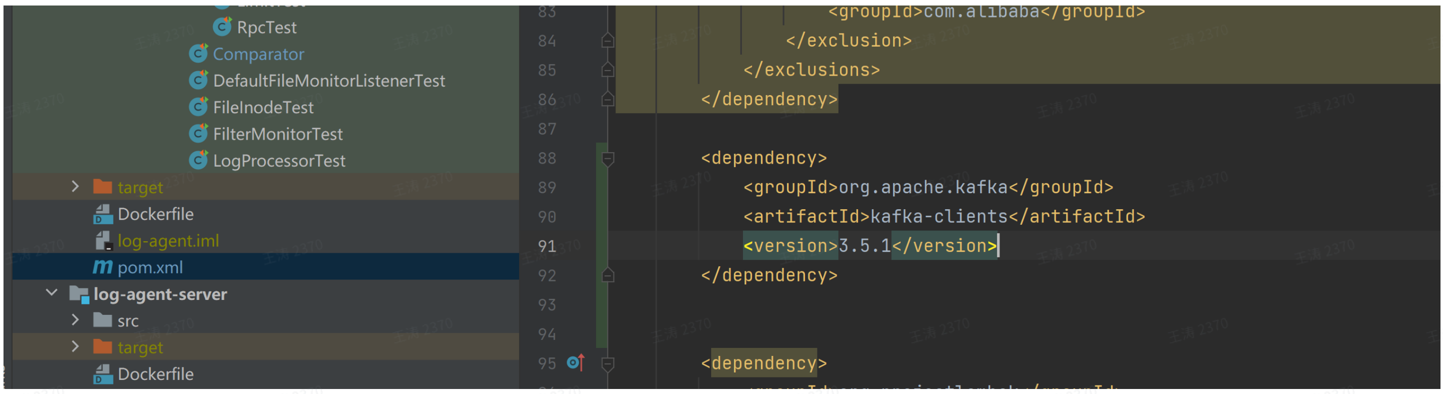
Introduction to the log-agent-exporter Extension Point

Implementation

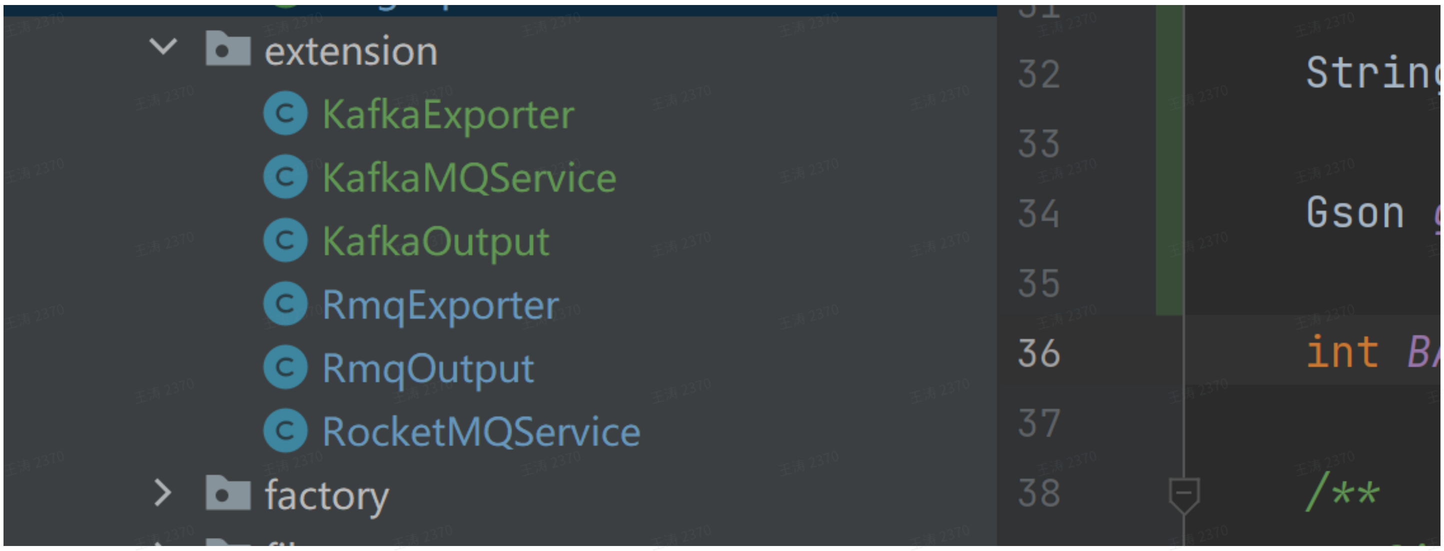
1.Currently, there are RocketMqServer, RmqExporter, and RmqOutput components.

2If you need to extend it to support Kafka, you can implement the corresponding interfaces and introduce the Kafka Maven coordinates. Usually, you should choose a relatively new version, which can be found in the Maven Central Repository, as shown below:



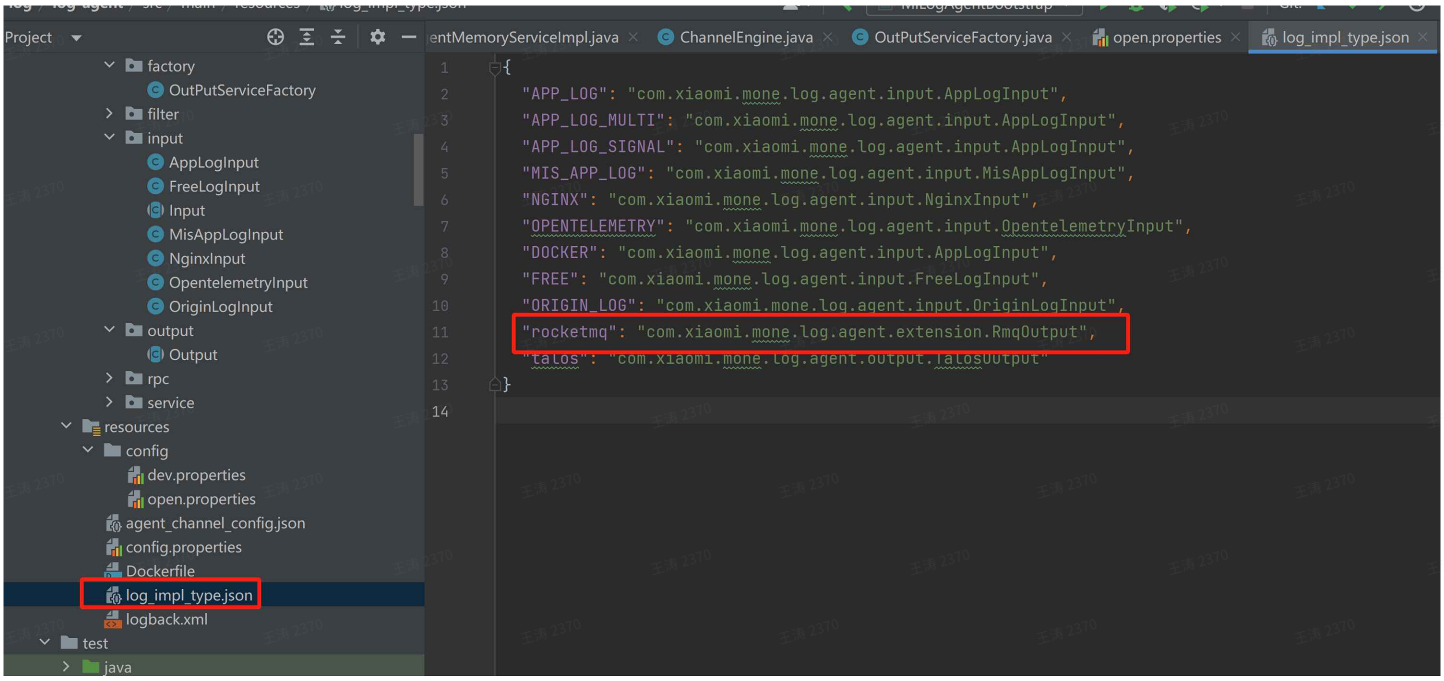
```
            <groupId>com.alibaba</groupId>
        </exclusion>
    </exclusions>
</dependency>

<dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka-clients</artifactId>
    <version>3.5.1</version>
</dependency>


<dependency>
```

Implement KafkaExporter, KafkaMQService, and KafkaOutput in the "extension" package based on the corresponding packages for RocketMQ, like this:



```
String
Gson g
int B
/**
```

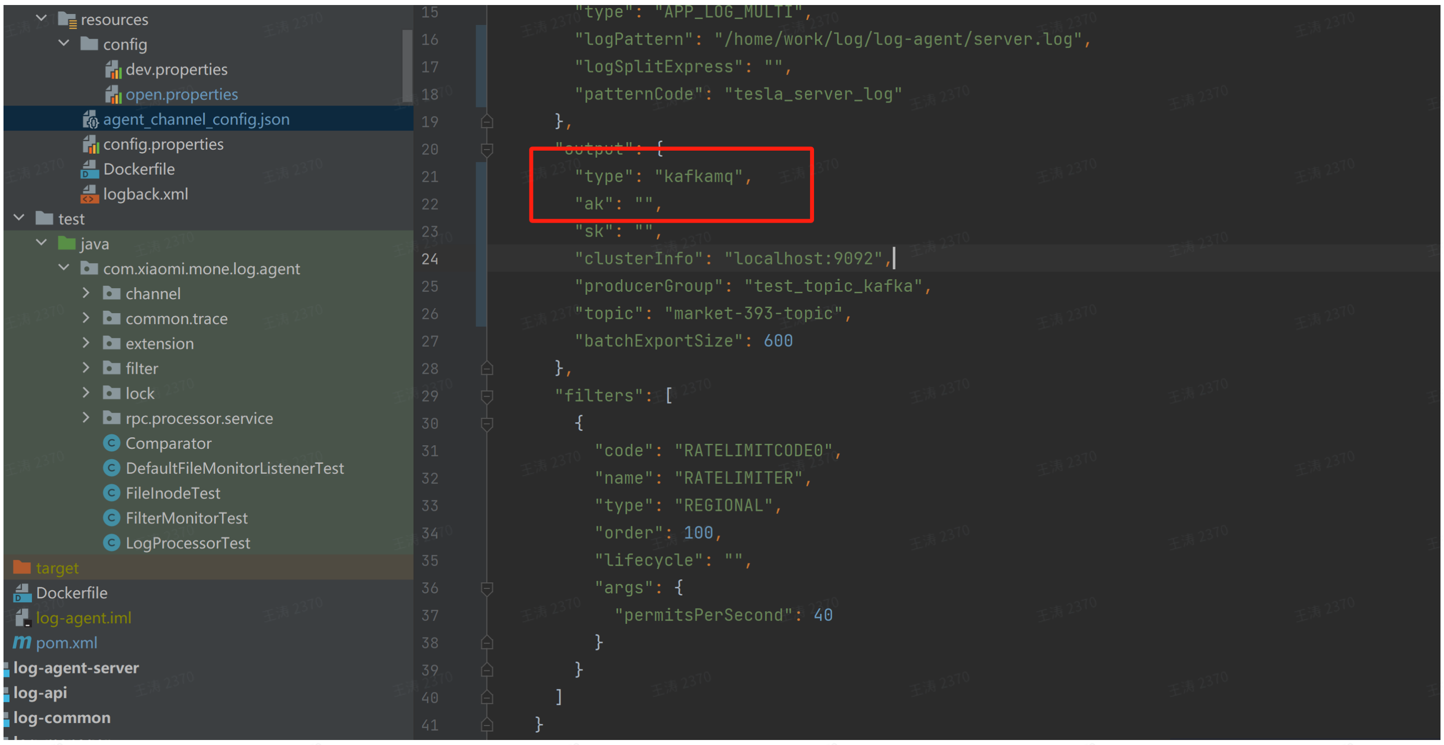
After creating KafkaOutput, add the serialization configuration for KafkaOutput in the "log_impl_type.json" file, as shown in the following example:



```
{
  "APP_LOG": "com.xiaomi.mone.log.agent.input.AppLogInput",
  "APP_LOG_MULTI": "com.xiaomi.mone.log.agent.input.AppLogInput",
  "APP_LOG_SIGNAL": "com.xiaomi.mone.log.agent.input.AppLogInput",
  "MIS_APP_LOG": "com.xiaomi.mone.log.agent.input.MisAppLogInput",
  "NGINX": "com.xiaomi.mone.log.agent.input.NginxInput",
  "OPENTELEMETRY": "com.xiaomi.mone.log.agent.input.OpentelemetryInput",
  "DOCKER": "com.xiaomi.mone.log.agent.input.AppLogInput",
  "FREE": "com.xiaomi.mone.log.agent.input.FreeLogInput",
  "ORIGIN_LOG": "com.xiaomi.mone.log.agent.input.OriginLogInput",
  "rocketmq": "com.xiaomi.mone.log.agent.extension.RmqOutput",
  "talos": "com.xiaomi.mone.log.agent.output.TalosOutput"
}
```

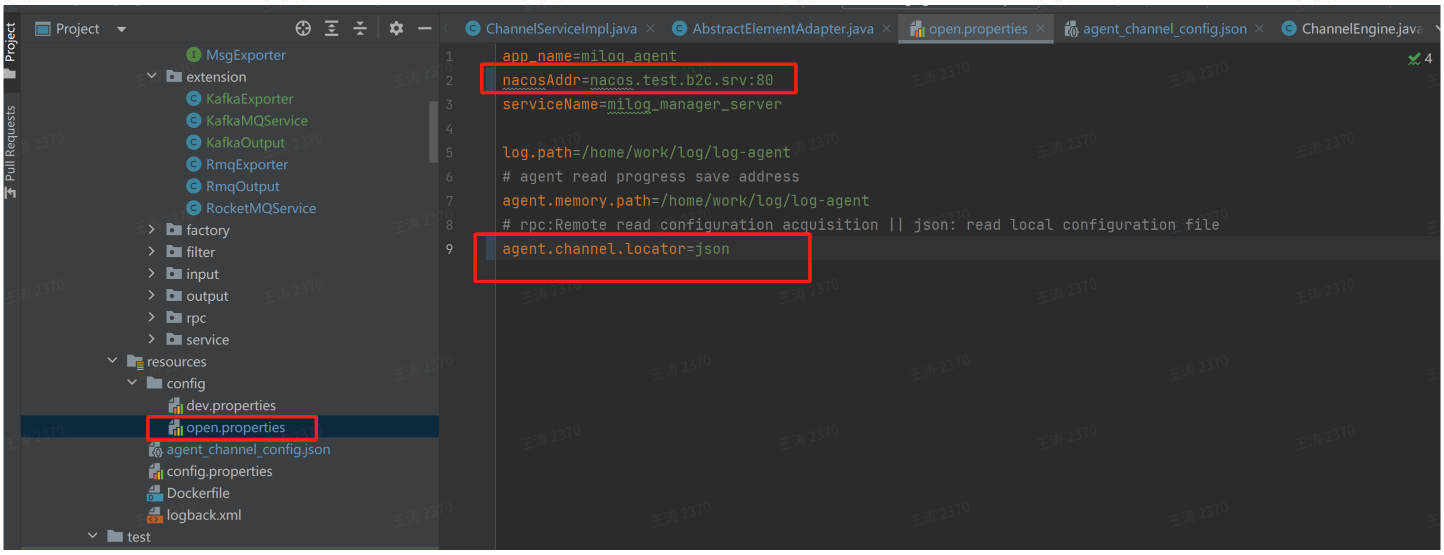
Testing:

Modify the "agent_channel_config.json" file located in the project's "resources" directory. Change "output" to "kafkamq" and update the Kafka cluster information, as shown below:



```
        "type": "APP_LOG_MULTI",
        "logPattern": "/home/work/log/log-agent/server.log",
        "logSplitExpress": "",
        "patternCode": "tesla_server_log"
      },
      "output": {
        "type": "kafkamq",
        "ak": "",
        "sk": "",
        "clusterInfo": "localhost:9092",
        "producerGroup": "test_topic_kafka",
        "topic": "market-393-topic",
        "batchExportSize": 600
      },
      "filters": [
        {
          "code": "RATELIMITCODE0",
          "name": "RATELIMITER",
          "type": "REGIONAL",
          "order": 100,
          "lifecycle": "",
          "args": {
            "permitsPerSecond": 40
          }
        }
      ]
    }
```

Modify the project's startup configuration file "open.properties" with the Nacos address and test the local configuration file:



```
app_name=milog_agent
nacosAddr=nacos.test.b2c.srv:80
serviceName=milog_manager_server

log.path=/home/work/log/log-agent
# agent read progress save address
agent.memory.path=/home/work/log/log-agent
# rpc:Remote read configuration acquisition || json: read local configuration file
agent.channel.locator=json
```