

系统间发布订阅设计原则与kafka使用规范

- 1、 批量发送 基类 提供 batch list 发送
- 2、 补偿 基类实现 （预定自动补偿）
- 3、 发送和接收 时间差， 比较大的情况处理。

消息前后顺序的问题。adj带着原始数据可以满足一些场景。

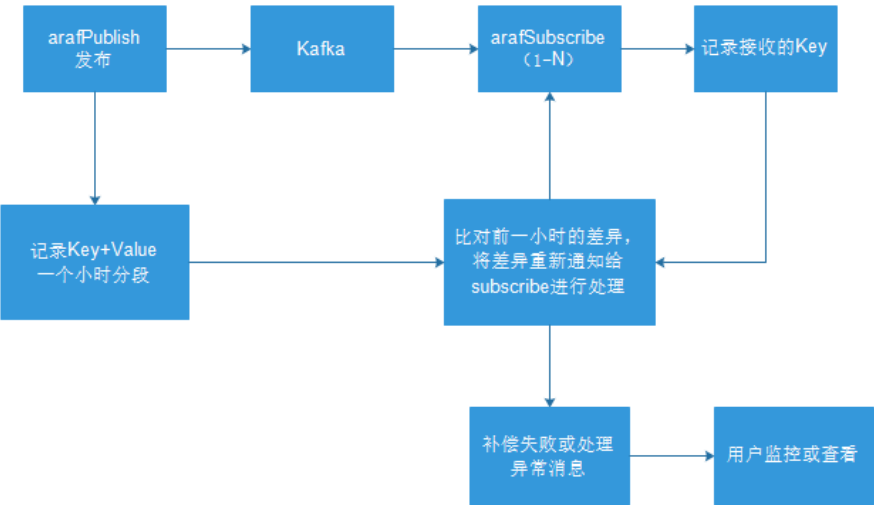
销售发布数据，税费，影像监听

发送全量数据。

运输实时查询销售。

销售 发布 销售数据到 kafka

引擎税费监听销售数据



Kafka队列命名规则

命名规则

{子系统名}_{模块名}_{业务名称}

Kafka配置

delete.topic.enable=true

```
partition.number=4
```

服务器需要配置默认分区为4. 增加并行度。

发布订阅配置

```
:ArafPublish.getDataTypeId() 与 :ArafSubscribe getDataTypeId() 一致
```

```
spring.adp.publish.semaphore: 100 发送最大并发。
```

发布订阅设计

依赖:

```
<dependency>
<groupId>com.acca</groupId>
<artifactId>araf-job-spring-boot-starter</artifactId>
<version>1.0.0</version>
</dependency>
```

架构封装基类，保证消息的完整性

发布基类实现部分代码说明

```
send(ArafPubSubKey k, V v)
```

ArafPublish

```
@Slf4j
@Component
public class SalRptTfsDetailPublish extends
ArafPublish<PubSalRptTfsDetailVO>{
    @Override
    public String getDataTypeId() {
        return Constants.PUB_SUB_SALRPTTFSDETAIL_IMPORT;
    }
    @Autowired
    private SalRptTfsDetailService salRptTfsDetailService;
    //
    @Scheduled(cron = "0 0/3 * * * ?")
    public void sendData() {
        log.debug("sendData ..... ");
        DevQueryVO qvo = new DevQueryVO();
        qvo.setSkip(0);
        qvo.setTake(100);
        Page<SalRptTfsDetail> list = salRptTfsDetailService.findAll(qvo);
        int count = 1;
        for(; count < 3000; ) {
            for (Iterator iterator = list.iterator(); iterator.hasNext();)
            {
                SalRptTfsDetail salRptTfsDetail = (SalRptTfsDetail)
iterator.next();
                salRptTfsDetail.setSeq(UUID.randomUUID().toString());
                salRptTfsDetailService.add(salRptTfsDetail);
                PubSalRptTfsDetailVO vo = new PubSalRptTfsDetailVO();
                ArafBeanUtils.copyProperties(salRptTfsDetail, vo);
                send(ArafPubSubKey.of(salRptTfsDetail.getSeq()), vo);
                count ++;
                if (count >= 3000) {
                    break;
                }
            }
        }
    }
}
```

订阅基类实现部分代码说明

```
DemoArafSubscribeService service.difference(dataTime)
```

ArafSubscribe

```
@Slf4j
@Component
public class SalRptTfsDetailSubscribe extends
ArafSubscribe<PubSalRptTfsDetailVO> {
    @Autowired
    private SalRptTfsDetailService salRptTfsDetailService;
    @Override
    protected void persist(ArafPubSubKey k, PubSalRptTfsDetailVO v) {
        log.debug("persist {} {} " , k , v);
        this.persist(v);
    }
    @Autowired
    private ArafPubSubStatus arafPubSubStatus;
    public PubSalRptTfsDetailVO persist(PubSalRptTfsDetailVO t) {
        SalRptTfsDetail x = new SalRptTfsDetail();
        ArafBeanUtils.copyProperties(t, x);
        salRptTfsDetailService.add(x);
        return t;
    }
    @Override
    public String getDataTypeId() {
        return Constants.PUB_SUB_SALRPTTFSDETAIL_IMPORT;
    }
}
```

- 可能的问题:
- 收消息完整性及做补偿

getOneDiff