端变得简单。需要创建一个接口并注解。它具有可插拔的注解特性，可使用 Feign 提供的注解。

Feign 支持可插拔的编码器和解码器。Feign 默认集成了 Ribbon，实现了负载均衡的效果。

简而言之：

- Feign 采用的是基于接口的注解。
- Feign 整合了 ribbon。

创建 service-order-feign 工程

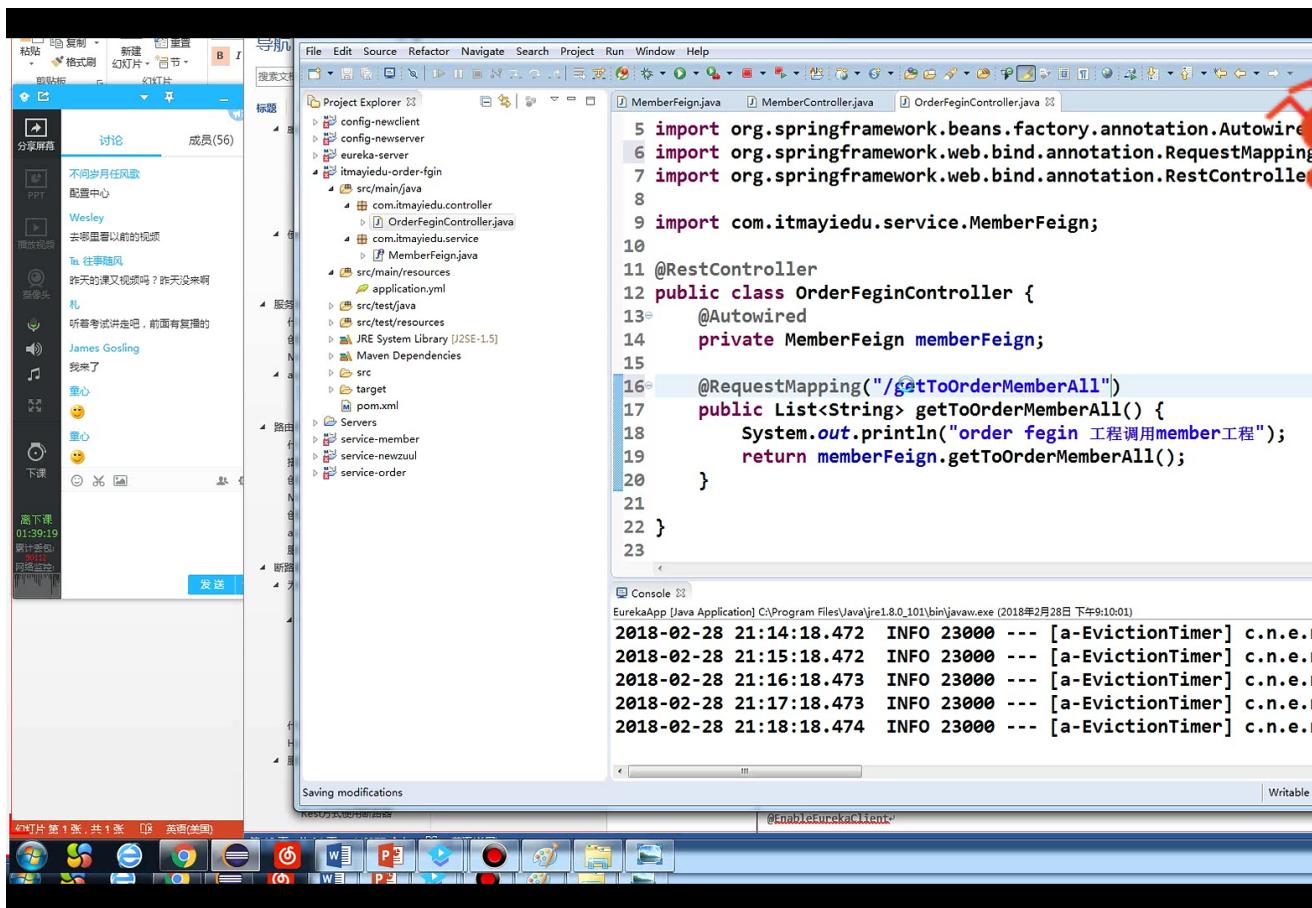
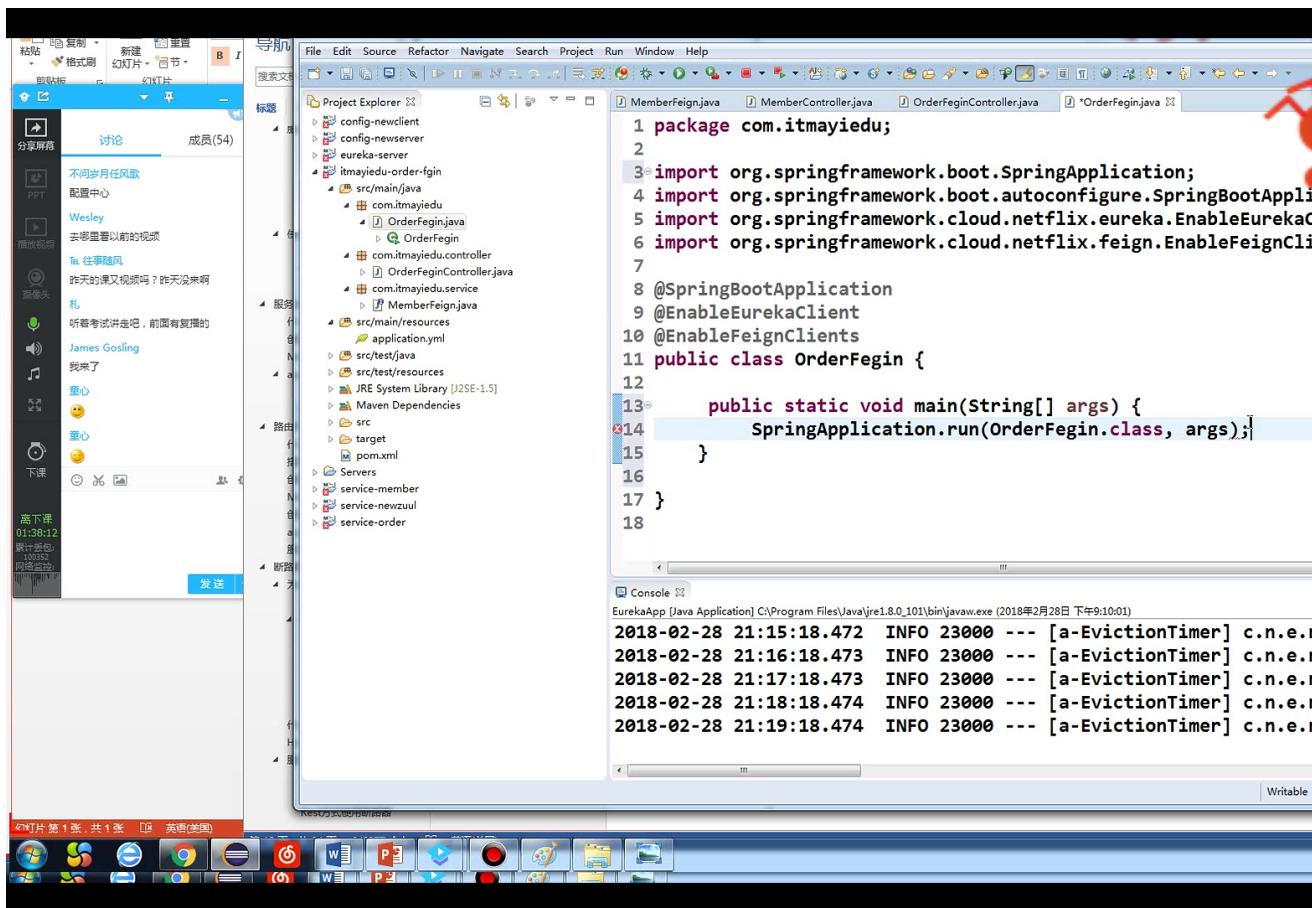
Maven 依赖

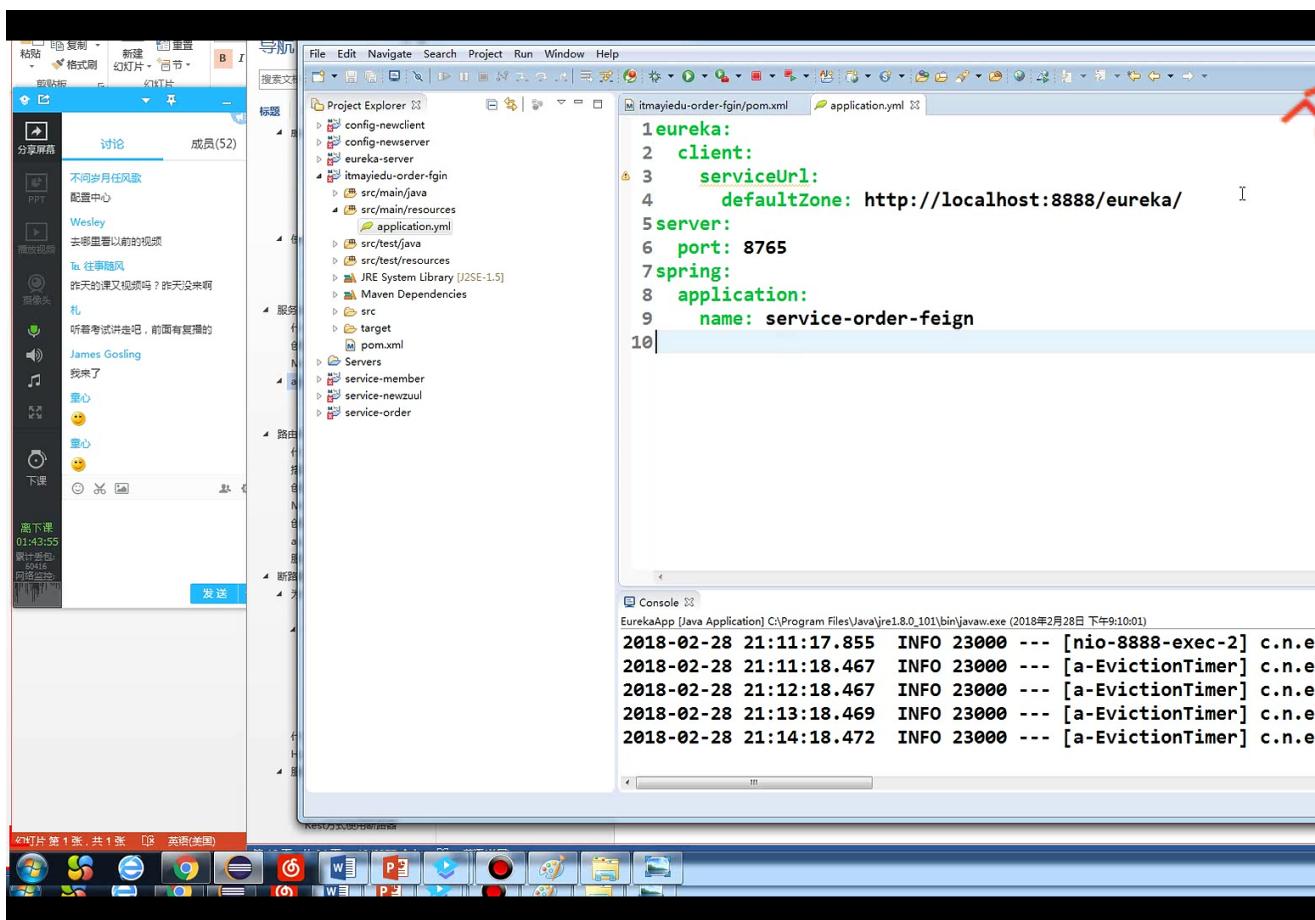
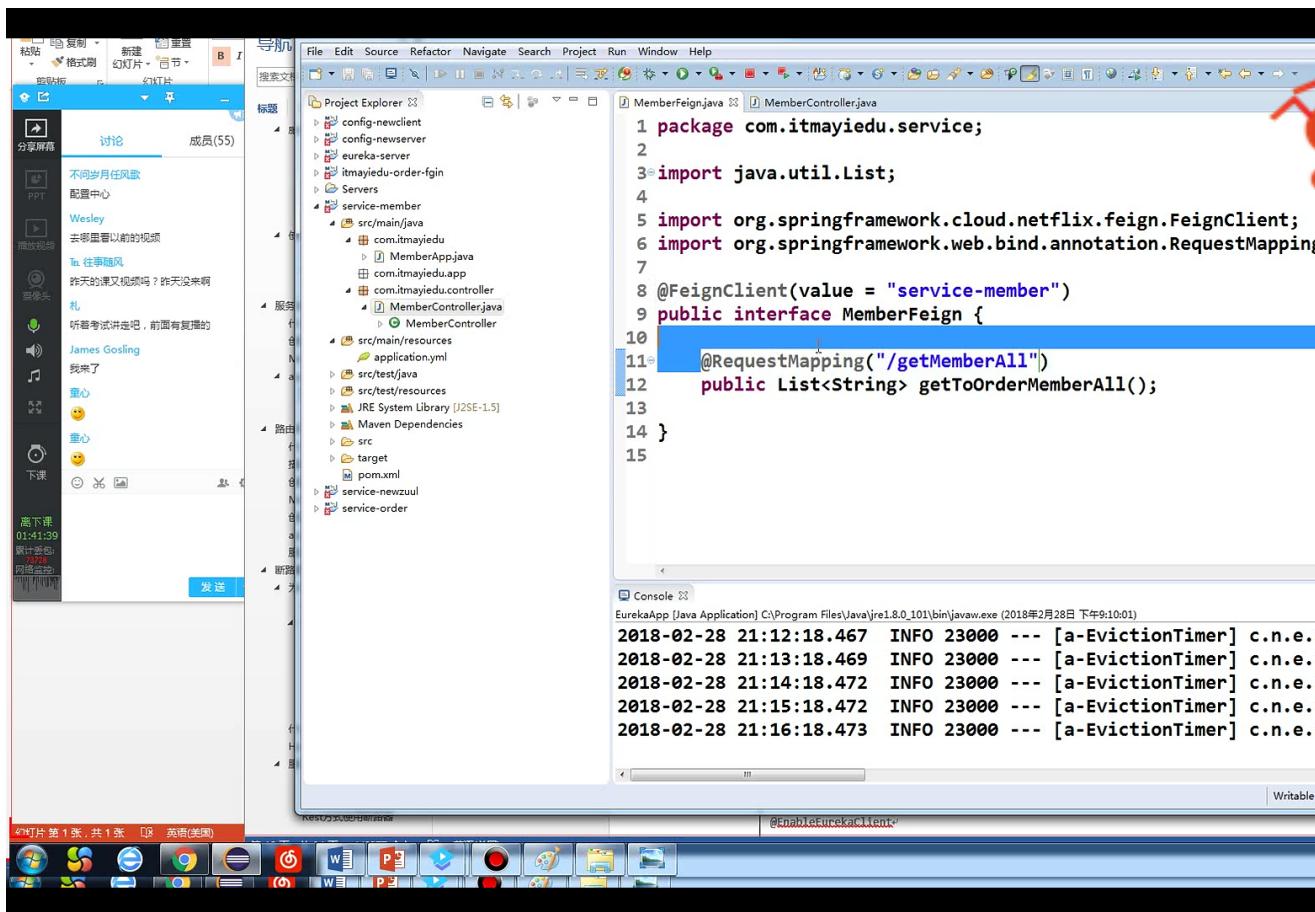
```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.5.2.RELEASE</version>
  <relativePath /> <!-- lookup parent from repository -->
</parent>
```

SpringCloud hystrix 熔断机制、服务降级、服务限流、解决服务雪崩效应

什么是服务雪崩效应？

The diagram shows a client sending 50 requests to an Order service (Tomcat server). The Order service then forwards these requests to a User service. The User service has a dependency on a Login service, which is called every 3 seconds. This causes a cascading failure where all services fail simultaneously due to high load.





文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记 设计 布局

导航

搜索文档

标题 | 页面 | 结果

- ◀ 服务消费者
 创建项目service-or...
 Maven依赖
 application.yml配置
 编写service,调用ser...
 演示效果
- ◀ 使用ribbon实现负载均衡
 什么是ribbon
 修改会员服务工程代...
 开启ribbon
- ◀ 服务消费者 (Feign)
 什么是Feign
 创建service-order-fei...
 Maven依赖
- ◀ application.yml配置
 编写service,调用ser...
 演示效果
- ◀ 路由网关(zuul)
 什么是网关
 搭建SpringCloud网关
 创建工作工程zuul
 Maven依赖
 创建工作工程service-zuul
 application.yml配置
 服务过滤
- ◀ 断路器(hystrix)
 为什么要 Hystrix?
 服务雪崩应对策略
 服务雪崩解决办法
 1.服务雪崩的原因
 2.解决或缓解服务...
 3.熔断设计
 4.隔离设计
 5.超时机制设计
 什么要熔断机制
 Hystrix使用
- ◀ 服务的降级
 什么是服务降级
 service-order工程...
 Rest方式使用断路器

```
</plugins>
</build>

<repositories>
    <repository>
        <id>spring-milestones</id>
        <name>Spring Milestones</name>
        <url>https://repo.spring.io/milestone</url>
    </repository>
</repositories>
```

(每特学院&蚂蚁课堂) 上海每特教育科技有限公司 www.itmayiedu.com 余老师 QQ644064779

```
<snapshots>
    <enabled>false</enabled>
</snapshots>
</repository>
</repositories>
```

▪ application.yml 配置

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记 设计 布局

导航

搜索文档

标题 | 页面 | 结果

- ◀ 服务消费者
 创建项目service-or...
 Maven依赖
 application.yml配置
 编写service,调用ser...
 演示效果
- ◀ 使用ribbon实现负载均衡
 什么是ribbon
 修改会员服务工程代...
 开启ribbon
- ◀ 服务消费者 (Feign)
 什么是Feign
 创建service-order-fei...
 Maven依赖
- ◀ application.yml配置
 编写service,调用ser...
 演示效果
- ◀ 路由网关(zuul)
 什么是网关
 搭建SpringCloud网关
 创建工作工程zuul
 Maven依赖
 创建工作工程service-zuul
 application.yml配置
 服务过滤
- ◀ 断路器(hystrix)
 为什么要 Hystrix?
 服务雪崩应对策略
 服务雪崩解决办法
 1.服务雪崩的原因
 2.解决或缓解服务...
 3.熔断设计
 4.隔离设计
 5.超时机制设计
 什么要熔断机制
 Hystrix使用
- ◀ 服务的降级
 什么是服务降级
 service-order工程...
 Rest方式使用断路器

~~```
<dependencyManagement>
 <dependencies>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-dependencies</artifactId>
 <version>Balaton.RC1</version>
 <type>POM</type>
 <scope>import</scope>
 </dependency>
 </dependencies>
</dependencyManagement>

<build>
 <plugins>
 <plugin>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-maven-plugin</artifactId>
 </plugin>
 </plugins>
</build>

<repositories>
 <repository>
 <id>spring-milestones</id>
 <name>Spring Milestones</name>
 <url>https://repo.spring.io/milestone</url>
 </repository>
</repositories>
```~~

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记 设计 布局

导航

标题 | 页面 | 结果

- ◀ 服务消费者
  - 创建项目service-or...  
Maven依赖  
application.yml配置  
编写service,调用ser...  
演示效果
  - ◀ 使用ribbon实现负载均衡
    - 什么是ribbon  
修改会员服务工程代...  
开启ribbon
  - ◀ 服务消费者 (Feign)
    - 什么公Feign  
创建service-order-fei...  
**Maven依赖**
    - ◀ application.yml配置
      - 编写service,调用ser...  
演示效果
  - ◀ 路由网关(zuul)
    - 什么公网关  
搭建SpringCloud网关  
创建工作(service-zuul  
Maven依赖  
创建工作(service-zuul  
application.yml配置  
服务过滤
  - ◀ 断路器(hystrix)
    - 为什么需要 Hystrix?  
服务雪崩应对策略
    - ◀ 服务雪崩解决办法
      - 1.服务雪崩的原因  
2.解决方案或...  
3.熔断设计  
4.隔离设计  
5.超时机制设计
      - 什么是熔断机制  
Hystrix使用
    - ◀ 服务的降级
      - 什么是服务降级  
service-order工程...  
Rest方式使用断路器

(每特学院&蚂蚁课堂) 上海每特教育科技有限公司 [www.itmayiedu.com](http://www.itmayiedu.com) 余老师 QQ644064779

```
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-ribbon</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-feign</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
```

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记

导航

标题 | 页面 | 结果

- ◀ 服务消费者
  - 创建项目service-or...  
Maven依赖  
application.yml配置  
编写service,调用ser...  
演示效果
  - ◀ 使用ribbon实现负载均衡
    - 什么是ribbon  
修改会员服务工程代...  
开启ribbon
  - ◀ 服务消费者 (Feign)
    - 什么公Feign  
**创建service-order-fei...  
Maven依赖**
    - ◀ application.yml配置
      - 编写service,调用ser...  
演示效果
  - ◀ 路由网关(zuul)
    - 什么公网关  
搭建SpringCloud网关  
创建工作(service-zuul  
Maven依赖  
创建工作(service-zuul  
application.yml配置  
服务过滤
  - ◀ 断路器(hystrix)
    - 为什么需要 Hystrix?  
服务雪崩应对策略
    - ◀ 服务雪崩解决办法
      - 1.服务雪崩的原因  
2.解决方案或...  
3.熔断设计  
4.隔离设计  
5.超时机制设计
      - 什么是熔断机制  
Hystrix使用
    - ◀ 服务的降级
      - 什么是服务降级  
service-order工程...  
Rest方式使用断路器

## 创建 service-order-feign 工程

### Maven 依赖

```
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>1.5.2.RELEASE</version>
<relativePath /> <!-- lookup parent from repository -->
</parent>

<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
<java.version>1.8</java.version>
</properties>

<dependencies>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
<dependency>
```

• 断路器(Hystrix)

## 为什么需要 Hystrix?

在微服务架构中，我们将业务拆分成一个个的服务，服务与服务之间可以相互调用 (RPC)。为了保证其高可用，单个服务又必须集群部署。由于网络原因或者自身的原因，服务并不能保证服务的 100% 可用，如果单个服务出现问题，调用这个服务就会出现网络延迟，此时若有大量的网络涌人，会形成任务累计，导致服务瘫痪，甚至导致服务“雪崩”。为了解决这个问题，就出现断路器模型。

Hystrix 是一个帮助解决分布式系统交互时超时处理和容错的类库，它同样拥有保护系统的能力。什么是服务雪崩？

分布式系统中经常会出现某个基础服务不可用造成整个系统不可用的情况，这种现象被称为服务雪崩效应。为了应对服务雪崩，一种常见的做法是手动服务降级，而 Hystrix 的出现，给我们提供了另一种选择。

## 服务雪崩应对策略

针对造成服务雪崩的不同原因，可以使用不同的应对策略：

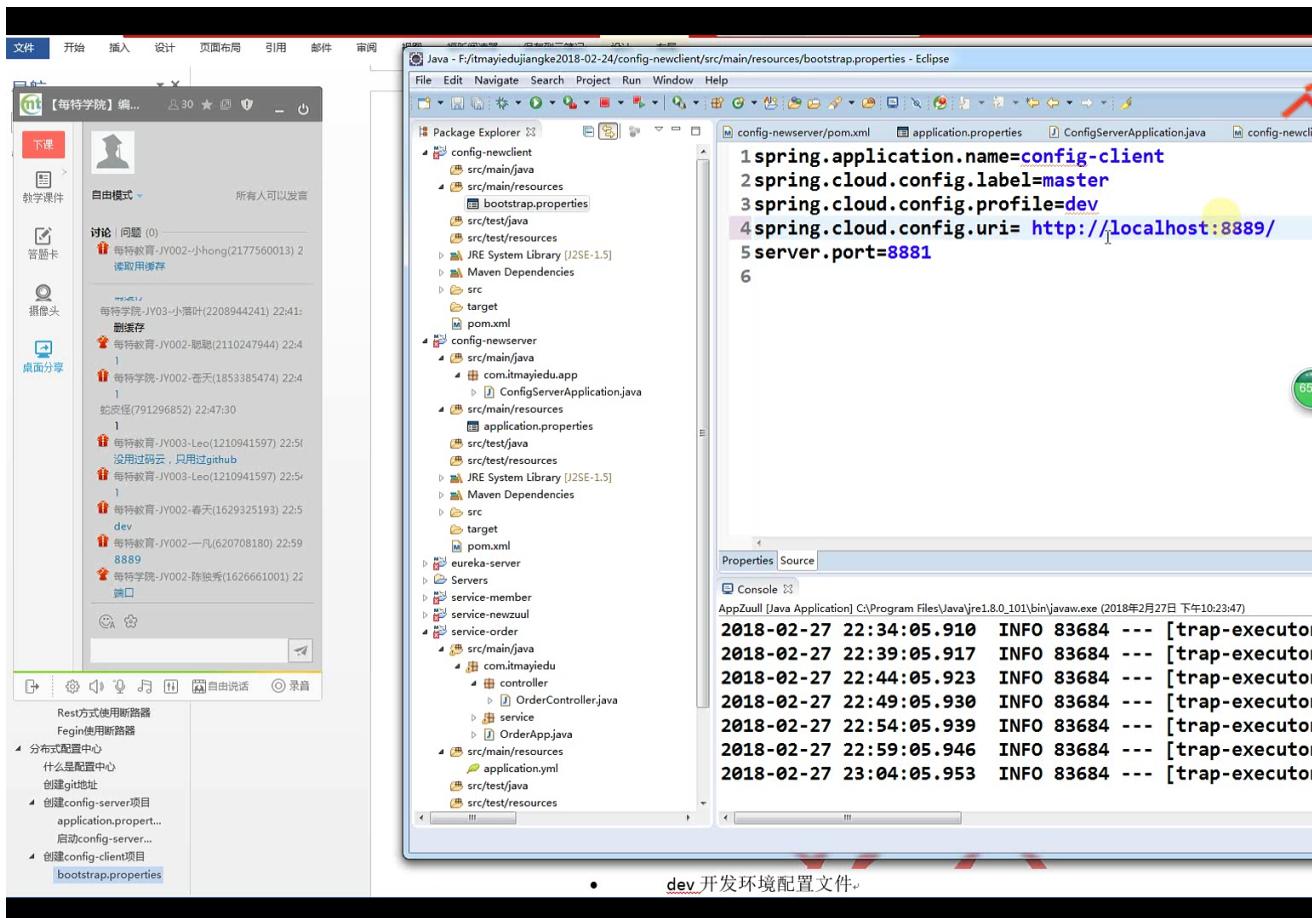
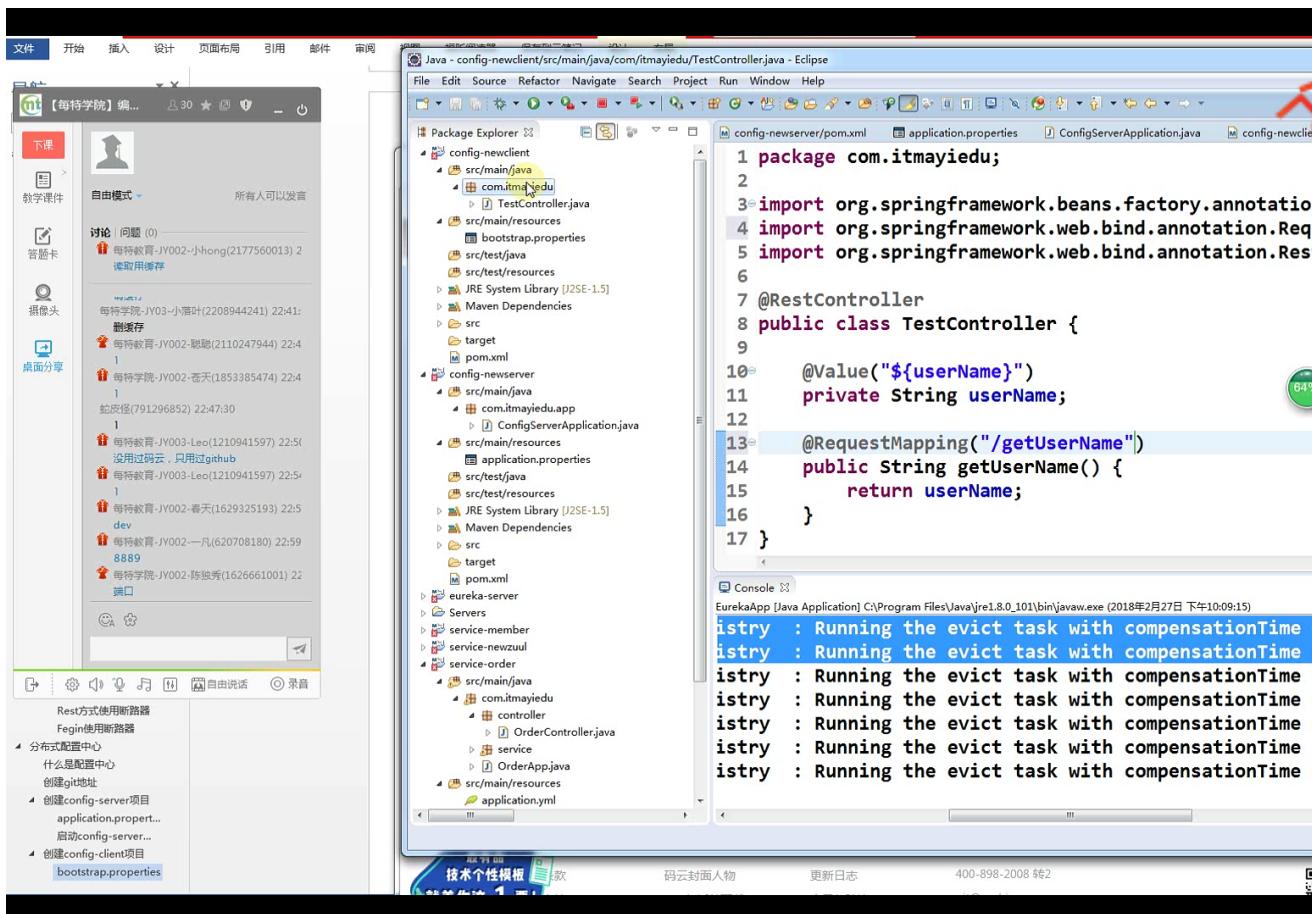
1. 流量控制。
2. 改进缓存模式。
3. 服务自动扩容。
4. 服务调用者降级服务。

讨论 (成员: 64)

- 不向岁月任风歌
- ribbon
- Jetty
- ribbon
- 不向岁月任风歌
- zuul
- 不向岁月任风歌
- 配置中心
- Wesley
- 去哪里看以前的视频
- Ta 往事随风
- 昨天的事又视频吗？昨天没来啊
- 礼
- 听说考试讲进吧，前面有直播的

1. SpringCloud 微服务解决方案 RPC 远程调用  
2. eureka 注册中心 ribbon 负载均衡客户端 zuul 网关 分布式配置中心  
3. 客户端调用工具 rest feign feign 客户端调用，SpringCloud 断路器 Hystrix  
服务降级、熔断机制、限流





文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记

导航 搜索文档

标题 页面 结果

- application.yml 编写service,调用ser... 演示效果
- 使用ribbon实现负载均衡 什么是ribbon 修改会员服务工程代... 开启ribbon
- 服务消费者(Feign) 什么是Feign 创建service-order-fei... Maven依赖
- application.yml配置 编写service,调用ser... 演示效果
- 路由网关(zuul) 什么是网关 基于SpringCloud网关 创建工程service-zuul Maven依赖 创建工程service-zuul application.yml配置 服务过滤
- 断路器(Hystrix) 为什么要Hystrix? 什么是服务雪崩 服务雪崩应对策略 Hystrix作用
- 服务的降级 什么是服务降级 service-order工程... Rest方式使用断路器 Feign使用断路器
- 分布式配置中心 什么是配置中心 创建git地址
- 创建config-server项目 application.propert... 启动config-server...
- 创建config-client项目 bootstrap.properties

(每特学院&蚂蚁课堂)上海每特教育科技有限公司 [www.itmayiedu.com](http://www.itmayiedu.com) 余老师 QQ644064779

```
<repository>
 <id>spring-milestones</id>
 <name>Spring Milestones</name>
 <url>https://repo.spring.io/milestone</url>
 <snapshots>
 <enabled>false</enabled>
 </snapshots>
</repository>
</repositories>
```

## bootstrap.properties

```
spring.application.name=config-client
spring.cloud.config.label=master
```

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记

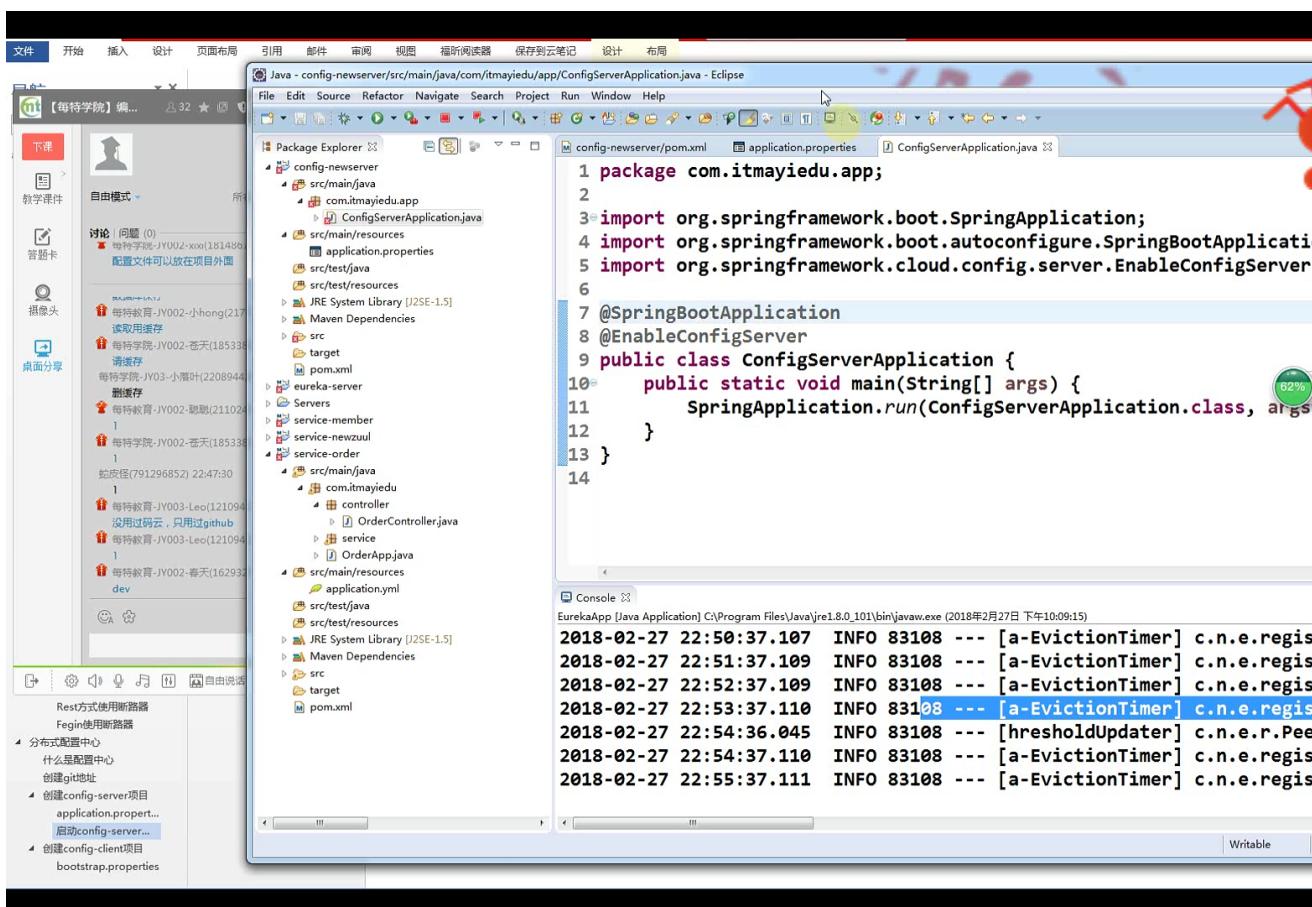
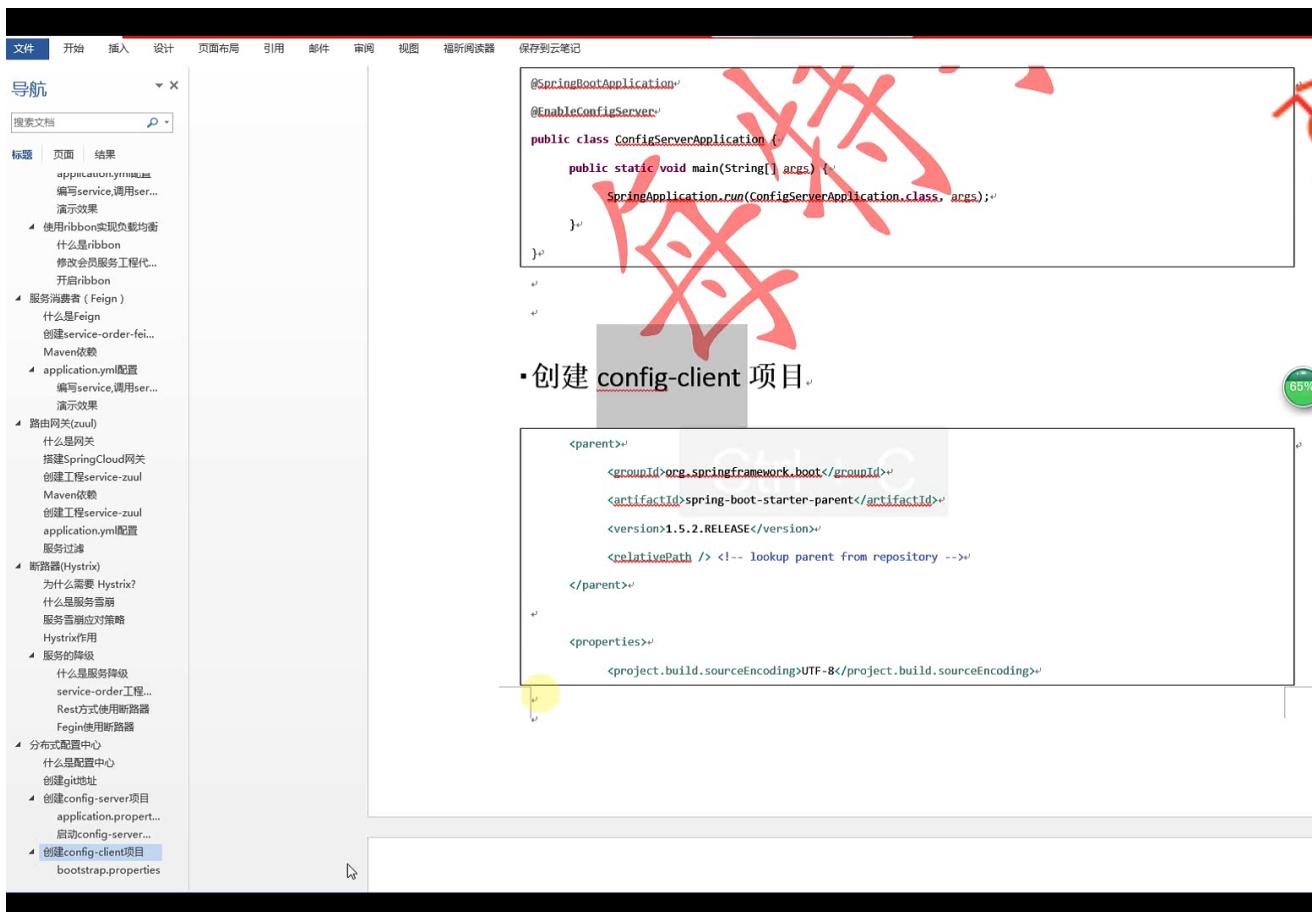
导航 搜索文档

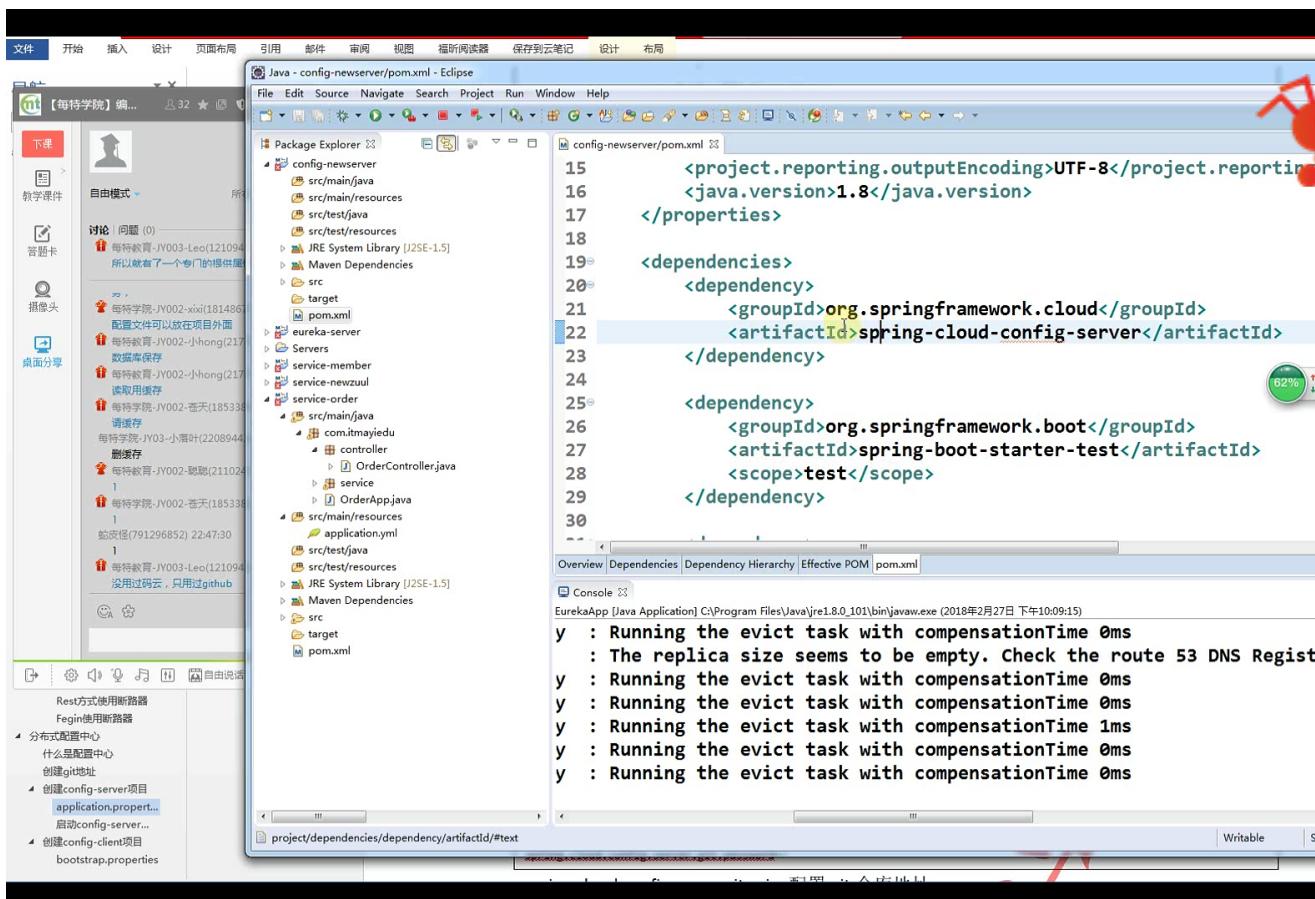
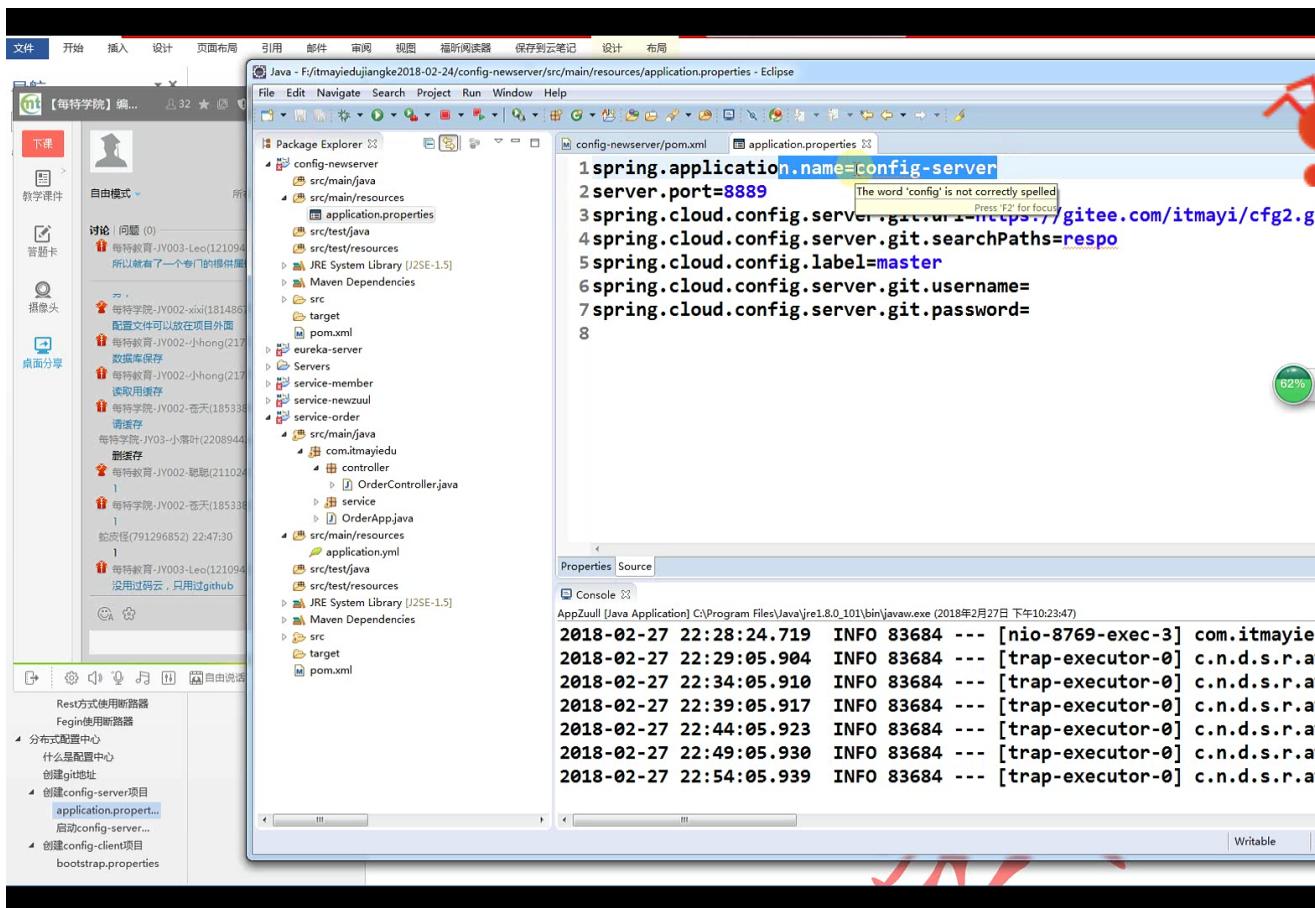
标题 页面 结果

- application.yml 编写service,调用ser... 演示效果
- 使用ribbon实现负载均衡 什么是ribbon 修改会员服务工程代... 开启ribbon
- 服务消费者(Feign) 什么是Feign 创建service-order-fei... Maven依赖
- application.yml配置 编写service,调用ser... 演示效果
- 路由网关(zuul) 什么是网关 基于SpringCloud网关 创建工程service-zuul Maven依赖 创建工程service-zuul application.yml配置 服务过滤
- 断路器(Hystrix) 为什么要Hystrix? 什么是服务雪崩 服务雪崩应对策略 Hystrix作用
- 服务的降级 什么是服务降级 service-order工程... Rest方式使用断路器 Feign使用断路器
- 分布式配置中心 什么是配置中心 创建git地址
- 创建config-server项目 application.propert... 启动config-server...
- 创建config-client项目 bootstrap.properties

```
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
<java.version>1.8</java.version>
<properties>
<dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-config</artifactId>
</dependency>
<dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-test</artifactId>
 <scope>test</scope>
</dependency>
</dependencies>
<dependencyManagement>
 <dependencies>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-dependencies</artifactId>
 <version>Dalston.RC1</version>

```





Spring Milestones

```
<name>Spring Milestones</name>
<url>https://repo.spring.io/milestone</url>
<snapshots>
 <enabled>false</enabled>
</snapshots>
```

(每特学院&蚂蚁课堂) 上海每特教育科技有限公司 [www.itmayiedu.com](http://www.itmayiedu.com) 余老师 QQ644064779

```
</repository>
</repositories>
```

## application.properties 配置文件

```
spring.application.name=config-server
server.port=8888
spring.cloud.config.server.git.uri=https://gitee.com/itmayi/cfg2.git
spring.cloud.config.server.git.searchPaths=respone
spring.cloud.config.label=master
spring.cloud.config.server.git.username=
spring.cloud.config.server.git.password=
```

Spring Boot Starter Test

```
<dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-test</artifactId>
 <scope>test</scope>
</dependency>
```

Spring Cloud Starter Eureka

```
<dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
```

Spring Cloud Dependencies

```
<dependencyManagement>
 <dependencies>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-dependencies</artifactId>
 <version>Camden.SR6</version>
 <type>pom</type>
 <scope>import</scope>
 </dependency>
 </dependencies>
</dependencyManagement>
```

Spring Boot Build Plugins

```
<build>
 <plugins>
 <plugin>
 <groupId>org.springframework.boot</groupId>
```

userName=yushengjun

## 创建 config-server 项目

```
<parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-parent</artifactId>
 <version>1.5.2.RELEASE</version>
 <relativePath /> <!-- lookup parent from repository -->
</parent>

<properties>
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
 <java.version>1.8</java.version>
</properties>

<dependencies>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-config-server</artifactId>
 </dependency>
</dependencies>
```

```
<dependencyManagement>
 <dependencies>
 <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-test</artifactId>
 <scope>test</scope>
 </dependency>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-eureka</artifactId>
 </dependency>
 </dependencies>
</dependencyManagement>

<build>
 <plugins>
 <plugin>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-maven-plugin</artifactId>
 </plugin>
 </plugins>
</build>
```

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记

导航

搜索文档

标题 | 页面 | 结果

application.yml配置  
编写service,调用ser...  
演示效果

使用ribbon实现负载均衡  
什么是ribbon  
修改会员服务工程代...  
开启ribbon

服务消费者（Feign）  
什么是Feign  
创建service-order-fei...  
Maven依赖

application.yml配置  
编写service,调用ser...  
演示效果

路由网关(zuul)  
什么是网关  
搭建SpringCloud网关  
创建工程service-zuul  
Maven依赖  
创建工程service-zuul  
application.yml配置  
服务过滤

断路器(Hystrix)  
为什么要使用Hystrix?  
什么是服务雪崩  
服务雪崩应对策略  
Hystrix作用

服务的降级  
什么是服务降级  
service-order工程...  
Rest方式使用断路器  
Feign使用断路器

分布式配置中心  
什么是配置中心  
创建git地址

创建config-server项目  
application.propert...  
启动config-server...

创建config-client项目  
bootstrap.properties

(每特学院&蚂蚁课堂)上海每特教育科技有限公司 [www.itmayiedu.com](http://www.itmayiedu.com) 余老师 QQ644064779

# 分布式配置中心

## 什么是配置中心

在分布式系统中，由于服务数量众多，为了方便服务配置文件统一管理，实时更新，所以需要分布式配置中心组件。在 Spring Cloud 中，有分布式配置中心组件 `spring cloud config`，它支持配置服务放在配置服务的内存中（即本地），也支持放在远程 Git 仓库中。在 `spring cloud config` 组件中，分两个角色，一是 `config server`，二是 `config client`。

## 创建 git 地址

1. 使用码云创建 git 地址 <https://gitee.com/itmav>
2. `config-client-dev.properties` ---dev 环境
3. 上传配置文件

```
userName=yushengjun
```

## 创建 config-server 项目

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记

导航

搜索文档

标题 | 页面 | 结果

application.yml配置  
编写service,调用ser...  
演示效果

使用ribbon实现负载均衡  
什么是ribbon  
修改会员服务工程代...  
开启ribbon

服务消费者（Feign）  
什么是Feign  
创建service-order-fei...  
Maven依赖

application.yml配置  
编写service,调用ser...  
演示效果

路由网关(zuul)  
什么是网关  
搭建SpringCloud网关  
创建工程service-zuul  
Maven依赖  
创建工程service-zuul  
application.yml配置  
服务过滤

断路器(Hystrix)  
为什么要使用Hystrix?  
什么是服务雪崩  
服务雪崩应对策略  
Hystrix作用

服务的降级  
什么是服务降级  
service-order工程...  
Rest方式使用断路器  
Feign使用断路器

分布式配置中心  
什么是配置中心  
创建git地址

创建config-server项目  
application.propert...  
启动config-server...

创建config-client项目  
bootstrap.properties

```
userName=yushengjun
```

## 创建 config-server 项目

```
<parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-parent</artifactId>
 <version>1.5.2.RELEASE</version>
 <relativePath /> <!-- lookup parent from repository -->
</parent>

<properties>
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
 <java.version>1.8</java.version>
</properties>

<dependencies>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-config-server</artifactId>

```

截屏显示了一个在线课堂界面，上方是PPT编辑器，下方是聊天记录。

**PPT编辑器内容：**

- 标题：远程地址
- 内容：
 

```
分布式 配置文件中心(git)
dev文件---userName=dev.itmayiedu
pre文件---userName=pre.itmayiedu
prd文件---userName=prd.itmayiedu
```
- 图示：一个中心方框“server-config 配置服务项目”通过箭头指向两个右侧方框“会员服务工程”。
- 下方文字说明：
  - 远程地址 git 主要存放配置文件信息
  - server-config 主要缓存配置文件信息，可以被其他服务进行调用读取。

**聊天记录（下方）：**

- 每特学院-JY002-陈独秀(1626661001) 22:39
 

```
1
 每特学院-JY002-陈独秀(1626661001) 22:39
 内存
 每特教育-JY002-小春(157905415) 22:39
 redis
 每特教育-JY003-Leo(1210941597) 22:39
 所以就有了一个专门的提供属性配置的服务
 每特学院-JY002-xixi(1814867731) 22:39
 配置文件可以放在项目外面
 每特教育-JY002-j-hong(2177560013) 2
 数据库保存
 每特教育-JY002-j-hong(2177560013) 2
 使用便捷
 每特学院-JY03-苍天(1853385474) 22:41
 请缓存
 每特学院-JY03-小落叶(2208944241) 22:41
 缓存
```

截屏显示了一个在线课堂界面，上方是Eclipse IDE，下方是多个窗口。

**Eclipse IDE窗口：**

- 左侧包资源管理器（Package Explorer）显示了项目结构，包括 `eureka-server`、`service-member` 和 `service-newzuul`。
- 右侧代码编辑器显示了 `MyFilter.java` 的代码，内容如下：

```
public Object run() {
 RequestContext ctx = RequestContext.get();
 HttpServletRequest request = ctx.getRequest();
 Log.info(String.format("%s >> %s", request.getMethod(),
 request.getRequestURI()));
 Object accessToken = request.getParameter("token");
 String refer=request.getHeader("Referer");
 if (accessToken != null) {
 return null;
 }
 Log.warn("token is empty");
 ctx.setSendZuulResponse(false);
 ctx.setResponseStatus(401);
 try {
 ctx.getResponse().getWriter().write("you are not login");
 } catch (Exception e) {
 }
}
```

**下方窗口：**

- 左侧窗口显示了命令行输出，包含日志信息：

```
y : Running the evict task with compensationTime
with compensationTime
to be empty. Check the
with compensationTime
```

- 右侧窗口显示了几个不同的UML类图或系统架构图。

Screenshot of a desktop environment showing multiple windows:

- Eclipse IDE (Left Window):** Shows the Java - service-newzuul project structure with files like application.yml, OrderController.java, MemberController.java, AppZuul.java, and MyFilter.java.
- Diagram (Center Window):** A hand-drawn diagram illustrating Zuul's role as an API gateway. It shows requests from clients (e.g., www.itmayiedu.com/membre, www.itmayiedu.com/order) being intercepted by the 'zuul' gateway. The gateway then forwards the requests to downstream services (e.g., member.itmayiedu.co, order.itmayiedu.co).
- Notepad (Right Window):** A text editor with a toolbar showing Chinese characters and colors. The text discusses Zuul's function as an API gateway, its interception of requests, and how it forwards them to real servers based on project names.
- File Explorer (Bottom Left):** Shows the file system with bootstrap.properties and OrderApp.java.

Screenshot of a desktop environment showing multiple windows:

- Eclipse IDE (Left Window):** Shows the Java - service-newzuul project structure with files like application.yml, OrderController.java, MemberController.java, AppZuul.java, and MyFilter.java.
- Code Editor (Main Window):** Displays the MyFilter.java code. The code implements a Zuul filter to check for an access token in the request parameters. If the token is present and valid, it continues the request; if not, it logs a warning and sends a response indicating the token is empty.
- File Explorer (Bottom Left):** Shows the file system with bootstrap.properties.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Structure:** The Package Explorer view shows a project structure with several modules: eureka-server, service-member, service-newzuul, service-order, and service-app.
- Code Editor:** The main editor displays the `MyFilter.java` file, which contains Java code for a Zuul filter extending `ZuulFilter`.
- Console Output:** The Console tab shows the application's startup logs:

```
y : Registered instance SERVICE-ZUUL/2012-20160701XG:service-zuul:8769 with status UP
y : Running the evict task with compensationTime 0ms
y : Running the evict task with compensationTime 0ms
: The replica size seems to be empty. Check the route 53 DNS Registry
y : Running the evict task with compensationTime 0ms
y : Running the evict task with compensationTime 1ms
y : Running the evict task with compensationTime 0ms
```
- Bottom Status Bar:** Shows the current zoom level at 60%.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Structure:** The Package Explorer view shows a project structure with several modules: eureka-server, service-member, service-newzuul, service-order, and service-app.
- Code Editor:** The main editor displays the `AppZuull.java` file, which contains Spring Boot and Netflix Zuul configuration code.
- Console Output:** The Console tab shows the application's startup logs:

```
EurekaApp [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (2018年2月27日 下午10:09:15)
```
- Bottom Status Bar:** Shows the current zoom level at 69%.

Java - service-member/src/main/resources/application.yml - Eclipse

```
1 eureka:
2 client:
3 serviceUrl:
4 defaultZone: http://localhost:8888/eureka/
5 server:
6 port: 8763
7 spring:
8 application:
9 name: service-member
10
```

Eureka 127.0.0.1:8762/getMe 127.0.0.1:8763/getMe 127.0.0.1:8764/getOrd 127.0.0.1:8769/api-mem 127.0.0.1:8763/getMe

# Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Tue Feb 27 22:15:35 CST 2018  
There was an unexpected error (type=Not Found, status=404).  
No message available

Java - service-newzuul/src/main/resources/application.yml - Eclipse

```
1 eureka:
2 client:
3 serviceUrl:
4 defaultZone: http://localhost:8888/eureka/
5 server:
6 port: 8769
7 spring:
8 application:
9 name: service-zuul
10 zuul:
11 routes:
12 api-a:
13 path: /api-member/**
14 service-id: service-member
15 api-b:
16 path: /api-order/**
17 service-id: service-order
```

Console

```
InstanceRegistry : Running the evict task with compensator
```

转发到 http://127.0.0.1:8762/getMemberAll

(每特学院&蚂蚁课堂) 上海每特教育科技有限公司 [www.itmayiedu.com](http://www.itmayiedu.com) 余老师 QQ644064779

```
<dependencies>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-eureka</artifactId>
 </dependency>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-zuul</artifactId>
 </dependency>
 <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-web</artifactId>
 </dependency>
 <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-test</artifactId>
 <scope>test</scope>
 </dependency>
</dependencies>
<dependencyManagement>
 <dependencies>
 <dependency>
```

搭建 SpringCloud 网关

Zuul 的主要功能是路由转发和过滤器。路由功能是微服务的一部分，比如 /api/user 转发到到 user 服务，/api/shop 转发到到 shop 服务。zuul 默认和 Ribbon 结合实现了负载均衡的功能，类似于 nginx 转发。

搭建 SpringCloud 网关

创建工程 service-zuul

Maven 依赖

创建工程 service-zuul

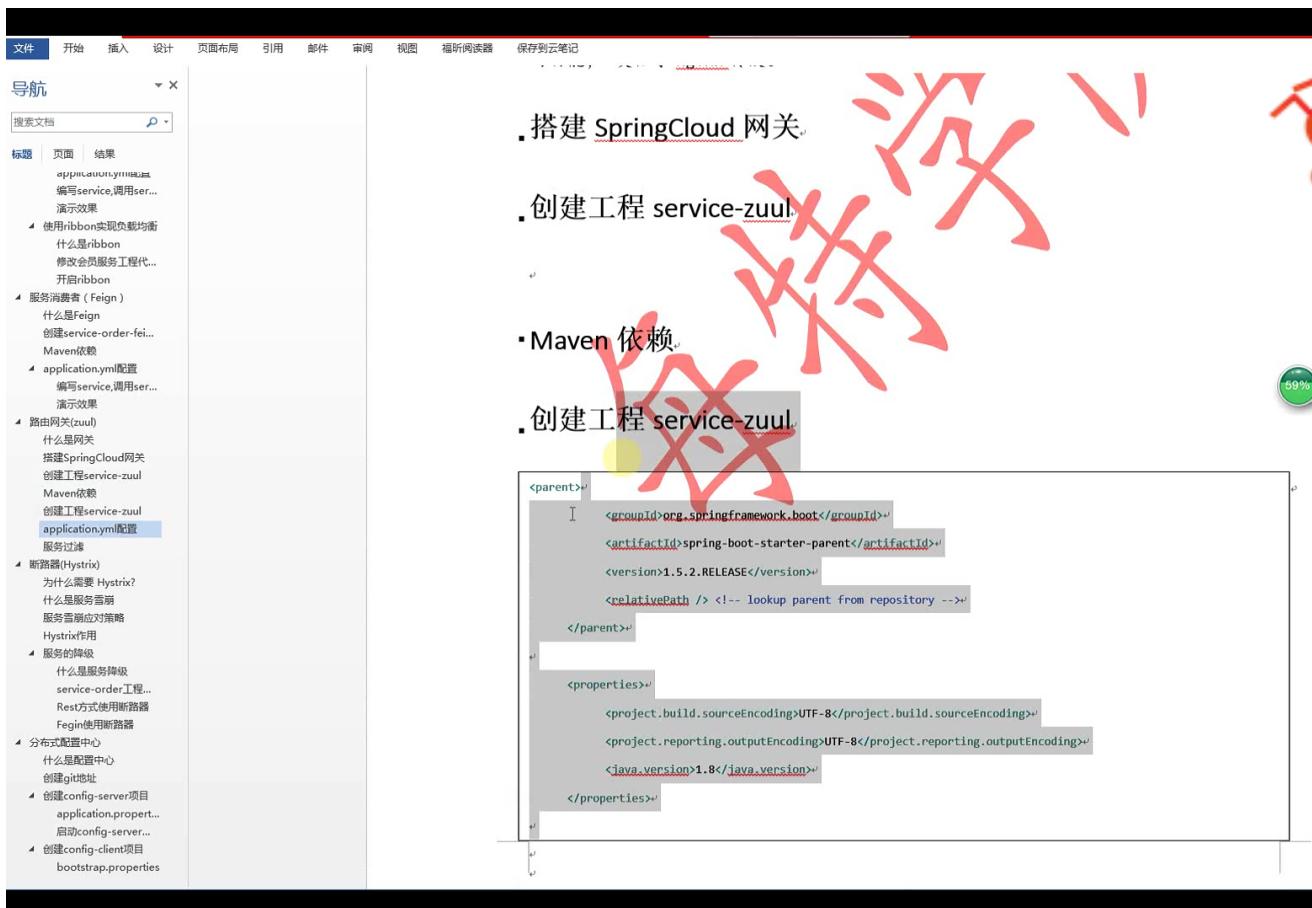
```
<parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-parent</artifactId>
 <version>1.5.2.RELEASE</version>
 <relativePath /> <!-- lookup parent from repository -->
</parent>
<properties>
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
```

搭建 SpringCloud 网关

· 创建工程 service-zuul

· Maven 依赖

· 创建工程 service-zuul

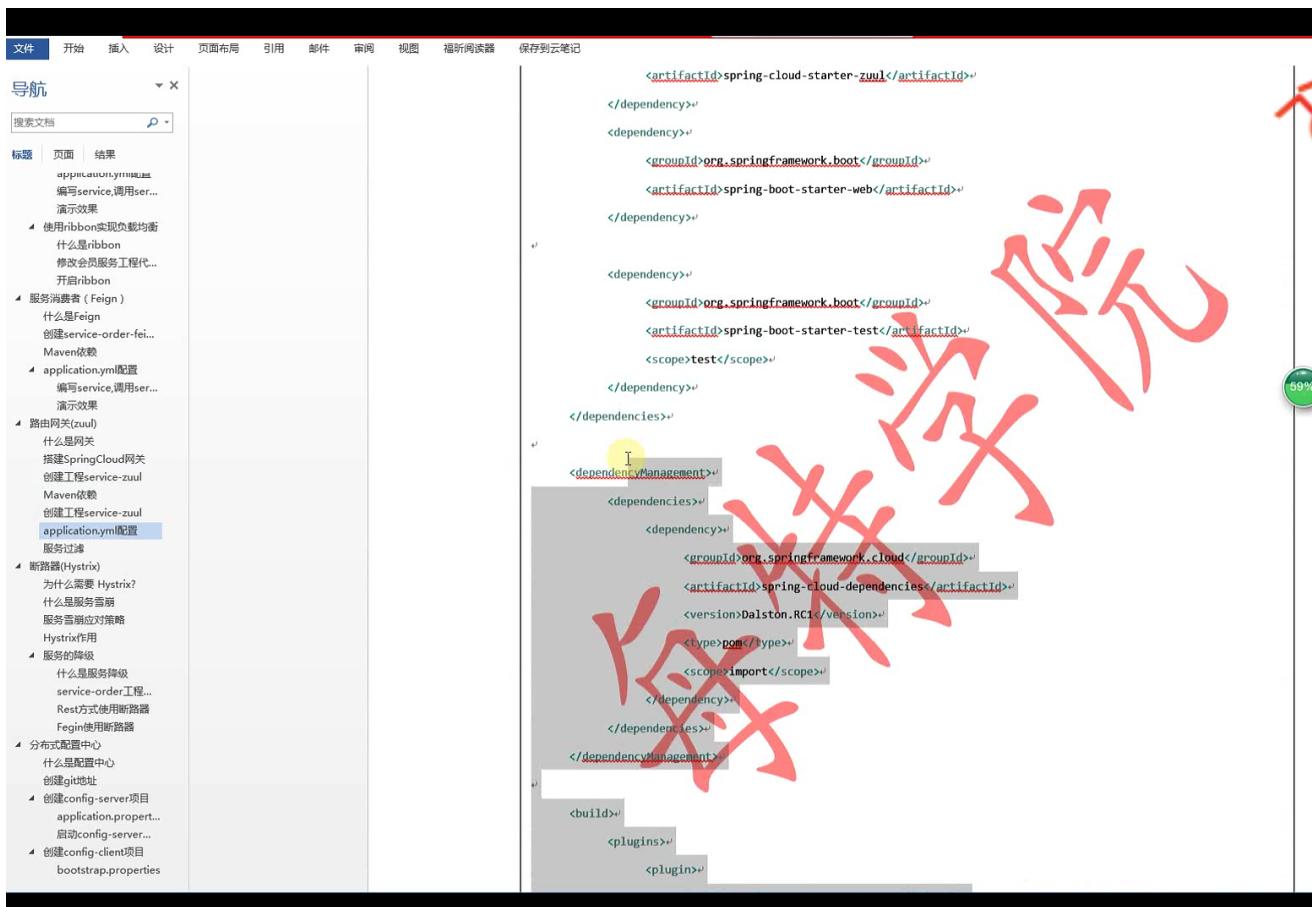


```
<parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-parent</artifactId>
 <version>1.5.2.RELEASE</version>
 <relativePath /> <!-- lookup parent from repository -->
</parent>

<properties>
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
 <java.version>1.8</java.version>
</properties>
```

· spring-cloud-starter-zuul

· dependencyManagement



```
<dependency>
 <artifactId>spring-cloud-starter-zuul</artifactId>
</dependency>
<dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-test</artifactId>
 <scope>test</scope>
</dependency>
</dependencies>
<dependencyManagement>
 <dependencies>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-dependencies</artifactId>
 <version>Dalston.RC1</version>
 <type>pom</type>
 <scope>import</scope>
 </dependency>
 </dependencies>
</dependencyManagement>
<build>
 <plugins>
 <plugin>
```

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记

导航

搜索文档

标题 页面 结果

application.yml  
编写service,调用ser...  
演示效果

使用ribbon实现负载均衡  
什么是ribbon  
修改会员服务工程代...  
开启ribbon

服务消费者(Feign)  
什么是Feign  
创建service-order-fei...  
Maven依赖

application.yml配置  
编写service,调用ser...  
演示效果

路由网关(zuul)  
什么是网关  
搭建SpringCloud网关  
创建工程service-zuul  
Maven依赖  
application.yml配置  
服务过滤

断路器(Hystrix)  
为什么需要Hystrix?  
什么是服务雪崩  
服务雪崩应对策略  
Hystrix作用

服务的降级  
什么是服务降级  
service-order工程...  
Rest方式使用断路器  
Feign使用断路器

分布式配置中心  
什么是配置中心  
创建git地址

创建config-server项目  
application.propert...  
启动config-server...

创建config-client项目  
bootstrap.properties

<groupId>org.springframework.boot</groupId>  
<artifactId>spring-boot-maven-plugin</artifactId>  
</plugin>  
</plugins>  
</build>  
</>  
<repositories>  
<repository>

(每节课学院&蚂蚁课堂) 上海每节课教育科技有限公司 www.itmayiedu.com 余老师 00644064779

59%

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记

导航

搜索文档

标题 页面 结果

application.yml  
编写service,调用ser...  
演示效果

使用ribbon实现负载均衡  
什么是ribbon  
修改会员服务工程代...  
开启ribbon

服务消费者(Feign)  
什么是Feign  
创建service-order-fei...  
Maven依赖

application.yml配置  
编写service,调用ser...  
演示效果

路由网关(zuul)  
什么是网关  
搭建SpringCloud网关  
创建工程service-zuul  
Maven依赖  
创建工程service-zuul  
application.yml配置  
服务过滤

断路器(Hystrix)  
为什么需要Hystrix?  
什么是服务雪崩  
服务雪崩应对策略  
Hystrix作用

服务的降级  
什么是服务降级  
service-order工程...  
Rest方式使用断路器  
Feign使用断路器

分布式配置中心  
什么是配置中心  
创建git地址

创建config-server项目  
application.propert...  
启动config-server...

创建config-client项目  
bootstrap.properties

## 路由网关(zuul)

什么是网关

Zuul的主要功能是路由转发和过滤器。路由功能是微服务的一部分，比如 /api/user 转发到到\_user 服务，/api/shop 转发到\_shop 服务。zuul默认和 Ribbon 结合实现了负载均衡的功能，类似于 nginx 转发。

搭建 SpringCloud 网关

创建工程 service-zuul

Maven 依赖

创建工程 service-zuul

```
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>1.5.2.RELEASE</version>
```

幻灯片编辑器界面，显示了每特学院的课程讨论和一个关于接口网关的示意图。

**讨论 | 问题 (0)**

- 每特教育-JY002-豌豆(2110247944) 21:5 跨域
- 每特教育-JY002-归零者(2109723617) 21:5 Nginx
- 每特学院-JY03-小雷(2208944241) 21:55 nginx
- 每特学院-JY03-Don't Shy(1713847856) 21: 反向代理
- javased(1228055872) 21:55:10 中转站
- 每特教育-JY002-春天(1629325193) 21:5 location
- 每特学院-JY03-Don't Shy(1713847856) 21: 会员 拉人
- 每特教育-JY002-Hugo(990992343) 21:1
- 每特学院JY\_002-小白(2183141496) 21:5 1

**图示说明：**

该图展示了接口网关（zuul）的作用。外部请求通过H5端调用（如www.itmayiedu.com/membre和www.itmayiedu.com/order）到达接口网关。接口网关拦截所有请求，并将它们转发到实际的服务（会员服务和订单服务）。同时，图中还显示了通过AJAX直接访问会员服务（member.itmayiedu.com）和订单服务（order.itmayiedu.com）的情况，这两个路径被标记为错误（X）。

Eclipse IDE界面，显示了Java项目的代码编辑器和控制台。

**Pom.xml文件内容：**

```

<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
<dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-ribbon</artifactId>
</dependency>
<dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-test</artifactId>
 <scope>test</scope>

```

**控制台输出：**

```

Started on port(s): 8763 (http)
port to 8763
MemberApp in 12.191 seconds (JVM running for 13.335)
yClient_SERVICE-MEMBER/2012-20160701KG:service-member:8763 - registration status:
zing Spring FrameworkServlet 'dispatcherServlet'
kServlet 'dispatcherServlet': initialization started
kServlet 'dispatcherServlet': initialization completed in 52 ms

```

The screenshot shows the Eclipse IDE interface. On the left is the Package Explorer view displaying project structures for 'eureka-server' and 'service-member'. The right side shows the MemberController.java file and a terminal window.

```

1 package com.itmayiedu.controller;
2
3 import java.util.ArrayList;
4
5 @RestController
6 public class MemberController {
7 @Value("${server.port}")
8 private String serverPort;
9
10 @RequestMapping("/getMemberAll")
11 public List<String> getMemberAll() {
12 List<String> listUser = new ArrayList<String>();
13 listUser.add("zhangsan");
14 listUser.add("lisi");
15 listUser.add("wangwu");
16 listUser.add("serverPort:" + serverPort);
17 return listUser;
18 }
19}

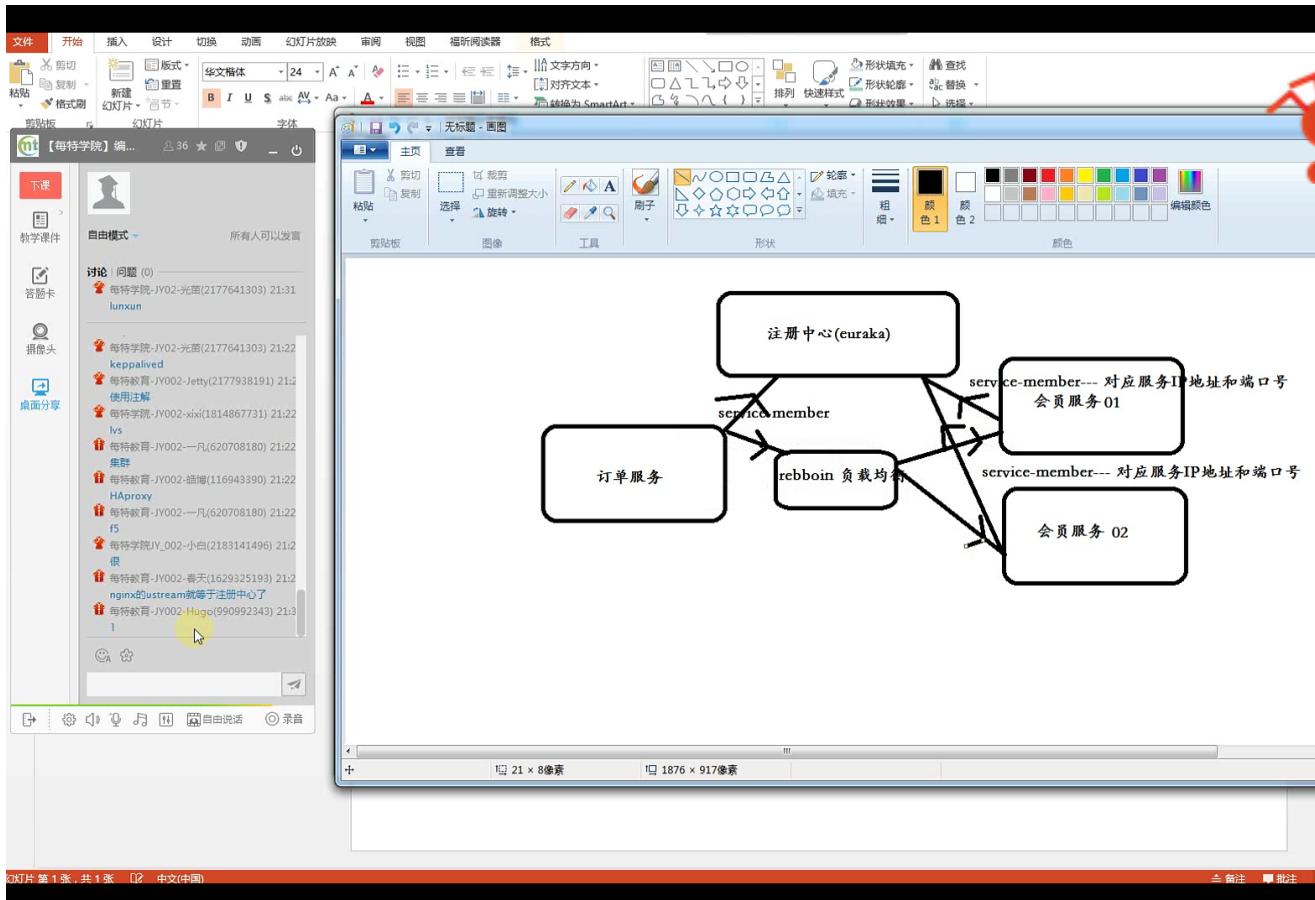
```

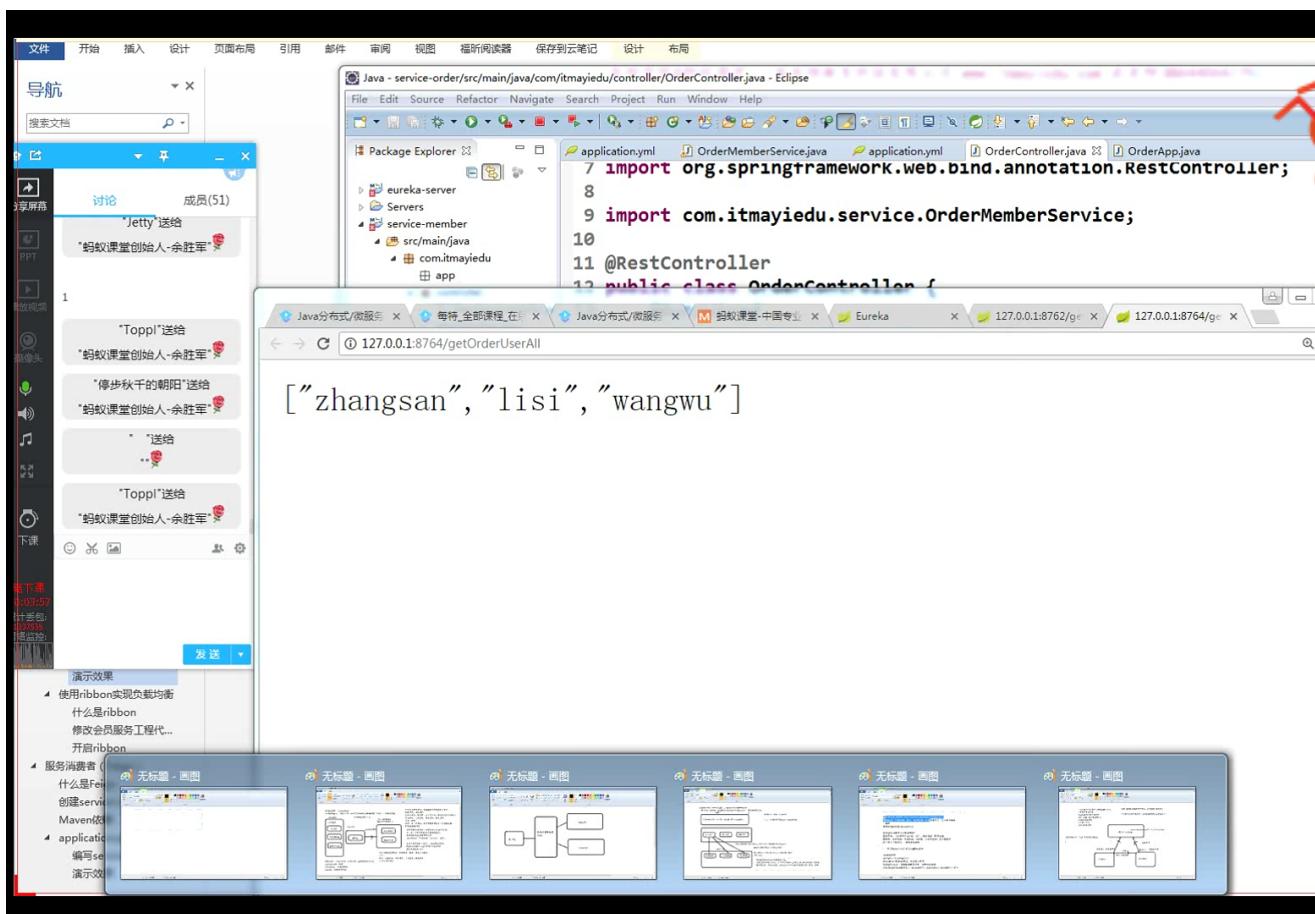
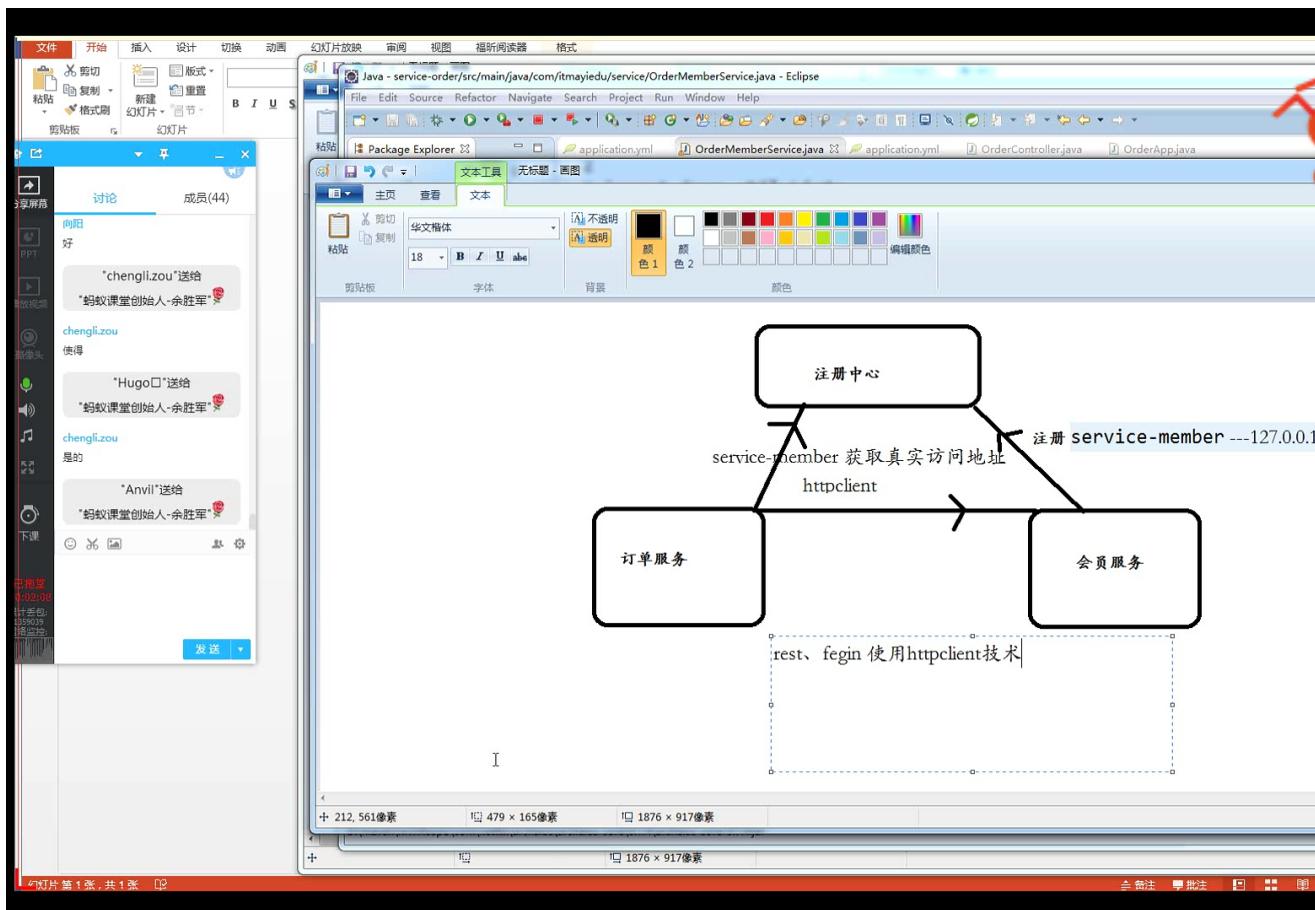
Console output:

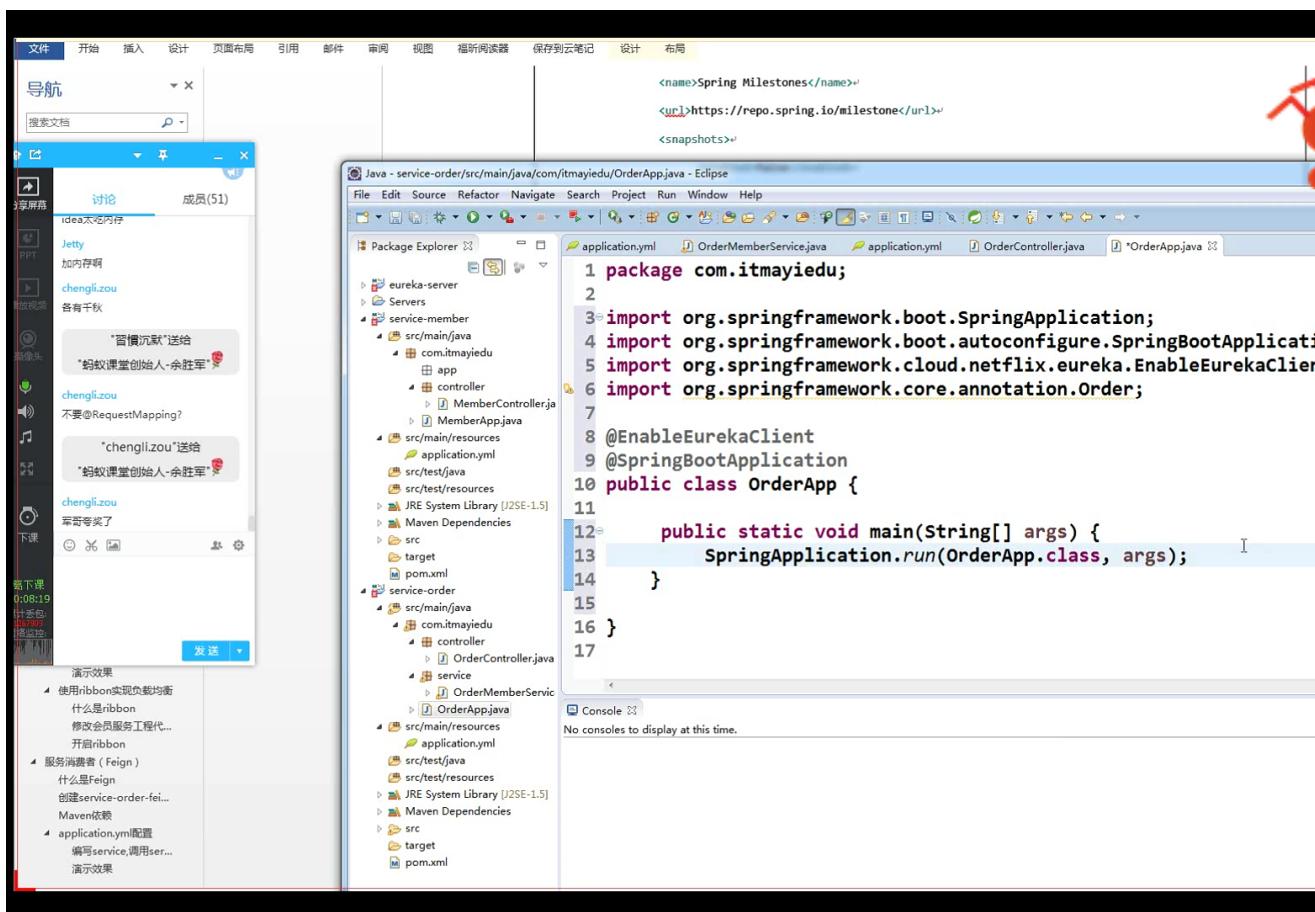
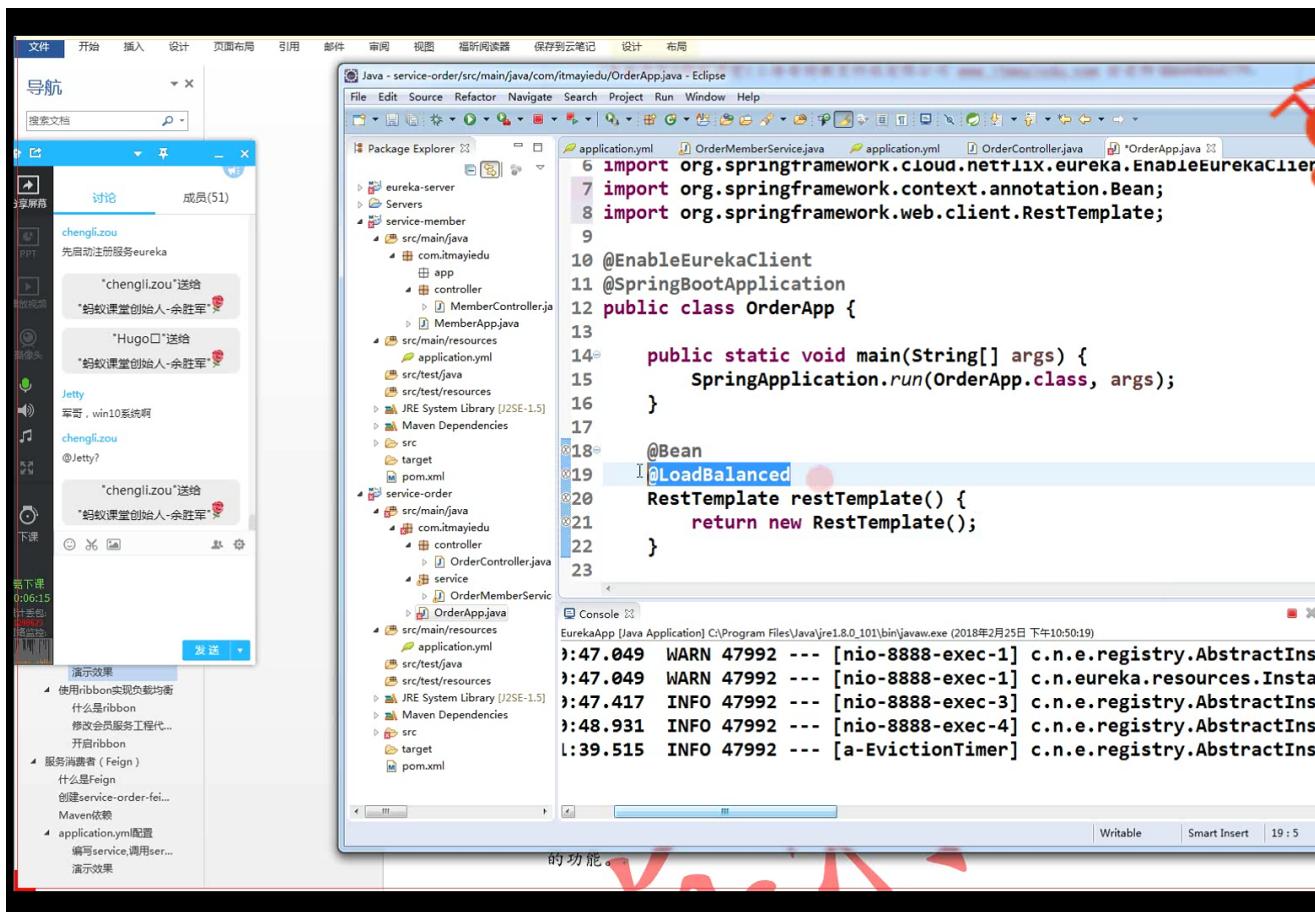
```

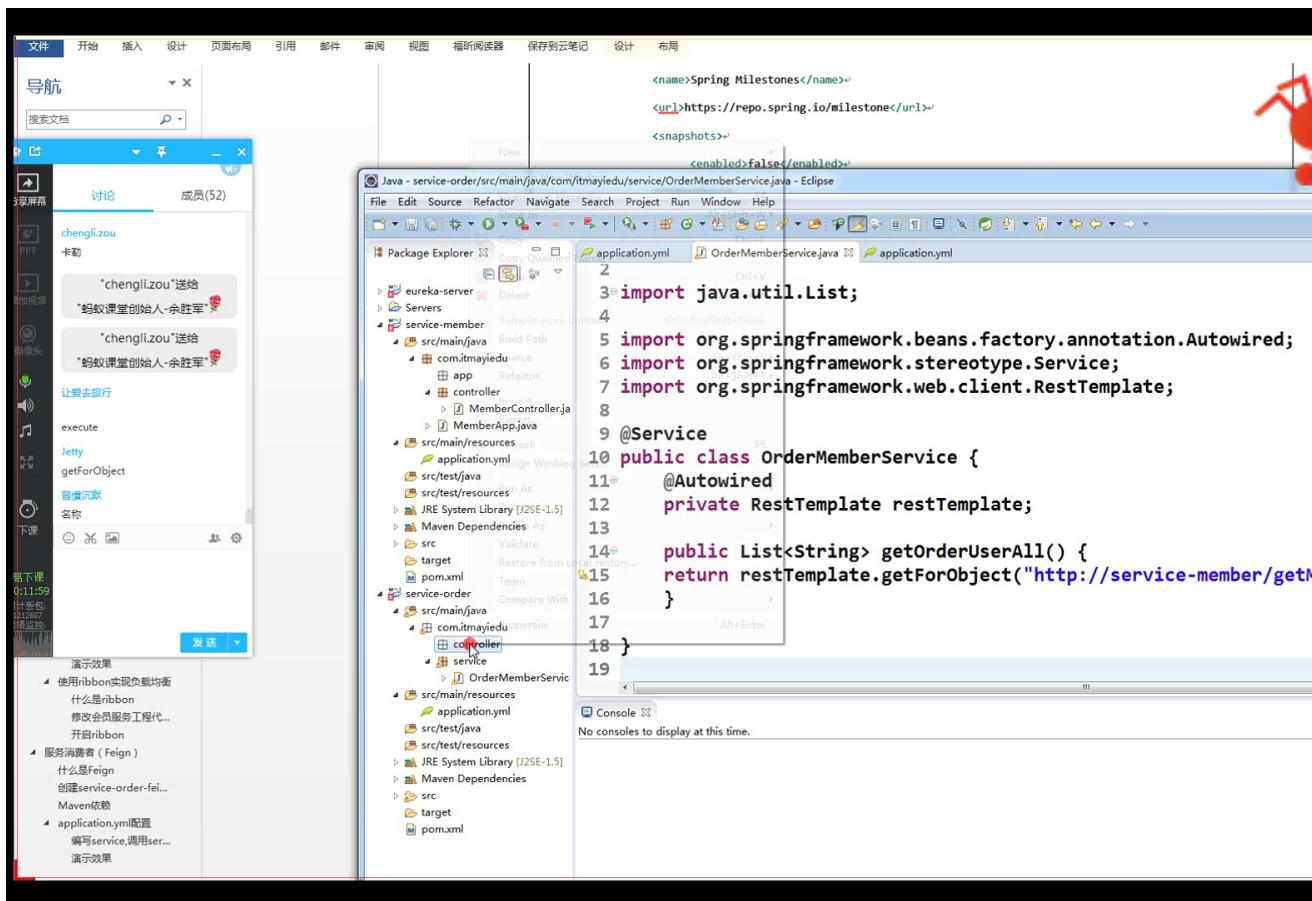
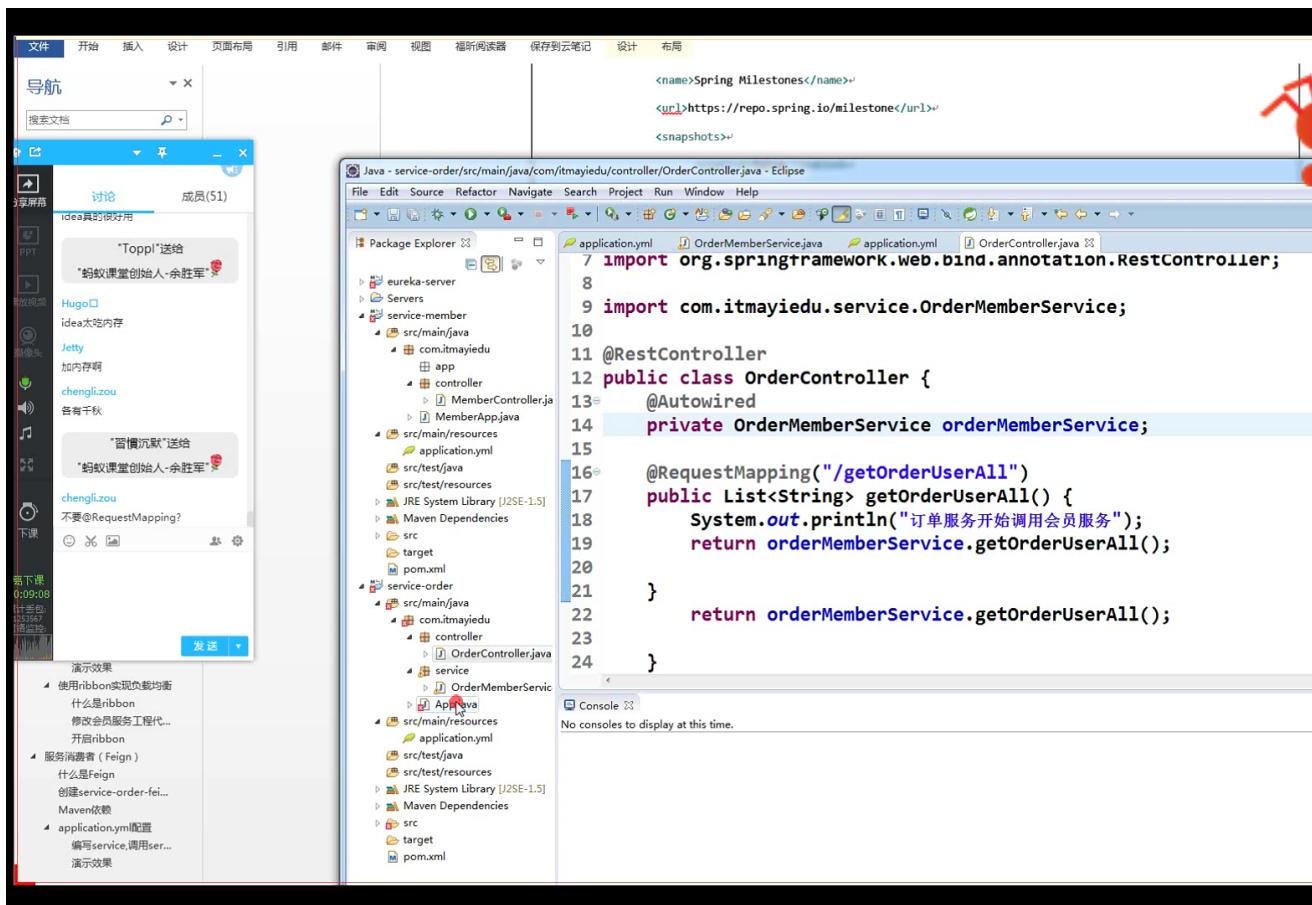
MemberApp [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (2018年2月27日下午9:38:32)
started on port(s): 8763 (http)
port to 8763
MemberApp in 12.191 seconds (JVM running for 13.335)
yClient_SERVICE-MEMBER/2012-20160701XG:service-member:8763 - registration statu
zing Spring FrameworkServlet 'dispatcherServlet'
kServlet 'dispatcherServlet': initialization started
kServlet 'dispatcherServlet': initialization completed in 52 ms

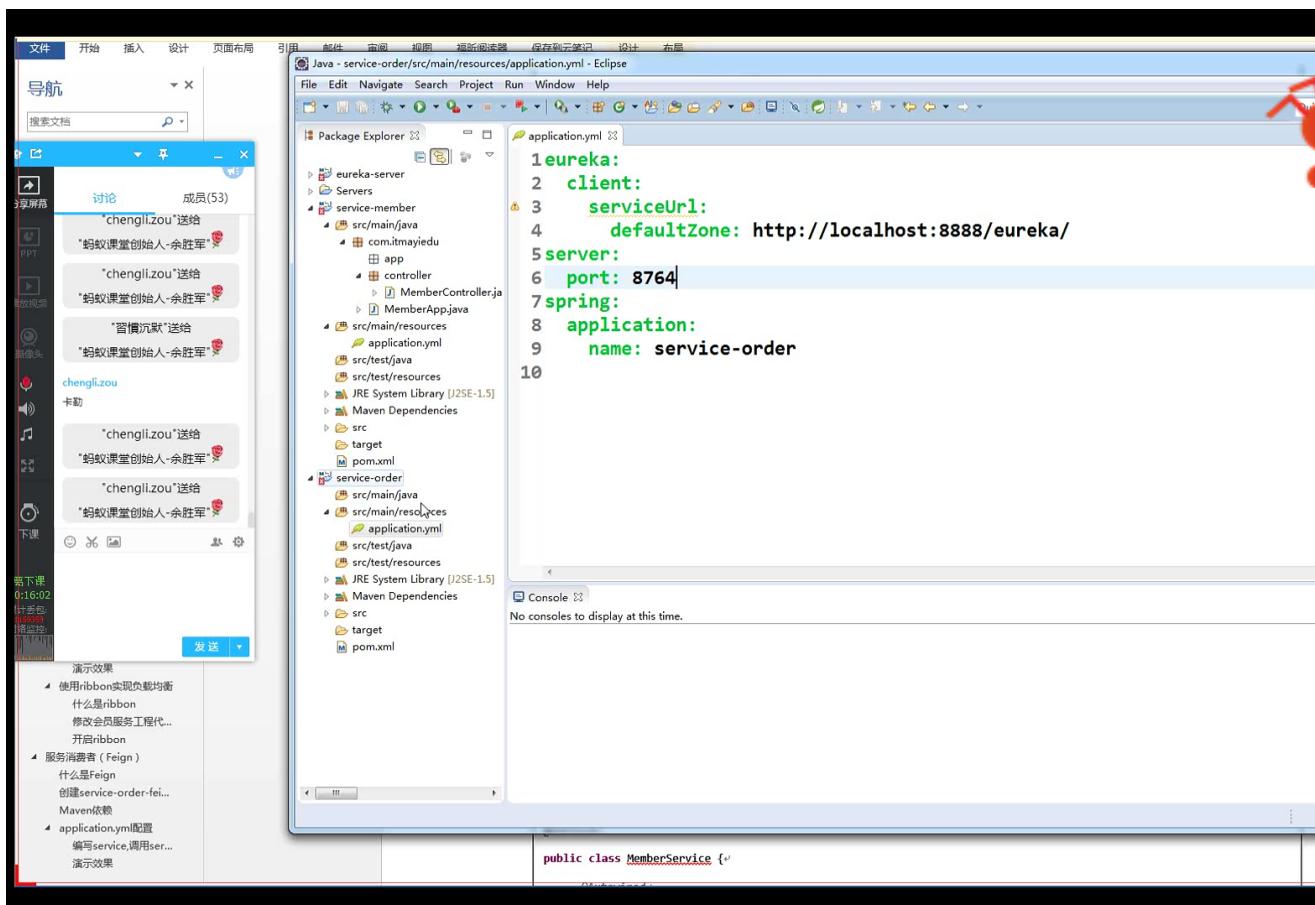
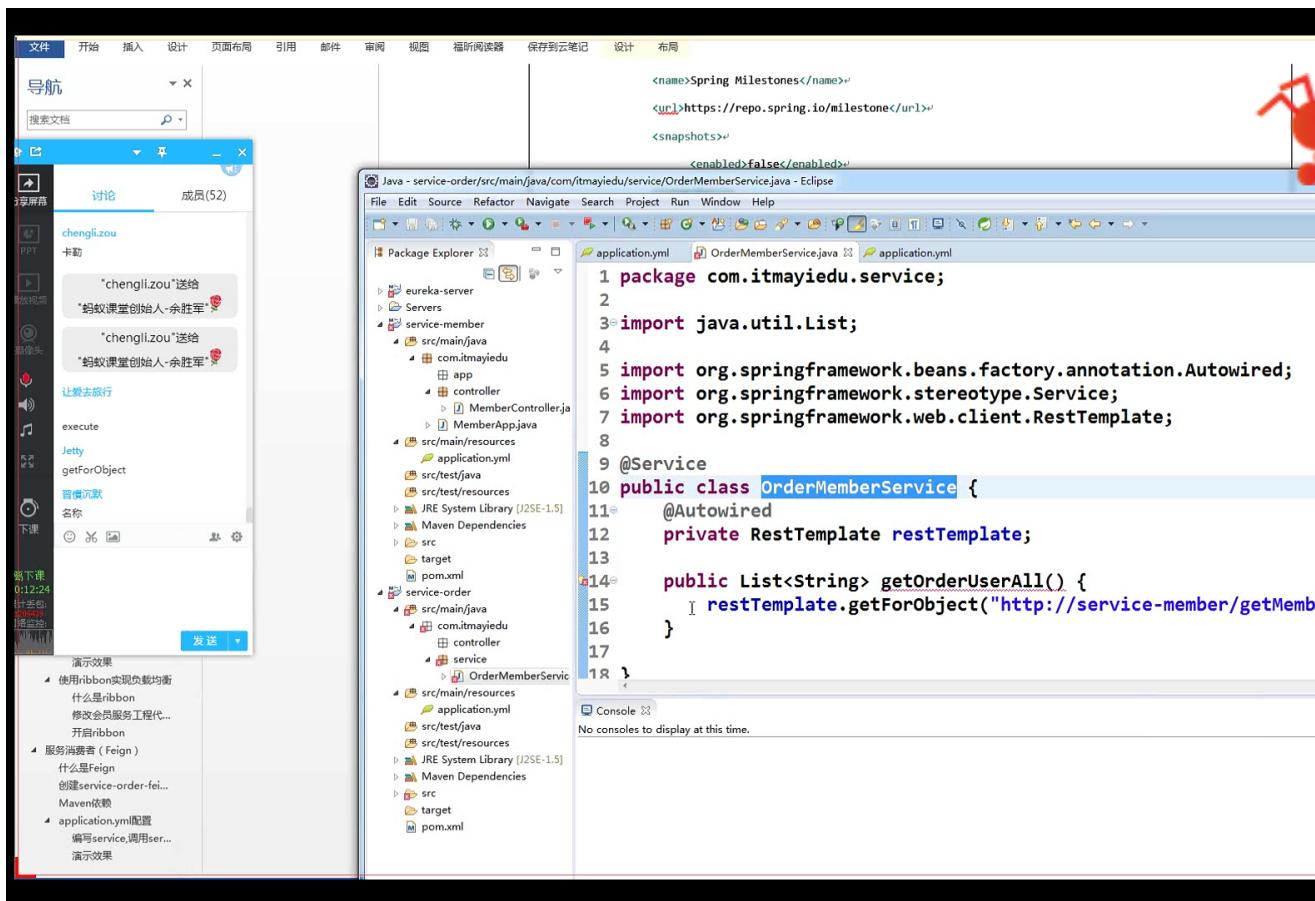
```











每特学院&蚂蚁课堂 上海每特教育科技有限公司 [www.itmayiedu.com](http://www.itmayiedu.com) 余老师 QQ644064779

```
<plugin>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-maven-plugin</artifactId>
</plugin>

</plugins>
</build>

<repositories>
 <repository>
 <id>spring-milestones</id>
 <name>Spring Milestones</name>
 <url>https://repo.spring.io/milestone</url>
 <snapshots>
 <enabled>false</enabled>
 </snapshots>
 </repository>
</repositories>
```

每特学院&蚂蚁课堂 上海每特教育科技有限公司 [www.itmayiedu.com](http://www.itmayiedu.com) 余老师 QQ644064779

```
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
<java.version>1.8</java.version>

<properties>

<dependencies>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-eureka</artifactId>
 </dependency>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-ribbon</artifactId>
 </dependency>
 <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-web</artifactId>
 </dependency>
 <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-test</artifactId>
 <scope>test</scope>
 </dependency>
</dependencies>

<dependencyManagement>
```

你会反观一个服务已经注册在服务中了，服务名为 SERVICE-HI，端口为 8062。  
这时打开 <http://127.0.0.1:8762/getUserList>，你会在浏览器上看到：  
[ "zhangsan", "lisi", "yusheng jun" ]

**Maven 依赖**

```
<parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-parent</artifactId>
 <version>1.5.2.RELEASE</version>
 <relativePath /> <!-- lookup parent from repository -->
</parent>

<properties>
```

讨论 成员(54)

\*chengli.zou\*送给 \*蚂蚁课堂创始人-余胜军\*

\*chengli.zou\*送给 \*蚂蚁课堂创始人-余胜军\*

\*習慣沉默\*送给 \*蚂蚁课堂创始人-余胜军\*

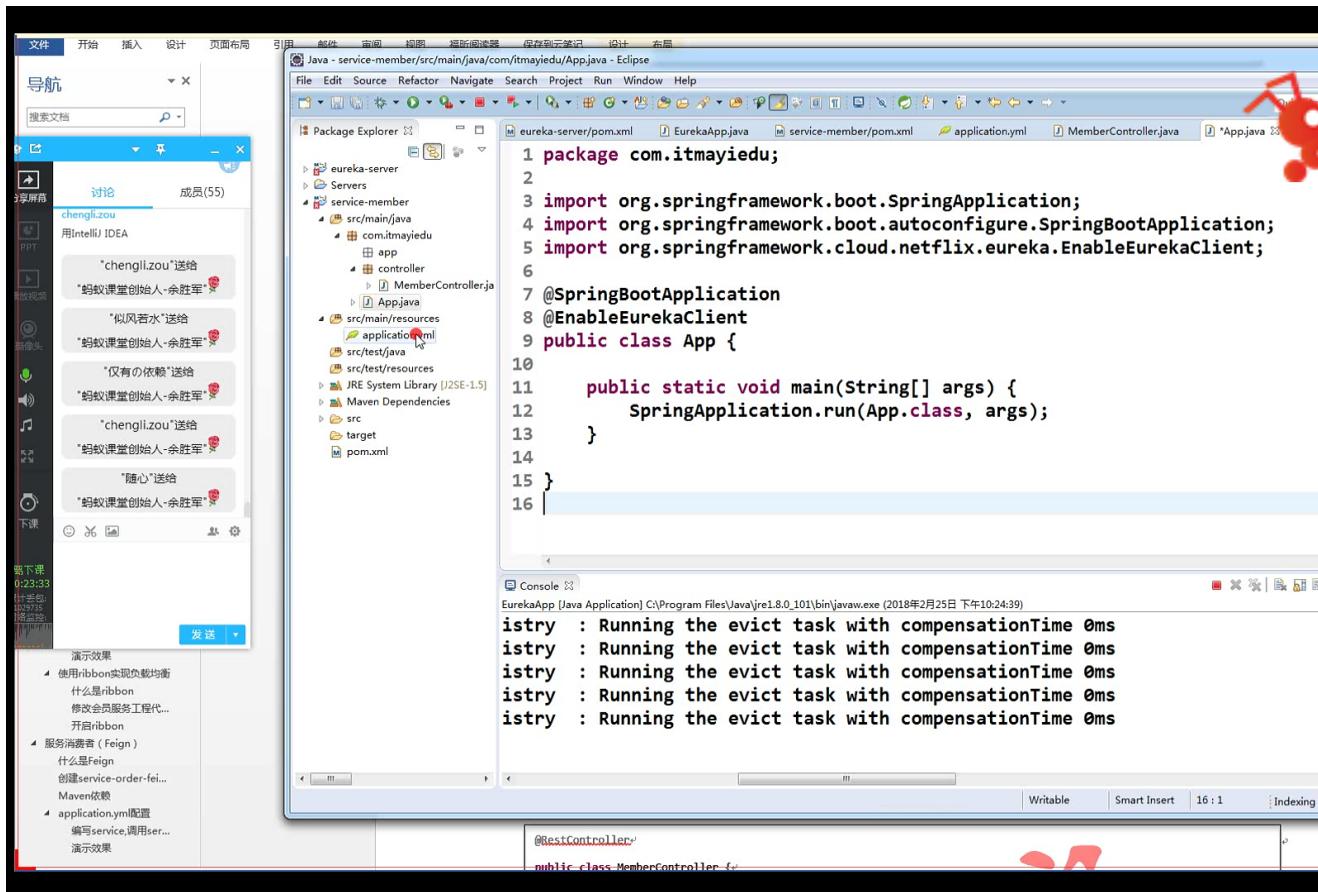
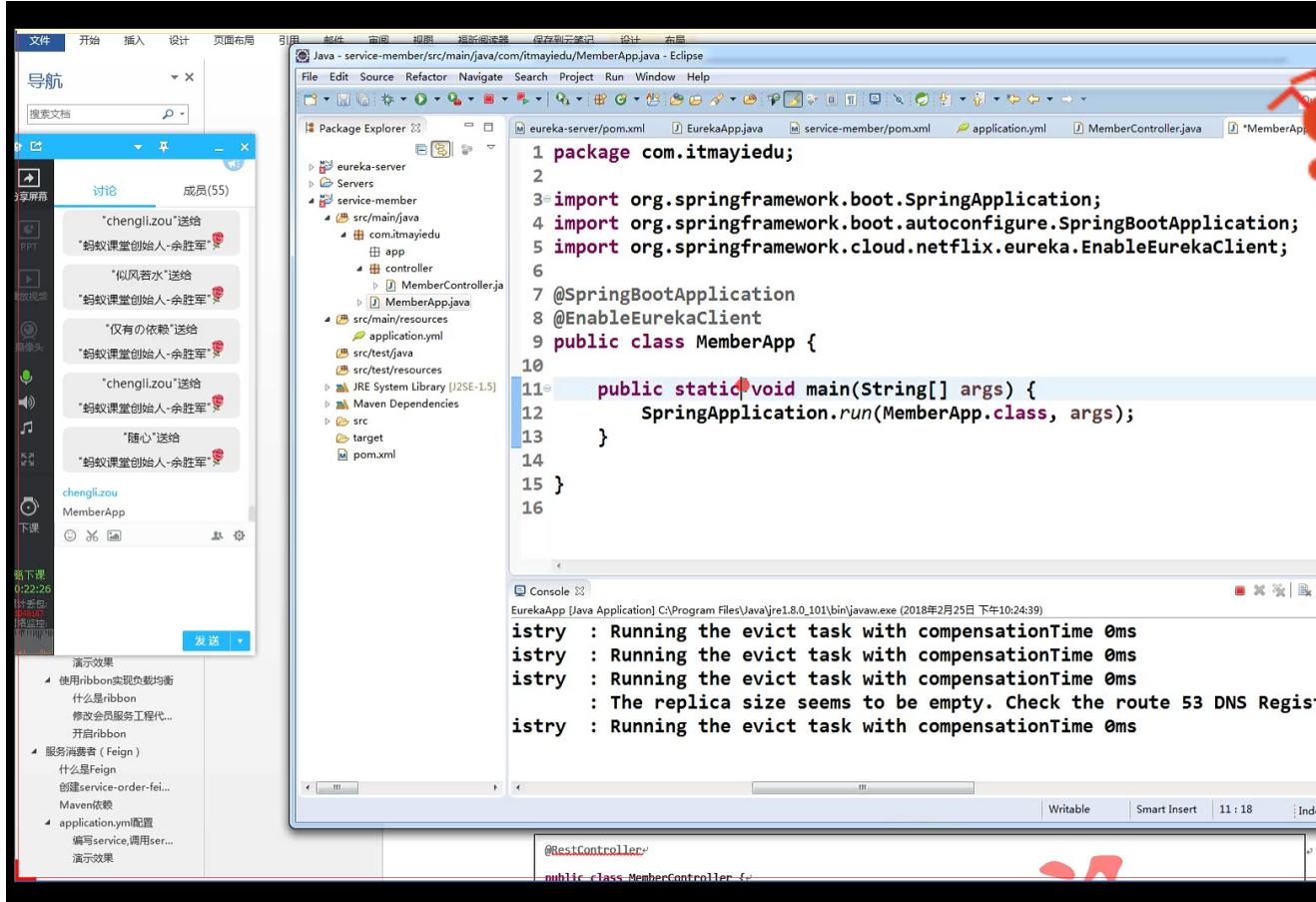
chengli.zou  
卡勤

\*chengli.zou\*送给 \*蚂蚁课堂创始人-余胜军\*

\*chengli.zou\*送给 \*蚂蚁课堂创始人-余胜军\*

案例: 会员服务提供用户信息、订单服务 查询订单  
订单服务需要查询用户，订单服务调用会员服务接口

1.SpringCloud注册中心环境搭建 eureka  
2.服务注册与发现  
3.SpringCloud客户端调用  
rest、feign 客户端调用工具  
ribbon 负载均衡  
zuul接口网关  
nreka服务注册



The screenshot shows the Eclipse IDE interface with the following details:

- File Explorer:** Shows the project structure for "eureka-server" and "service-member". The "MemberController.java" file is selected.
- Code Editor:** Displays the following Java code:

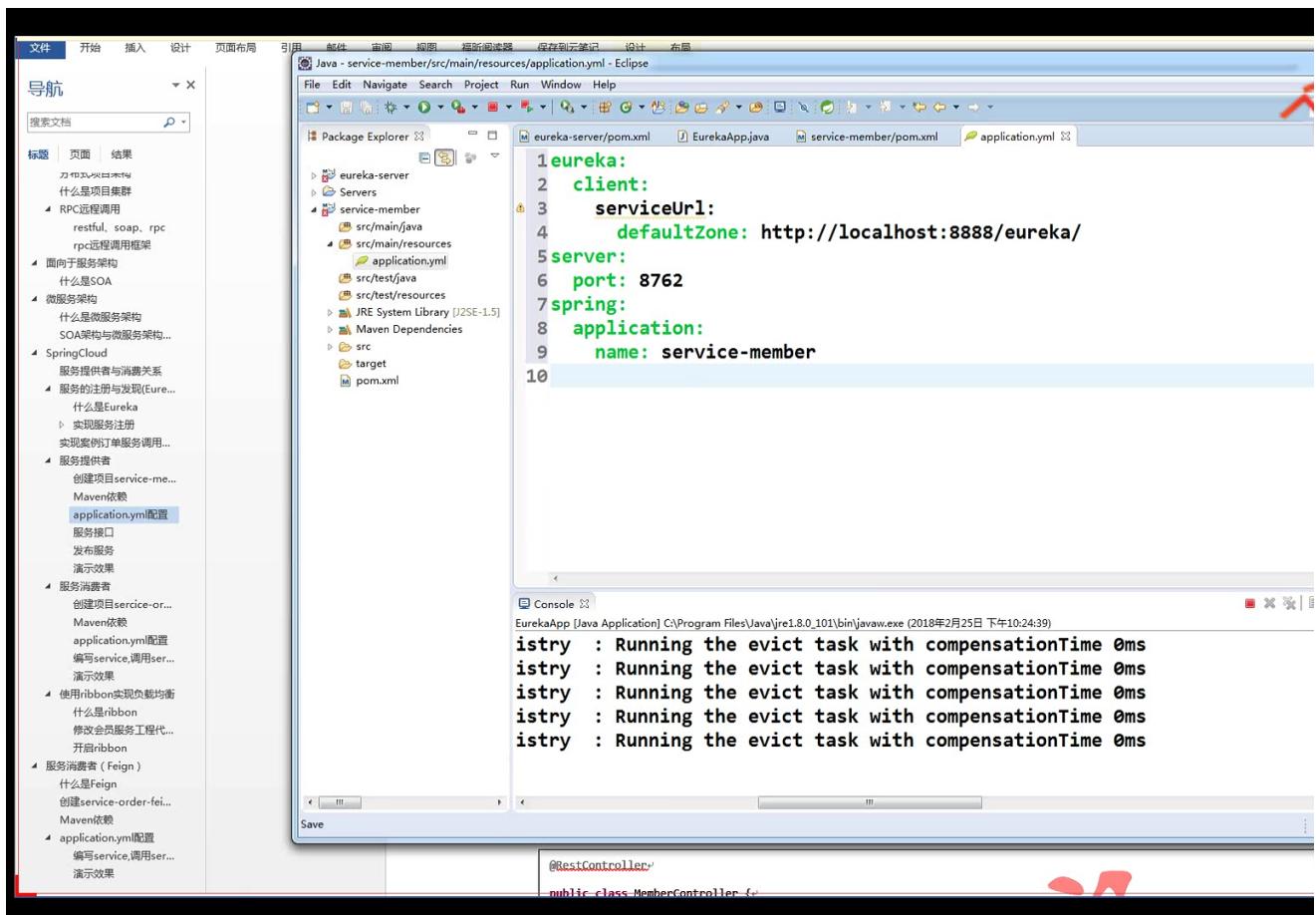
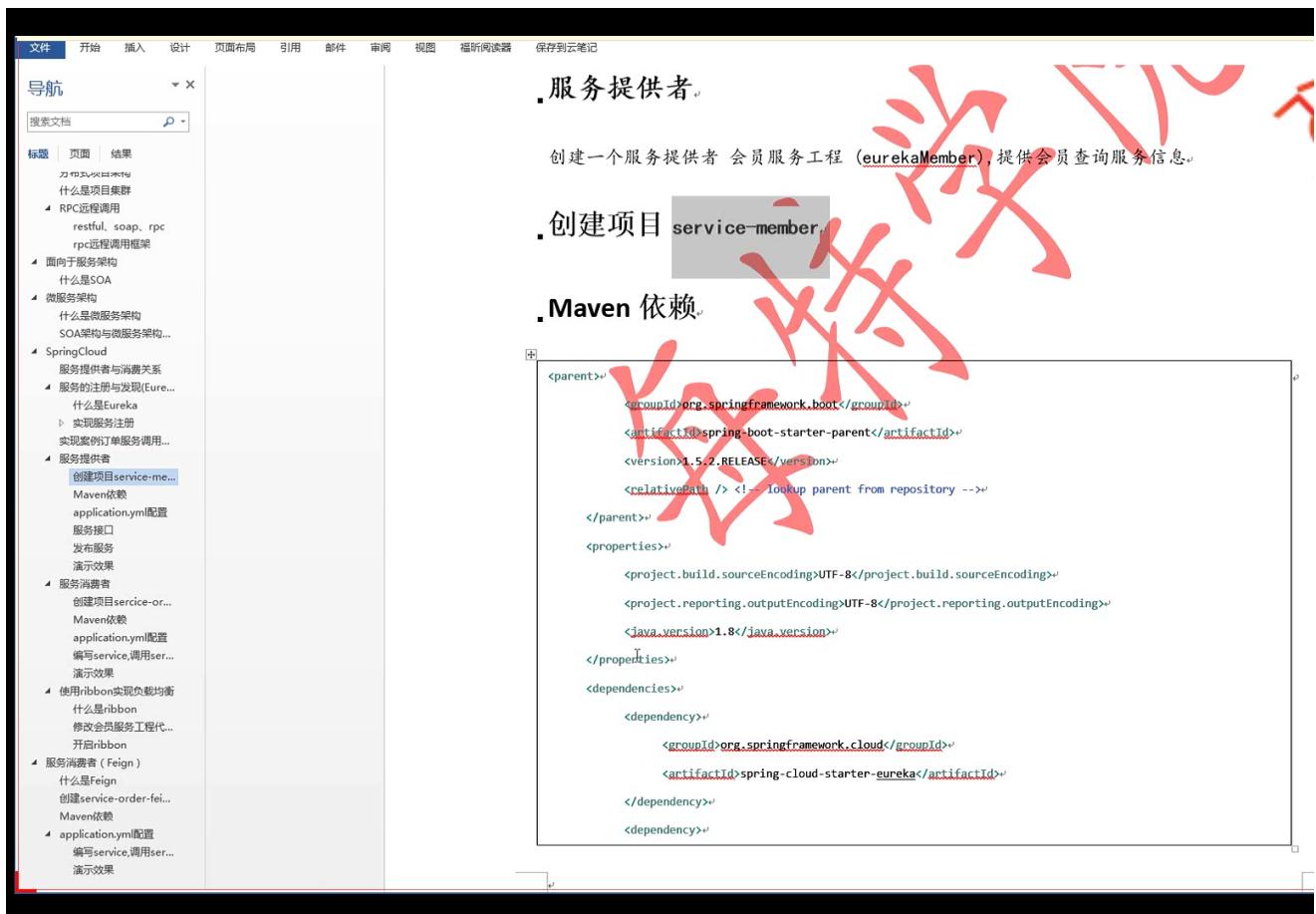
```
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RestController;
8
9 @RestController
10 public class MemberController {
11
12 @RequestMapping("/getMemberAll")
13 public List<String> getMemberAll() {
14 List<String> listUser = new ArrayList<String>();
15 listUser.add("zhangsan");
16 listUser.add("lisi");
17 listUser.add("wangwu");
18
19 }
}
```
- Console:** Shows the command-line output of the application running:

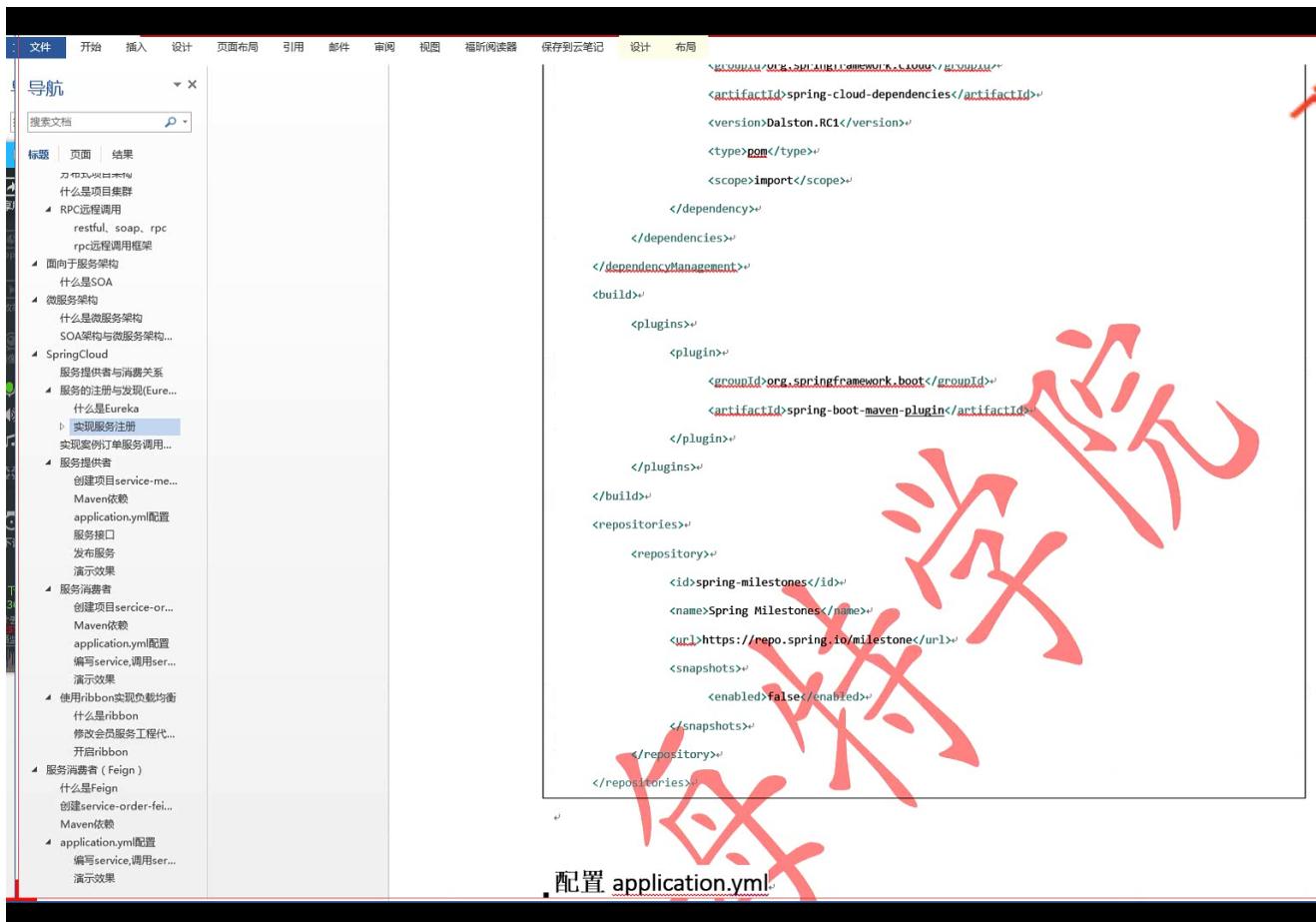
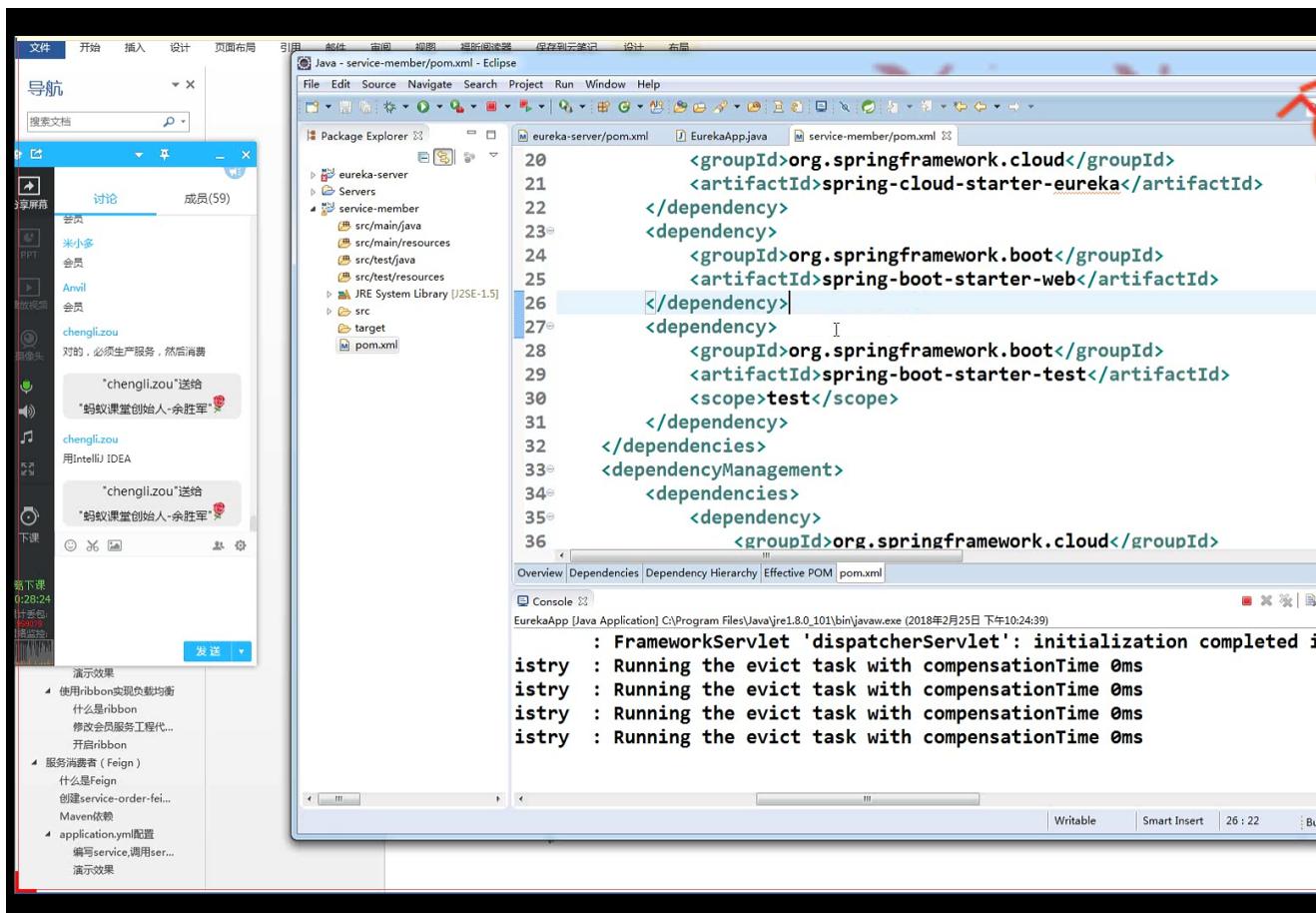
```
EurekaApp [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (2018年2月25日下午10:24:39)
istray : Running the evict task with compensationTime 0ms
istray : Running the evict task with compensationTime 0ms
istray : Running the evict task with compensationTime 0ms
istray : Running the evict task with compensationTime 0ms
istray : Running the evict task with compensationTime 0ms
```

The screenshot shows the course navigation sidebar with the following handwritten annotations:

- 顶部:** 手把手项目实操
- 标题:** 页面 结果
- RPC远程调用:** restful, soap, rpc  
什么是RPC
- 面向于服务架构:** 什么是SOA
- 微服务架构:** 什么是微服务架构  
SOA架构与微服务架构...
- SpringCloud:** 服务提供者与消费关系  
服务的注册与发现(Eureka)  
什么是Eureka  
实现案例订单服务调用...
- 服务提供者:** 创建项目service-me...  
Maven依赖  
application.yml配置  
服务接口  
发布服务  
演示效果
- 服务消费者:** 创建项目service-or...  
Maven依赖  
application.yml配置  
编写service,调用ser...  
演示效果
- 使用ribbon实现负载均衡:** 什么是ribbon  
修改会员服务工程代...  
开启ribbon
- 服务消费者(Feign):** 什么是Feign  
创建service-order-fei...  
Maven依赖  
application.yml配置  
编写service,调用ser...  
演示效果

大量的红色手写注释覆盖了右侧的内容，包括对Spring Cloud和相关技术的深入分析。





**System Status**

Environment	test	Current time	2018-02-25T22:25:44
Data center	default	Uptime	00:01
		Lease expiration enabled	false
		Renews threshold	1
		Renews (last min)	0

**DS Replicas**

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
No instances available			

**General Info**

Name	Value
total-avail-memory	398mb
environment	test
num-of-cpus	4
current-memory-usage	157mb (39%)
server-upptime	00:01
registered-replicas	unavailable-replicas

**DS Replicas**

Instances currently registered with Eureka

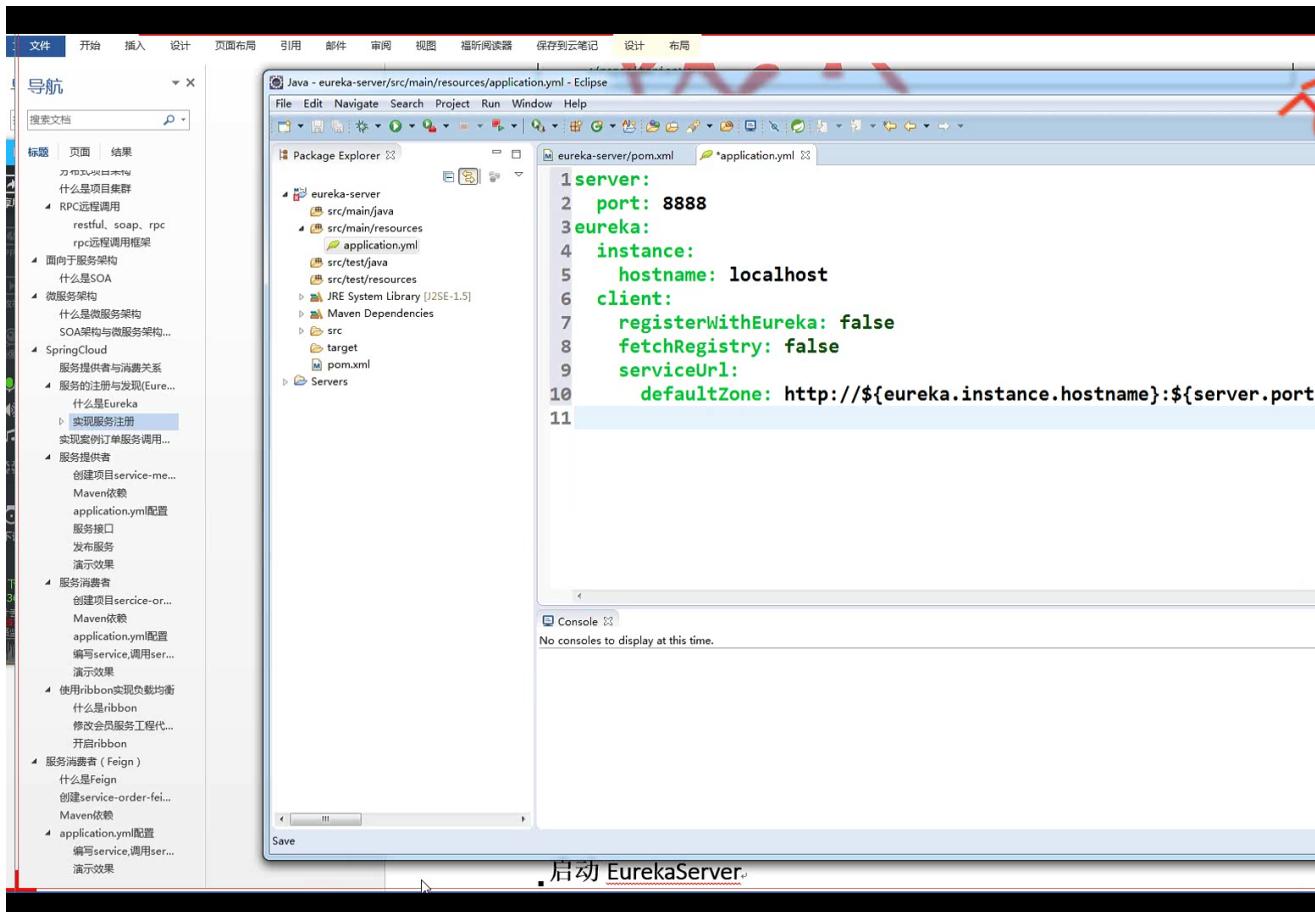
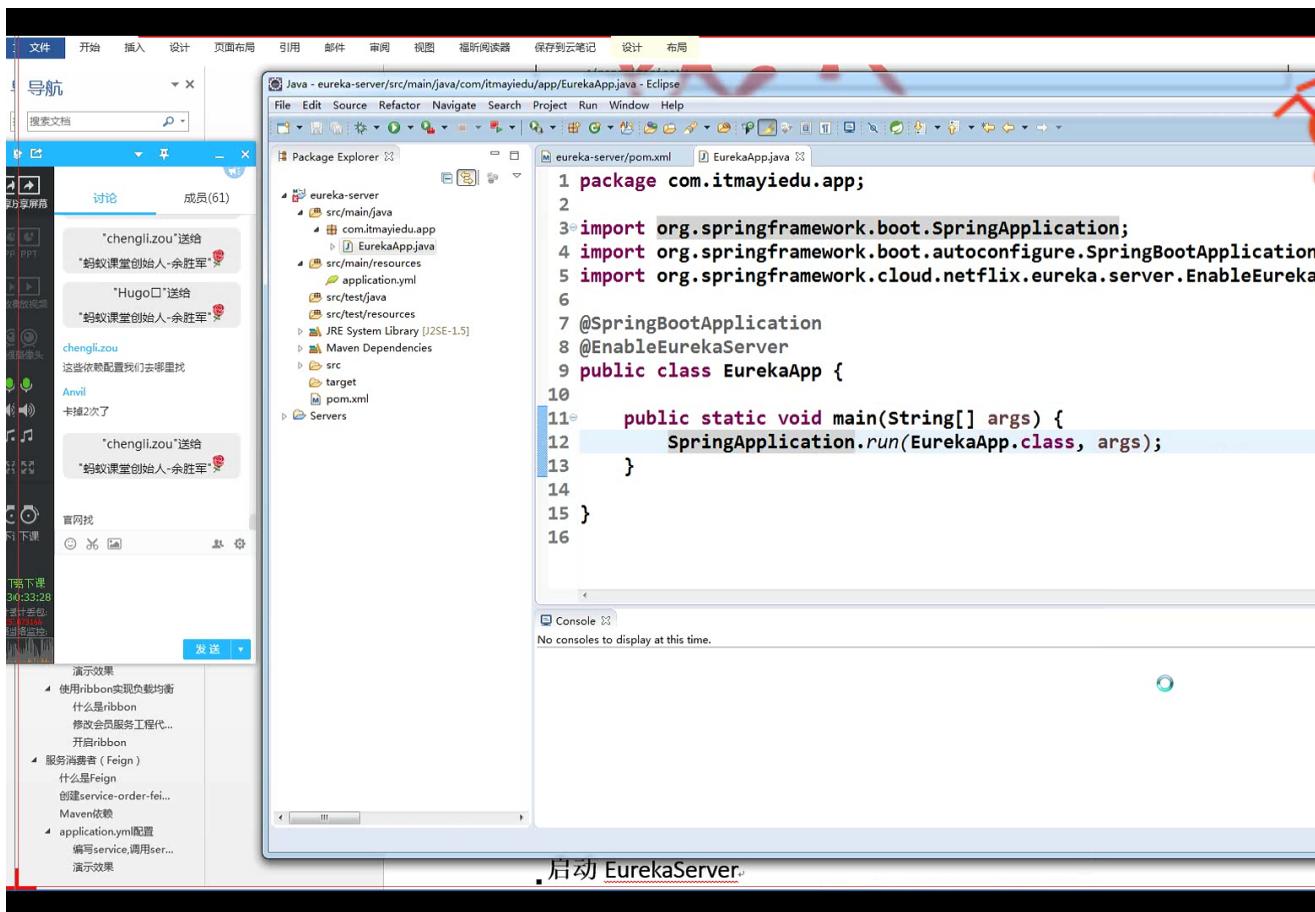
Application	AMIs	Availability Zones	Status
No instances available			

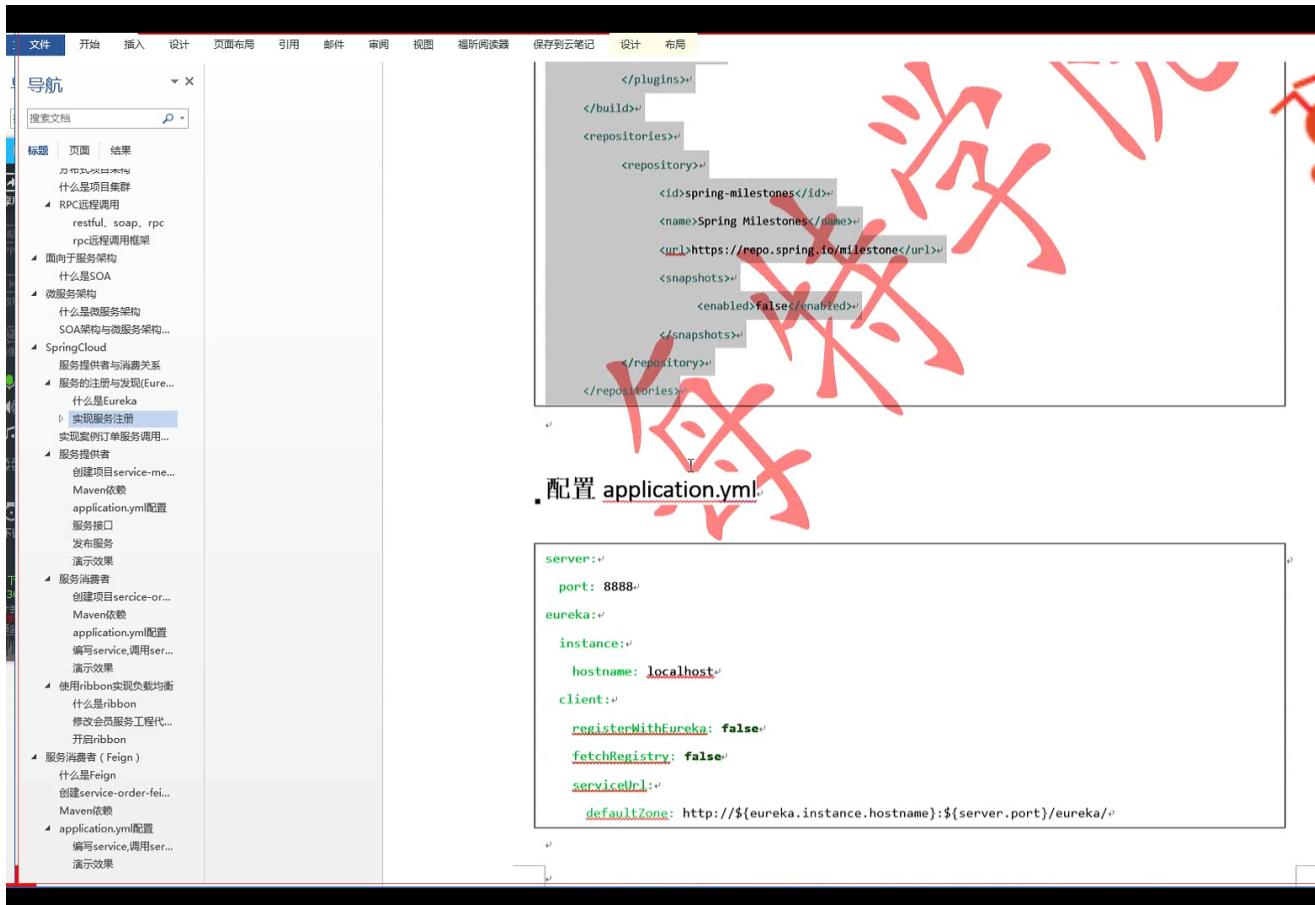
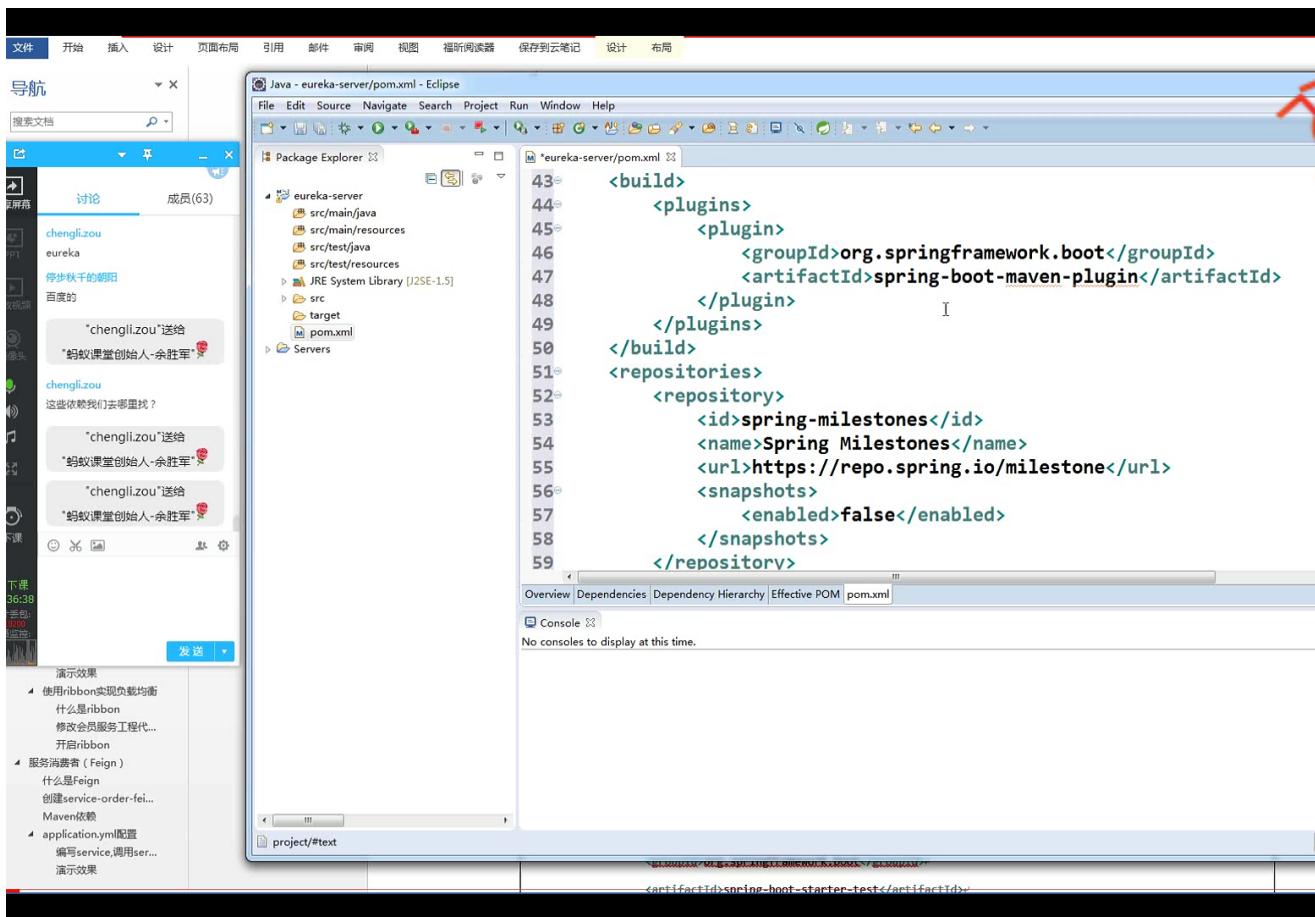
**General Info**

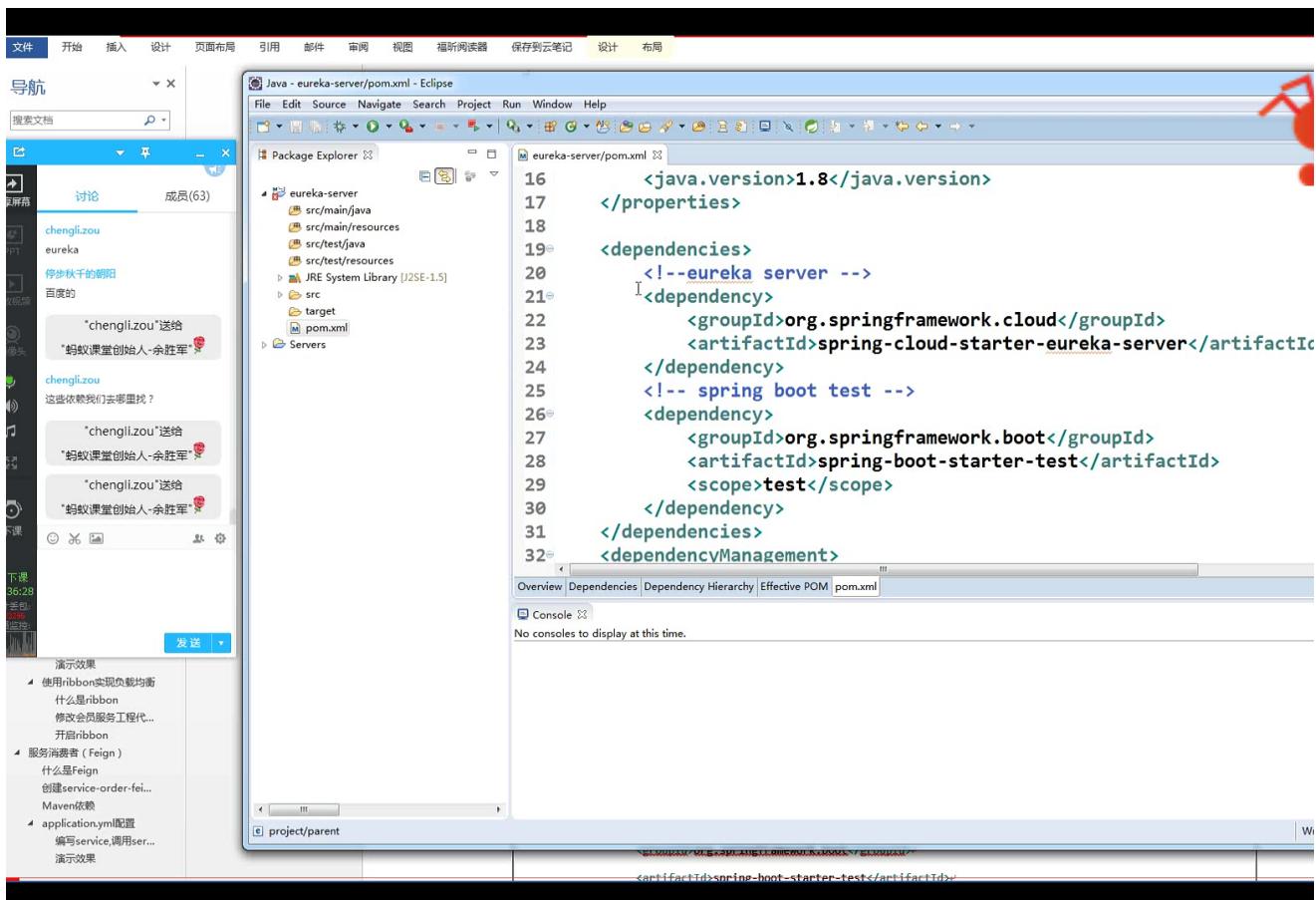
Name	Value
total-avail-memory	398mb
environment	test
num-of-cpus	4
current-memory-usage	157mb (39%)
server-upptime	00:01
registered-replicas	unavailable-replicas
available-replicas	

**Instance Info**

Name	Value
ipAddr	192.168.1.3
status	UP

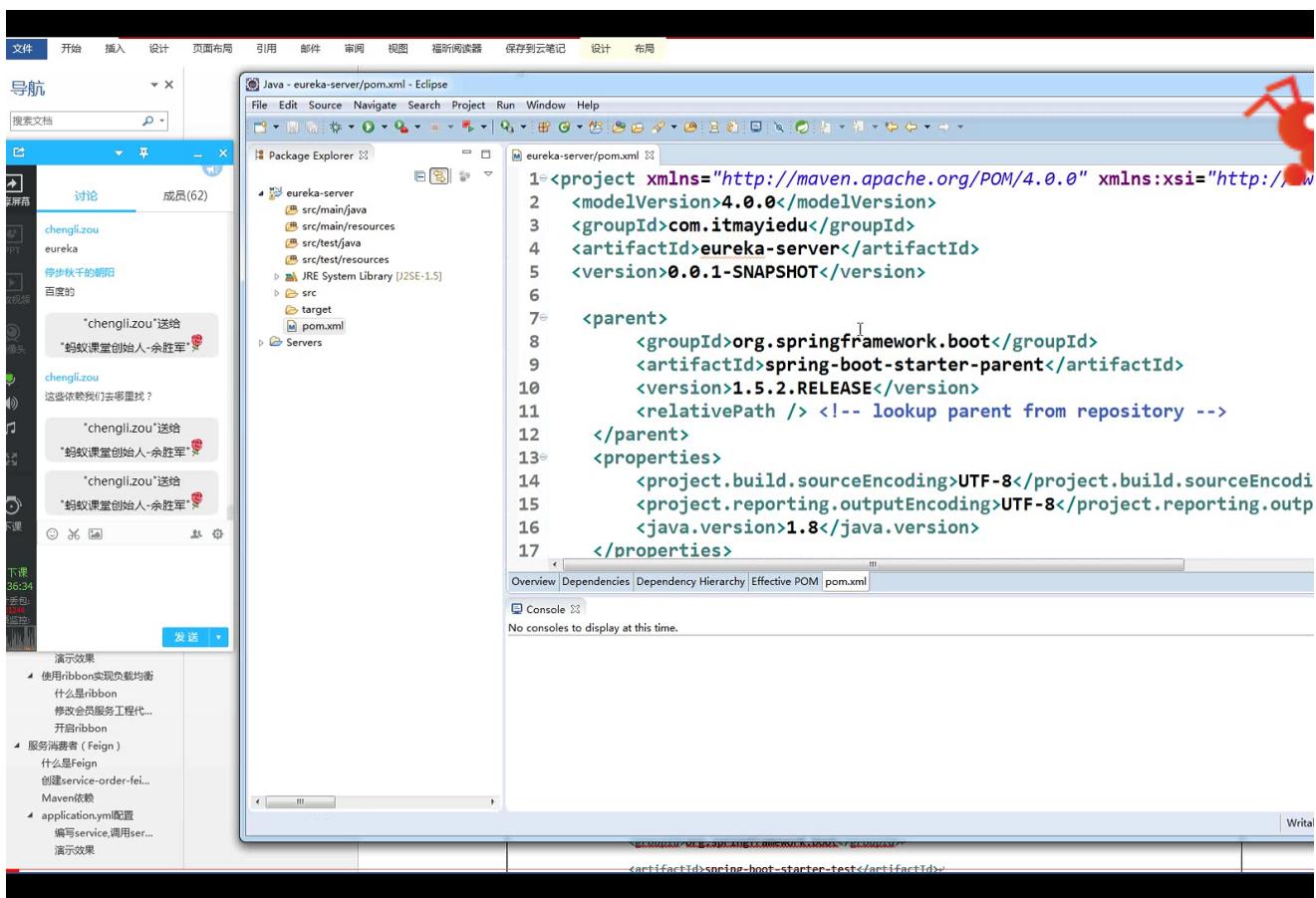






```
<java.version>1.8</java.version>
</properties>

<dependencies>
 <!--eureka server -->
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-eureka-server</artifactId>
 </dependency>
 <!-- spring boot test -->
 <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-test</artifactId>
 <scope>test</scope>
 </dependency>
</dependencies>
<dependencyManagement>
```



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>com.itmayiedu</groupId>
 <artifactId>eureka-server</artifactId>
 <version>0.0.1-SNAPSHOT</version>

 <parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-parent</artifactId>
 <version>1.5.2.RELEASE</version>
 <relativePath /> <!-- lookup parent from repository -->
 </parent>
 <properties>
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
 <java.version>1.8</java.version>
 </properties>
```

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记

导航

搜索文档

标题 | 页面 | 结果

什么是Eureka  
什么是项目集群  
RPC远程调用  
restful, soap, rpc  
rpc远程调用框架  
面向服务架构  
什么是SOA  
微服务架构  
什么是微服务架构  
SOA架构与微服务架构...  
SpringCloud  
服务提供者与消费者关系  
服务的注册与发现(Eure...  
什么是Eureka  
实现服务注册  
实现案例:单服务调用...  
服务提供者  
创建项目service-me...  
Maven依赖  
application.yml配置  
服务接口  
发布服务  
演示效果  
服务消费者  
创建项目service-or...  
Maven依赖  
application.yml配置  
编写service,调用ser...  
演示效果  
使用ribbon实现负载均衡  
什么是ribbon  
修改会员服务工程代...  
开启ribbon  
服务消费者(Feign)  
什么是Feign  
创建service-order-fei...  
Maven依赖  
application.yml配置  
编写service,调用ser...  
演示效果

▪ 创建 EureKaserver 项目。

Maven 依赖

```
<parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-parent</artifactId>
 <version>1.5.2.RELEASE</version>
 <relativePath /> <!-- lookup parent from repository -->
</parent>
<properties>
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
 <java.version>1.8</java.version>
</properties>
<dependencies>
 <!-- eureka server -->
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-eureka-server</artifactId>
 </dependency>
 <!-- spring boot test -->
 <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-test</artifactId>
 </dependency>
</dependencies>
<dependencyManagement>
```

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记

导航

搜索文档

标题 | 页面 | 结果

什么是Eureka  
什么是项目集群  
RPC远程调用  
restful, soap, rpc  
rpc远程调用框架  
面向服务架构  
什么是SOA  
微服务架构  
什么是微服务架构  
SOA架构与微服务架构...  
SpringCloud  
服务提供者与消费者关系  
服务的注册与发现(Eure...  
什么是Eureka  
实现服务注册  
实现案例:单服务调用...  
服务提供者  
创建项目service-me...  
Maven依赖  
application.yml配置  
服务接口  
发布服务  
演示效果  
服务消费者  
创建项目service-or...  
Maven依赖  
application.yml配置  
编写service,调用ser...  
演示效果  
使用ribbon实现负载均衡  
什么是ribbon  
修改会员服务工程代...  
开启ribbon  
服务消费者(Feign)  
什么是Feign  
创建service-order-fei...  
Maven依赖  
application.yml配置  
编写service,调用ser...  
演示效果

▪ 实现服务注册

(每特学院&蚂蚁课堂) 上海每特教育科技有限公司 [www.itmayiedu.com](http://www.itmayiedu.com) 余老师 QQ644064779

```
<dependencies>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-starter-eureka-server</artifactId>
 </dependency>
 <!-- spring boot test -->
 <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-test</artifactId>
 <scope>test</scope>
 </dependency>
</dependencies>
<dependencyManagement>
```

```
<dependencies>
 <dependency>
 <groupId>org.springframework.cloud</groupId>
 <artifactId>spring-cloud-dependencies</artifactId>
 </dependency>
 <version>Dalston.RC1</version>
```

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福昕阅读器 保存到云笔记

导航 搜索文档

标题 | 页面 | 结果

- 什么是项目集群
- RPC远程调用
  - restful, soap, rpc
  - rpc远程调用框架
- 面向服务架构
  - 什么是SOA
- 微服务架构
  - 什么是微服务架构
  - SOA架构与微服务架构...
- SpringCloud
  - 服务提供者与消费者关系
  - 服务的注册与发现(Eureka)
    - 什么是Eureka
    - 实现服务注册
    - 实例案例订单服务调用...
  - 服务提供者
    - 创建项目service-me...
    - Maven依赖
    - application.yml配置
    - 服务接口
    - 发布服务
    - 演示效果
  - 服务消费者
    - 创建项目service-or...
    - Maven依赖
    - application.yml配置
    - 编写service,调用ser...
    - 演示效果
  - 使用ribbon实现负载均衡
    - 什么是ribbon
    - 修改会员服务工程代...
    - 开启ribbon
  - 服务消费者(Feign)
    - 什么是Feign
    - 创建service-order-fei...
    - Maven依赖
  - application.yml配置
    - 编写service,调用ser...
    - 演示效果

eureka(注册中心)

服务消费者 会员系统

服务提供者 订单系统

rpc

eureka作用:服务协调、负载均衡、容错、路由等。

## 什么是 Eureka?

官方的介绍在这里 Eureka wiki。Eureka 是 Netflix 开源的一个 RESTful 服务，主要用于服务的注册发现。Eureka 由两个组件组成：Eureka 服务器和 Eureka 客户端。Eureka 服务器用作服务注册服务器。Eureka 客户端是一个 java 客户端，用来简化与服务器的交互、作为轮询负载均衡器，并提供服务的故障切换支持。Netflix 在其生产环境中使用的是另外的客户端，它提供基于流量、资源利用率以及出错状态的加权负载均衡。

在我看来，Eureka 的吸引力来源于以下几点：

文件 开始 插入 设计 页面布局

导航 搜索文档

讨论 成员(63)

分享屏幕

注册中心 停步秋千的朝阳

注册中心 Jetty

注册中心 eureka

注册中心 chengjizou

注册中心用zookeeper?

\*“清枫醉月”送给“蚂蚁课堂创始人-余胜军”\*

\*“未知”送给“...”\*

\*“...”送给“蚂蚁课堂创始人-余胜军”\*

\*“...”送给“...”\*

离下课 00:40:50

演示效果

使用ribbon实现负载均衡
 

- 什么是ribbon
- 修改会员服务工程代...
- 开启ribbon

服务消费者(Feign)
 

- 什么是Feign
- 创建service-order-fei...
- Maven依赖

application.yml配置
 

- 编写service,调用ser...
- 演示效果

1.SpringCloud注册中心环境搭建 curaka

2.服务注册与发现

3.SpringCloud客户端调用工具

rest, feign 客户端调用工具

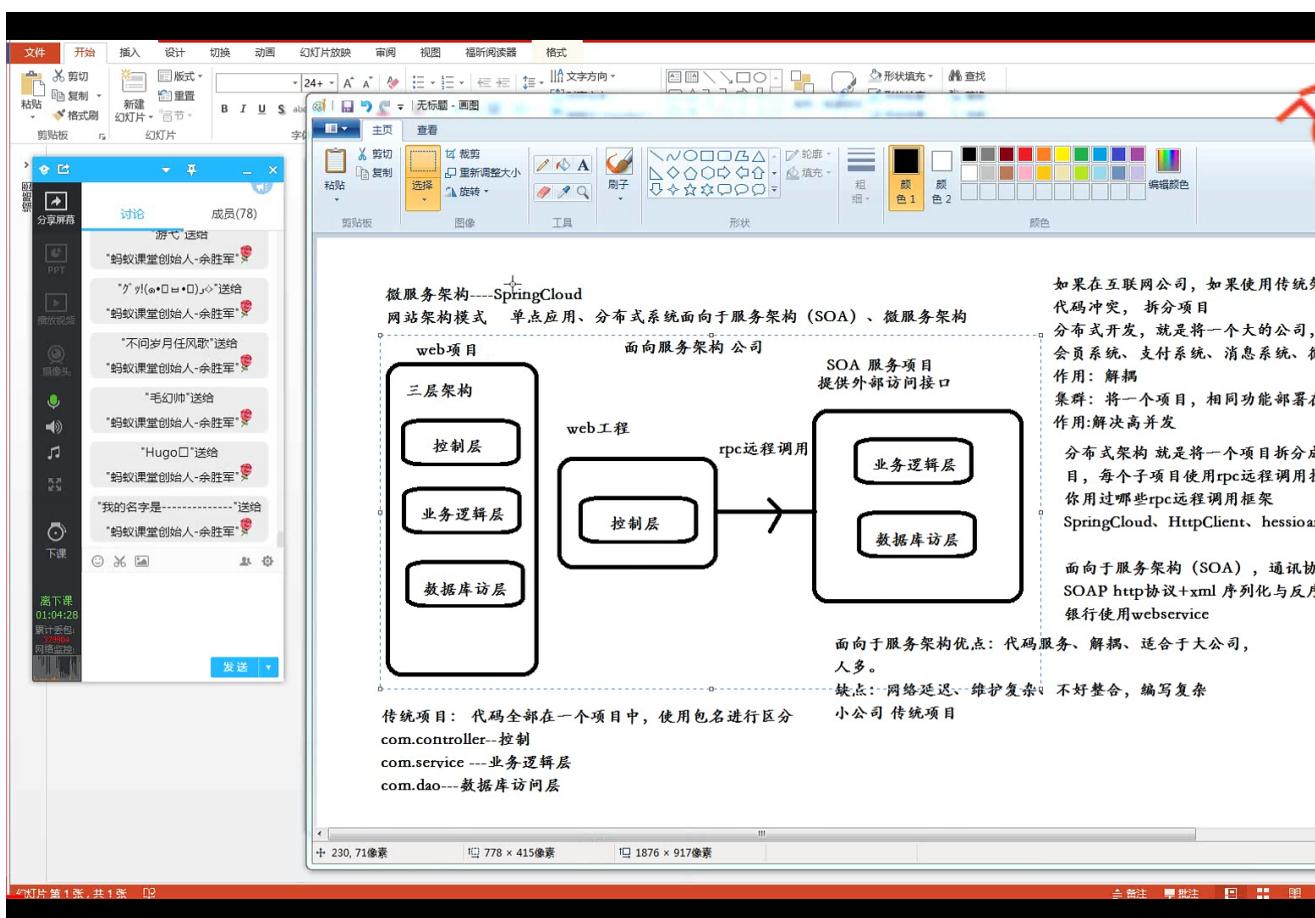
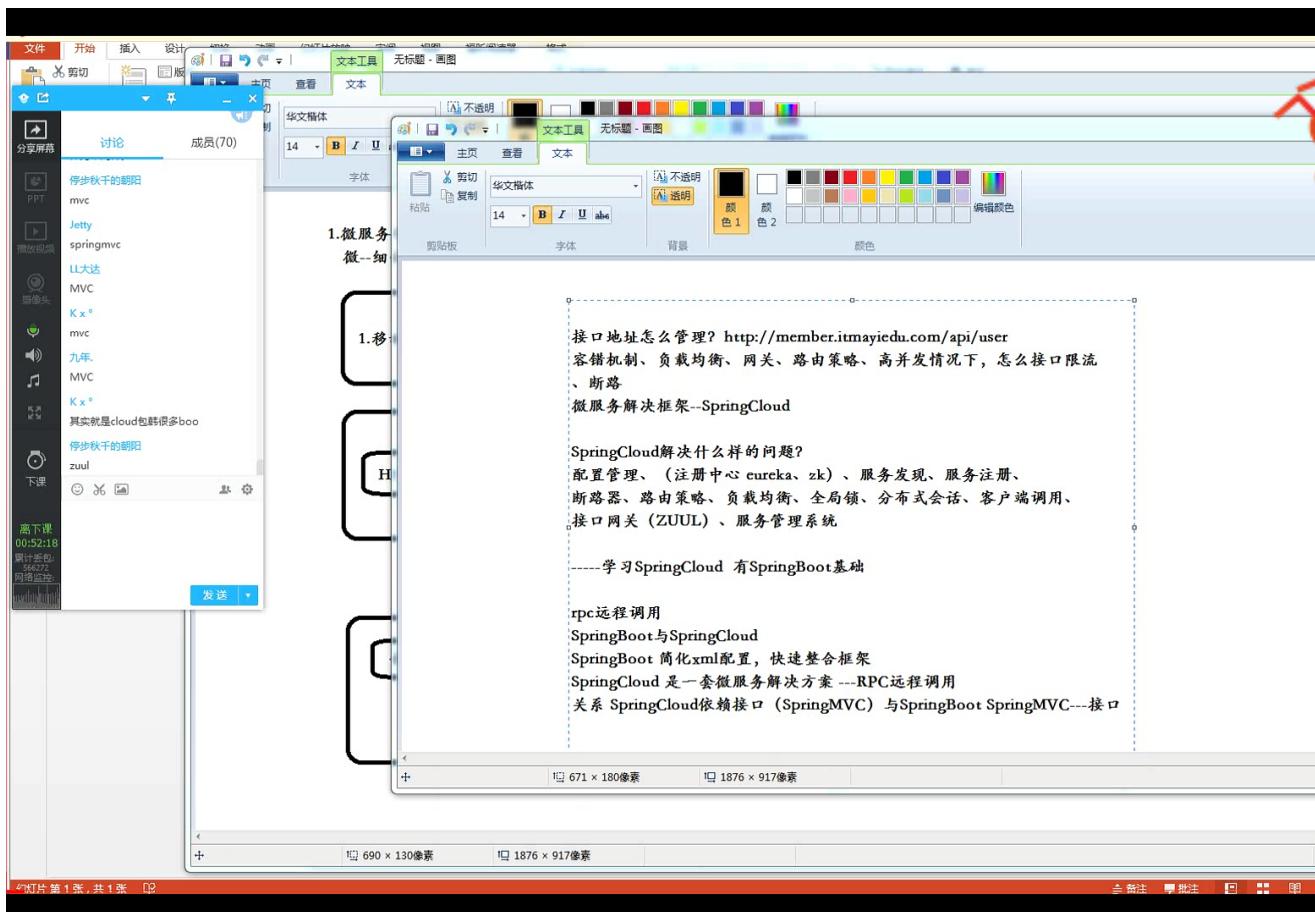
ribbon 负载均衡

zuul接口网关

rnreka服务注册

案例：会员服务提供用户信息、订单服务查询订单

订单服务需要查询用户，订单服务调用会员服务接口



讨论 成员(77)

向阳 服务提供方与调用方接口依赖方式太强  
停步秋天的朝阳 我们现在在用

Jetty 企业服务总线  
游弋 面向于服务架构不也是独立运行么  
青空 restful写错了

LL大大 解耦  
你微笑时好美 和调用微信接口差不多 对吗

下课 离下课 01:10:11

发送

**1. 微服务架构（分布式架构），实现传统soa架构领域升级**

微-细分，轻量级，通讯协议http协议+rest风格+json 服务都是独立运行

每个服务---对应一个数据库

1. 移动端(安卓、IOS端) 2.PC端 3.H5 (手机浏览)

每个服务---对应一个数据库

主流：rpc解决框架dubbo，springcloud

RPC远程调用 http协议+json格式+rest 互联网公司 httpclient  
使用比较简单通讯 使用httpclient

接口只允许在内网进行访问，和外网接口进行对接 https

**微服务架构与面向于服务架构区别：**  
面向于服务架构（SOA）主要针对在银行 xml格式  
微服务系统，会跟家细分，Http+json+restful进行

讨论 成员(76)

Jetty  
soap  
流星  
http+xml  
Sorry'HK  
http+  
Jetty  
smtp  
流星  
不高，数据量信息也大  
Don't Shy  
SOAP协议 = HTTP协议 + XML数据格式  
Jetty  
还有保险

下课 离下课 01:33:10

发送

**微服务架构---SpringCloud**

网站架构模式 单点应用、分布式系统面向于服务架构（SOA）、微服务架构

**web项目**

三层架构  
控制层  
业务逻辑层  
数据库访层

**面向服务架构公司**

SOA 服务项目 提供外部访问接口

控制层 → 业务逻辑层  
控制层 → 数据库访层

**传统项目：**代码全部在一个项目中，使用包名进行区分  
com.controller---控制  
com.service ---业务逻辑层  
com.dao---数据库访问层

**面向于服务架构优点：**代码服务、解耦、适合于大公司，人多。  
**缺点：**网络延迟、维护复杂、不好整合，编写复杂

小公司 传统项目

如果在互联网公司，如果使用  
代码冲突，拆分项目  
分布式开发，就是将一个大的  
会员系统、支付系统、消息系  
统作用：解耦  
集群：将一个项目，相同功  
能作用：解决高并发  
分布式架构 就是将一个项目  
目，每个子项目使用rpc远  
你用过哪些rpc远程调用框架  
SpringCloud、HttpClient、h  
面向于服务架构（SOA），  
SOAP http协议+xml 序列化  
银行使用webservice

幻灯片放映 审阅 视图 福昕阅读器 格式

剪切 复制 粘贴 格式刷 剪切板 分享屏幕

讨论 成员(78)

1. 微服务架构（分布式架构），实现传统soa架构领域升级  
微-细分，轻量级，通讯协议http协议+rest风格+json 服务都是独立运行  
来源 每个服务---对应一个数据库

1. 移动端(安卓、IOS端) 2.PC端 3.H5 (手机浏览)

```
graph TD; H5[H5 工程] --> MS[会员服务]; PC[PC 工程] --> OS[订单服务]; MD[混合开发] --> PS[支付服务]; subgraph External [RPC 远程调用 http 协议 + json 格式 + rest 互联网公司 httpclient]; direction TB; End1 --- MS; End1 --- OS; End1 --- PS; end;
```

幻灯片第 1 张, 共 1 张