

课时11

Spring Cloud 常见问题及解决方案

1. Eureka 服务快速发现的背景
2. Eureka 服务发现慢的原因
3. Spring Cloud 各组件超时
4. Spring Cloud 各组件调优
5. Zuul 动态路由
6. Feign 动态日志级别

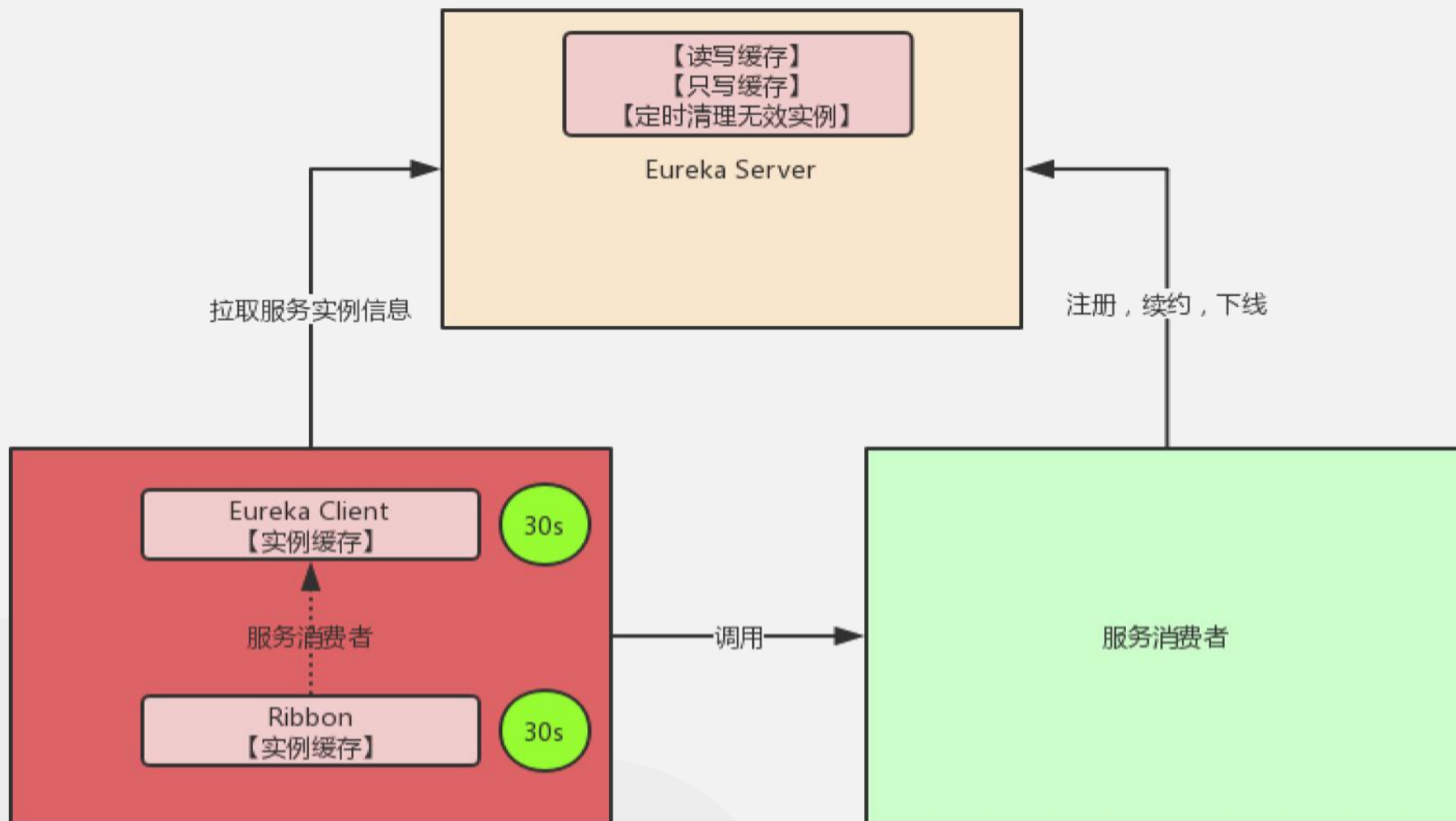


新服务上线后，服务消费者中不能访问到刚上线的新服务
过一段时间后才能访问？



将服务下线后，服务还是会被调用到，一段时候后才不会被
调用到，前期会导致频繁报错？

Eureka 服务发现慢的原因



服务端缓存

```
# 只读缓存更新时间间隔
eureka.server.response-
cache-update-interval-ms=3000
# 检测无效服务时间间隔
eureka.server.eviction-
interval-timer-in-ms=3000
# 禁用只读缓存
#eureka.server.use-read-
only-response-cache=false
```

客户端缓存

Eureka Client缓存
解决方案：
eureka.client.registryF
etchIntervalSeconds=5

Ribbon缓存
解决方案：
ribbon.ServerListRefreshInterval=3000



- Zuul
zuul.host.connect-timeout-millis
zuul.host.socket-timeout-millis
- Ribbon
ribbon.ReadTimeout
ribbon.ConnectTimeout
- Hystrix
hystrix.command.default.execution.isolation.thread.timeoutInMilliseconds
- Feign
feign.client.config.optimization-user.connectTimeout
feign.client.config.optimization-user.readTimeout

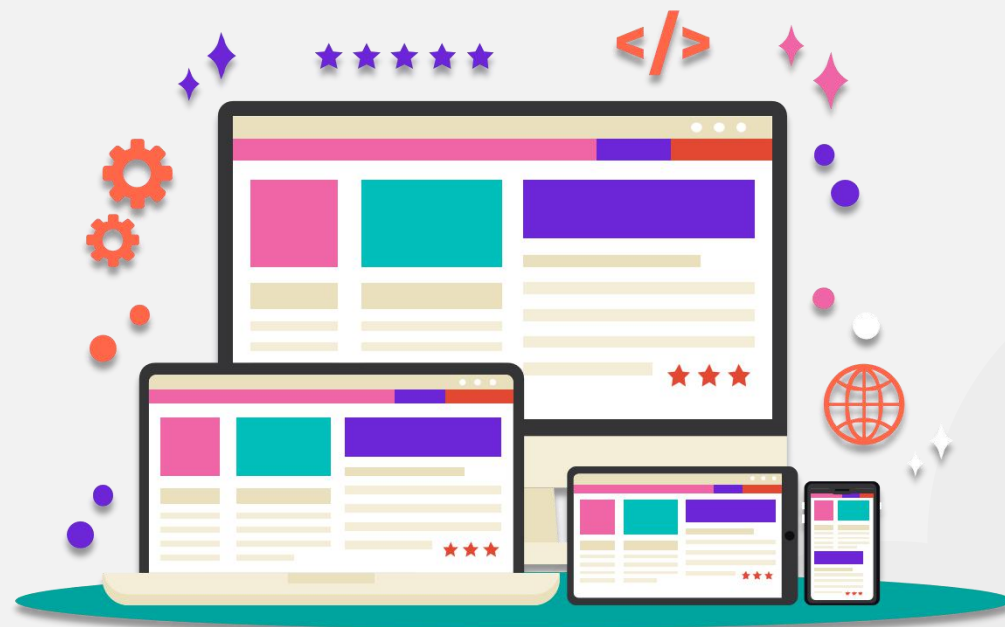
Spring Cloud 各组件重试

- Ribbon
 - # 对当前实例的重试次数
`ribbon.MaxAutoRetries=1`
 - # 切换实例的重试次数
`ribbon.MaxAutoRetriesNextServer=3`
 - # 对所有操作请求都进行重试
`ribbon.OkToRetryOnAllOperations=true`

- Zuul
`zuul.retryable=true`

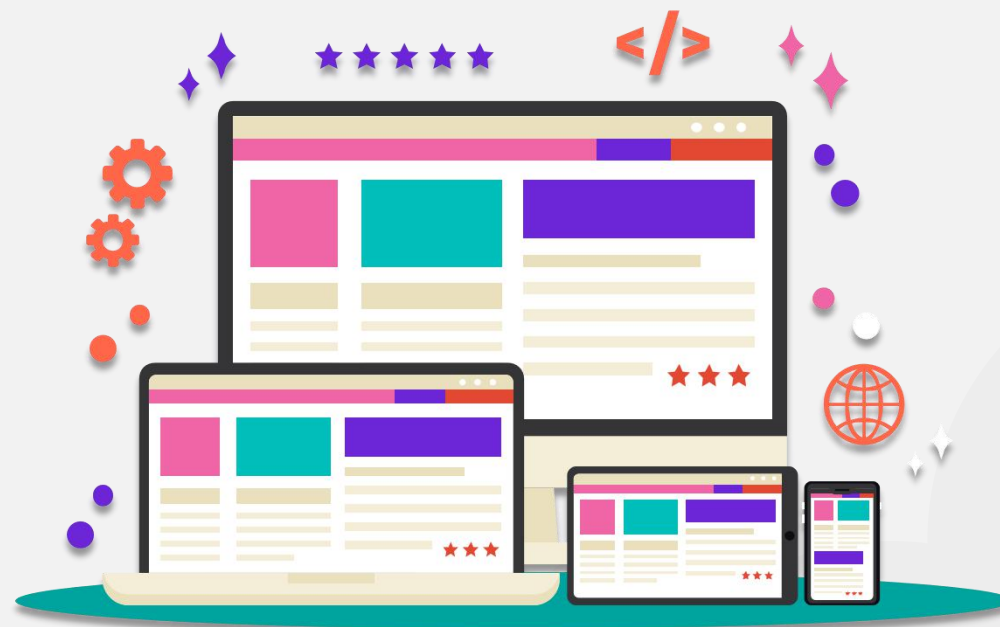
spring-retry依赖

```
<dependency>  
  <groupId>org.springframework.retry</groupId>  
  <artifactId>spring-retry</artifactId>  
</dependency>
```



Spring Cloud 各组件调优

- Zuul
zuul.host.max-per-route-connections
zuul.host.max-total-connections
- Hystrix
zuul.semaphore.max-semaphores
hystrix.threadpool.default.coreSize
hystrix.threadpool.default.maximumSize
hystrix.threadpool.default.maxQueueSize
- Ribbon
ribbon.MaxConnectionsPerHost
ribbon.MaxTotalConnections
- Feign
feign.httpclient.max-connections-per-route
feign.httpclient.max-connections



结合Apollo动态刷新路由

```
applicationContext.publishEvent(new EnvironmentChangeEvent(changeEvent.changedKeys()));
```

```
applicationContext.publishEvent(new RoutesRefreshedEvent(routeLocator));
```



Feign动态日志级别

项目中指定FeignClient的日志级别：

```
feign.client.config.<feignClientName>.loggerLevel=full
```

需要输出请求信息将对应的日志级别调整为debug，配置在Apollo中：

```
logging.level.com.example.article.client.UserRemoteClient = debug
```

监听Apollo配置修改，动态调整日志级别：

```
applicationContext.publishEvent(new EnvironmentChangeEvent(changedKeys));
```

Next: 课时12 《Spring Cloud 综合案例》