

微服务开发实例demo之库存管理服务

微服务开发实例demo之库存管理服务

- 1、提前条件
 - 1.1 创建经销商信息表
 - 1.2 创建门店信息表
 - 1.3 创建门店详情表
- 2、创建数据库
- 3、创建对应的表
 - 2.1 根据业务流程图，整理出ER图
- 4、generate生成对应的模块代码
 - 4.1 创建提前中的Agent和store通用代码
 - 4.2 如果hucais-generator-stock项目不存在就新建项目
 - 4.3 生成store_stock库存JAVA代码和Mapper.xml
- 5、编写接口文档 (showdoc.cc)
- 7、创建库存微服务
- 6、编写mybatis/**Mapper.xml文件
- 7、编写Mapper文件
- 8、编写service接口和serviceImp实现
 - 8.1 编写本地的service接口和方法
 - 8.2 编写远程调用的接口
- 9、编写controller
- 10、编写Feign客户端接口和Feign客户端实现
- 11、添加Hystrix熔断器
- 12、单元测试
- 13、PostMan接口测试
- 14、前端测试联调
 - 14.1 首先修改hucais-gateway路由地址为本地服务地址
 - 14.2 前端开发人员修改HOST地址
- 15、更新代码并提交到develop分支

1、提前条件

- 在hc_auth_platform数据里面创建经销商信息表、门店信息表、门店信息详情表

创建表的原则：独立、可拆分

1.1 创建经销商信息表

- 新建create_table_agent.sql

```
1  /*
2  经销商信息表
3  */
4  CREATE TABLE `agent` (
5    `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '表主键',
6    `agent_name` varchar(300) DEFAULT NULL COMMENT '经销商名称',
```

```

7  `short_name` varchar(100) DEFAULT NULL COMMENT '经销商名称简写',
8  `agent_sn` varchar(20) DEFAULT NULL COMMENT '经销商简写编码',
9  `agent_type` varchar(20) DEFAULT NULL COMMENT '经销商类型',
10 `description` varchar(255) DEFAULT NULL COMMENT '经销商描述信息',
11 `open_status` char(1) DEFAULT NULL COMMENT '开通状态0-开通 C-关闭 L-锁定',
12 `open_start_time` datetime DEFAULT NULL COMMENT '开通时间',
13 `open_end_time` datetime DEFAULT NULL COMMENT '开通结束时间',
14 `max_no` int(11) DEFAULT NULL COMMENT '经销商最大员工数量',
15 `orders` int(11) DEFAULT '0' COMMENT '预留排序字段',
16 `linkman` varchar(10) DEFAULT NULL,
17 `domain_name` varchar(50) DEFAULT NULL COMMENT '二级域名',
18 `creator` varchar(20) CHARACTER SET utf8 DEFAULT '' COMMENT '创建人',
19 `creator_id` bigint(20) DEFAULT NULL COMMENT '创建人ID',
20 `created_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
21 `last_operator` varchar(20) CHARACTER SET utf8 DEFAULT '' COMMENT '最近操作人',
22 `last_operator_id` bigint(20) DEFAULT NULL COMMENT '最后操作人ID',
23 `update_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '更新时间',
24 PRIMARY KEY (`id`),
25 UNIQUE KEY `uk_agent_sn` (`agent_sn`),
26 KEY `idx_agent_short_name` (`short_name`) USING BTREE,
27 KEY `idx_agent_type` (`agent_type`) USING BTREE,
28 KEY `idx_agent_open_status` (`open_status`) USING BTREE,
29 KEY `idx_agent_orders` (`orders`) USING BTREE
30 ) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=utf8 COMMENT='经销商信息表';

```

1.2 创建门店信息表

- 新建create_table_store.sql

```

1  /*
2  门店信息表
3  */
4  CREATE TABLE `store` (
5      `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '表主键',
6      `agent_id` bigint(20) DEFAULT NULL COMMENT '经销商ID, agent.id',
7      `store_name` varchar(300) DEFAULT NULL COMMENT '门店名称',
8      `short_name` varchar(100) DEFAULT NULL COMMENT '门店名称简写',
9      `store_sn` varchar(20) DEFAULT NULL COMMENT '门店简写编码',
10     `store_type` varchar(20) DEFAULT NULL COMMENT '门店类型',
11     `description` varchar(255) DEFAULT NULL COMMENT '门店描述信息',
12     `open_status` char(1) DEFAULT NULL COMMENT '开通状态0-开通 C-关闭 L-锁定',
13     `open_start_time` datetime DEFAULT NULL COMMENT '开通时间',
14     `open_end_time` datetime DEFAULT NULL COMMENT '开通结束时间',
15     `max_no` int(11) DEFAULT NULL COMMENT '门店最大员工数量',
16     `orders` int(11) DEFAULT '0' COMMENT '预留排序字段',
17     `linkman` varchar(10) DEFAULT NULL,
18     `domain_name` varchar(50) DEFAULT NULL COMMENT '二级域名',
19     `creator` varchar(20) CHARACTER SET utf8 DEFAULT '' COMMENT '创建人',
20     `creator_id` bigint(20) DEFAULT NULL COMMENT '创建人ID',
21     `created_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
22     `last_operator` varchar(20) CHARACTER SET utf8 DEFAULT '' COMMENT '最近操作人',
23     `last_operator_id` bigint(20) DEFAULT NULL COMMENT '最后操作人ID',
24     `update_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '更新时间',

```

```

25     PRIMARY KEY (`id`),
26     UNIQUE KEY `uk_store_sn` (`store_sn`),
27     KEY `idx_store_short_name` (`short_name`) USING BTREE,
28     KEY `idx_store_type` (`store_type`) USING BTREE,
29     KEY `idx_store_open_status` (`open_status`) USING BTREE,
30     KEY `idx_store_orders` (`orders`) USING BTREE,
31     KEY `idx_store_agent_id` (`agent_id`)
32 ) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=utf8 COMMENT='门店信息表';

```

1.3 创建门店详情表

- 新建create_table_store_description_sql【未完成】

```

1  /*
2  门店信息详情表
3  */
4  CREATE TABLE `store_description` (
5      `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '表主键',
6      `store_id` bigint(20) DEFAULT NULL COMMENT '门店ID, store.id',
7      `store_img` varchar(300) DEFAULT NULL COMMENT '门店名称',
8      `creator` varchar(20) CHARACTER SET utf8 DEFAULT '' COMMENT '创建人',
9      `creator_id` bigint(20) DEFAULT NULL COMMENT '创建人ID',
10     `created_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
11     `last_operator` varchar(20) CHARACTER SET utf8 DEFAULT '' COMMENT '最近操作人',
12     `last_operator_id` bigint(20) DEFAULT NULL COMMENT '最后操作人ID',
13     `update_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '更新时间',
14     PRIMARY KEY (`id`)
15 ) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=utf8 COMMENT='门店信息详情表';

```

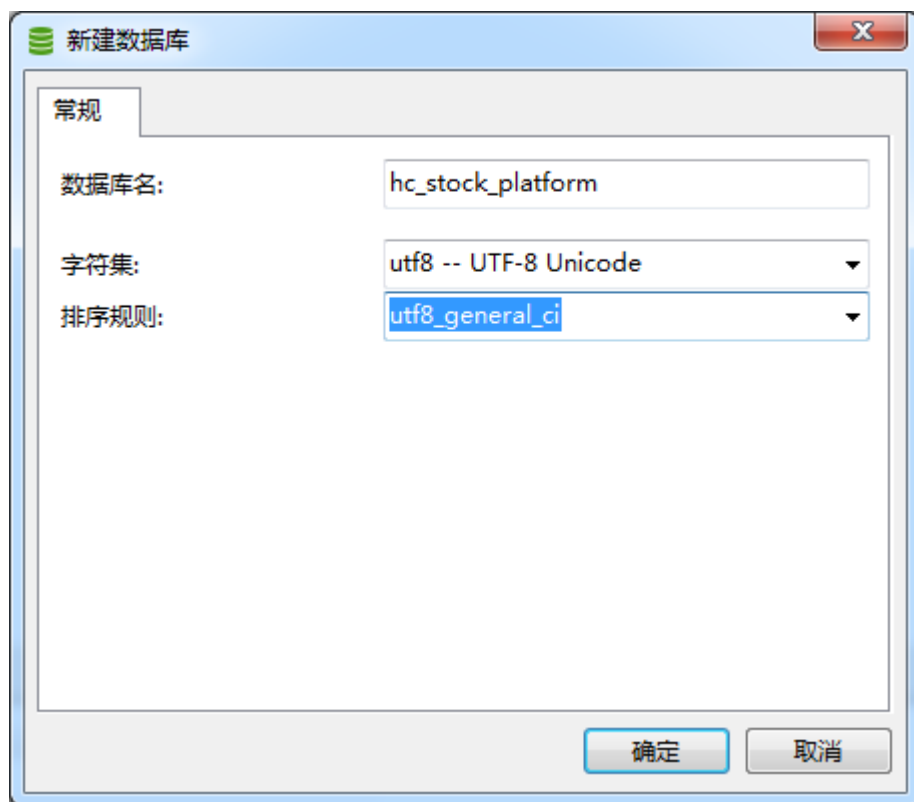
2、创建数据库

- 以库存管理微服务为例，需要单独创建一个门店的数据库hc_stock_platform
- 新建create_database_hc_stock_platform.sql

```

1  CREATE DATABASE hc_stock_platform;

```



3、创建对应的表

2.1 根据业务流程图，整理出ER图

整理出ER图，能够更加清晰的看出每一个表之间关联

- 进入hc_stock_platform数据库，创建库存表（以门店库存管理为例子）
- 新建create_table_store_stock.sql

```
1  /*
2  门店库存表
3  */
4  CREATE TABLE `store_stock` (
5      `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '表主键',
6      `store_id` bigint(20) DEFAULT NULL COMMENT '门店ID, store.id',
7      `product_id` bigint(20) DEFAULT NULL COMMENT '商品ID, product.id',
8      `num` bigint(20) DEFAULT 0 COMMENT '门店商品库存数量',
9      `creator` varchar(20) CHARACTER SET utf8 DEFAULT '' COMMENT '创建人',
10     `creator_id` bigint(20) DEFAULT NULL COMMENT '创建人ID',
11     `created_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
12     `last_operator` varchar(20) CHARACTER SET utf8 DEFAULT '' COMMENT '最近操作人',
13     `last_operator_id` bigint(20) DEFAULT NULL COMMENT '最后操作人ID',
14     `update_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '更新时间',
15     PRIMARY KEY (`id`),
16     UNIQUE KEY `uk_store_store_product` (`store_id`, `product_id`)
17 ) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=utf8 COMMENT='门店库存表';
```

4、generate生成对应的模块代码

4.1 创建提前中的Agent和store通用代码

修改hucais-generator-uac的generator中的配置文件

- 修改generator-A.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE generatorConfiguration
3     PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
4     "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
5
6 <generatorConfiguration>
7     <properties resource="config.properties" />
8
9     <context id="Mysql" targetRuntime="MyBatis3Simple"
10         defaultModelType="flat">
11         <property name="beginningDelimiter" value="`" />
12         <property name="endingDelimiter" value="`" />
13
14         <plugin type="${mapper.plugin}">
15             <property name="mappers" value="${mapper.Mapper}" />
16         </plugin>
17
18         <jdbcConnection driverClass="${jdbc.driverClass}"
19             connectionURL="${jdbc.url}" userId="${jdbc.user}"
20             password="${jdbc.password}">
21         </jdbcConnection>
22
23         <javaModelGenerator targetPackage="${targetModelPackage}"
24             targetProject="${targetJavaProject}" />
25
26         <sqlMapGenerator targetPackage="${targetXMLPackage}"
27             targetProject="${targetResourcesProject}" />
28
29         <javaClientGenerator targetPackage="${targetMapperPackage}"
30             targetProject="${targetJavaProject}" type="XMLMAPPER" />
31
32         <!-- 设置要生成代码的表 -->
33         <!--<table tableName="tbl_rmc_warehouse_area"
34             domainObjectName="RmcWarehouseArea"> -->
35             <!--<generatedKey column="id" sqlStatement="Mysql" identity="true"/> -->
36             <!--</table> -->
37             <!--<table tableName="tbl_rmc_device_detail"
38                 domainObjectName="RmcDeviceDetail"> -->
39                 <!--<generatedKey column="id" sqlStatement="Mysql" identity="true"/> -->
40                 <!--</table> -->
41
42         <table tableName="agent" domainObjectName="UacAgent">
```

```

42         <generatedKey column="id" sqlStatement="Mysql" identity="true" />
43     </table>
44
45     <table tableName="store" domainObjectName="UacStore">
46         <generatedKey column="id" sqlStatement="Mysql" identity="true" />
47     </table>
48
49
50 </context>
51 </generatorConfiguration>

```

- 修改generate-B.xml

- 修改为我自己本地的路径，比如我的项目路径：E:\workspace\hucais\hucais-generator\hucais-generator-uac

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE generatorConfiguration
3      PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
4      "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
5  <generatorConfiguration>
6  <properties resource="config.properties"/>
7  <!-- 数据库驱动 -->
8      <context id="DB2Tables" targetRuntime="MyBatis3">
9          <commentGenerator>
10             <property name="suppressDate" value="true"/>
11             <!-- 是否去除自动生成的注释 true:是 : false:否 -->
12             <property name="suppressAllComments" value="false"/>
13         </commentGenerator>
14         <!-- 数据库链接URL，用户名、密码 -->
15         <jdbcConnection driverClass="{jdbc.driverClass}"
16             connectionURL="{jdbc.url}" userId="{jdbc.user}" password="{jdbc.password}">
17         </jdbcConnection>
18         <javaTypeResolver>
19             <property name="forceBigDecimals" value="false"/>
20         </javaTypeResolver>
21         <!-- 生成模型的包名和位置 -->
22         <javaModelGenerator targetPackage="com.hucais.provider.model.domain"
23             targetProject="E:\workspace\hucais\hucais-generator\hucais-generator-
24             uac\src\main\java">
25             <property name="enableSubPackages" value="true"/>
26             <property name="trimStrings" value="true"/>
27         </javaModelGenerator>
28         <!-- 生成映射文件的包名和位置 -->
29         <sqlMapGenerator targetPackage="mapper"
30             targetProject="E:\workspace\hucais\hucais-generator\hucais-generator-
31             uac\src\main\resources">
32             <property name="enableSubPackages" value="true"/>
33         </sqlMapGenerator>
34         <!-- 生成DAO的包名和位置 -->
35         <javaClientGenerator type="XMLMAPPER"
36             targetPackage="com.hucais.provider.mapper" targetProject="E:\workspace\hucais\hucais-
37             generator\hucais-generator-uac\src\main\java">

```

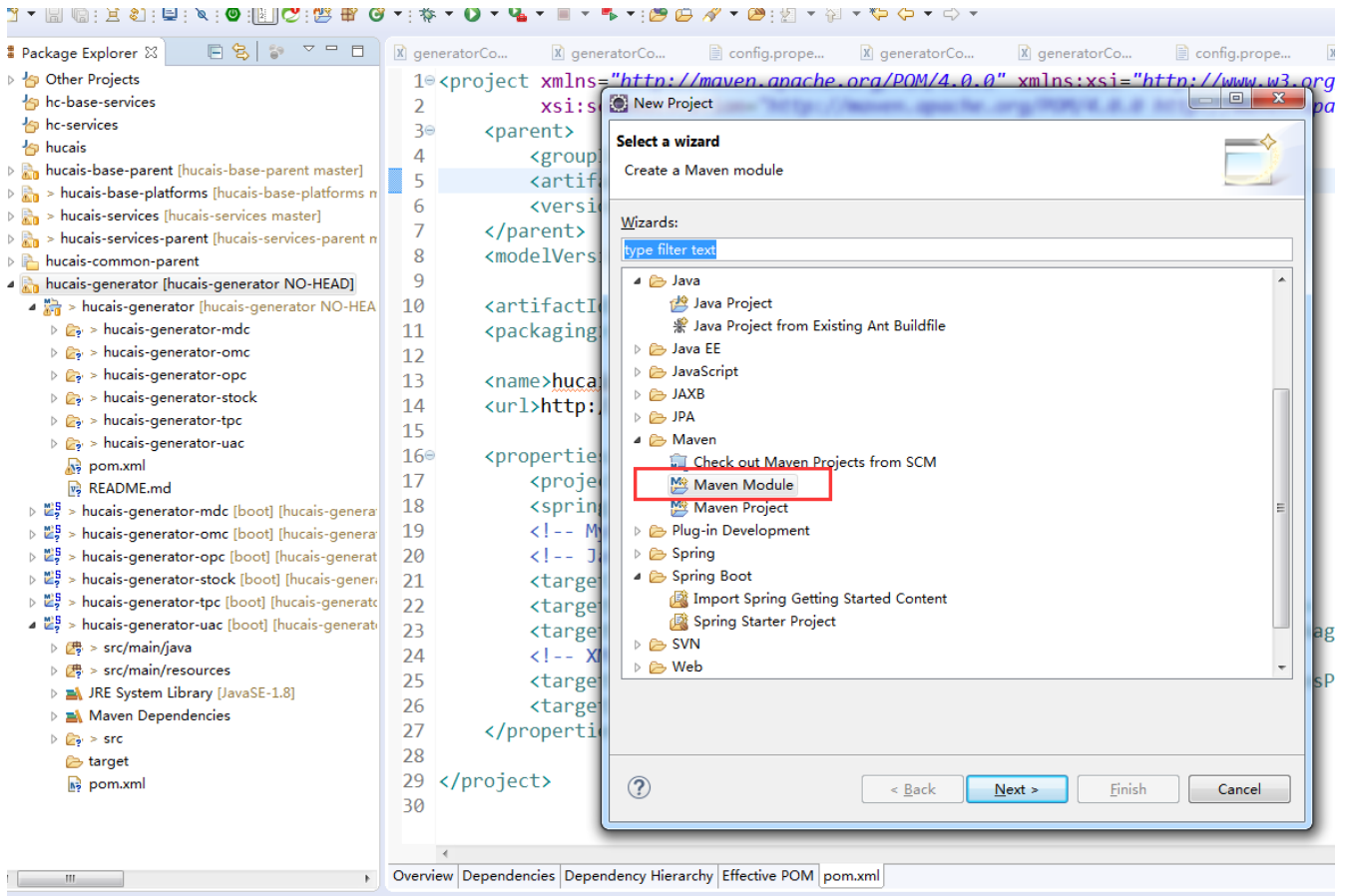
```

31         <property name="enableSubPackages" value="true"/>
32     </javaClientGenerator>
33     <!-- 要生成哪些表 -->
34     <!--
35     <table tableName="uac_action" domainObjectName="UacAction"
36         enableCountByExample="false" enableUpdateByExample="false"
37         enableDeleteByExample="false" enableSelectByExample="false"
38         selectByExampleQueryId="false"></table>
39
40     <table tableName="uac_group" domainObjectName="UacGroup"
41         enableCountByExample="false" enableUpdateByExample="false"
42         enableDeleteByExample="false" enableSelectByExample="false"
43         selectByExampleQueryId="false"></table>
44
45     <table tableName="uac_log" domainObjectName="UacLog"
46         enableCountByExample="false" enableUpdateByExample="false"
47         enableDeleteByExample="false" enableSelectByExample="false"
48         selectByExampleQueryId="false"></table>
49
50     <table tableName="uac_role" domainObjectName="UacRole"
51         enableCountByExample="false" enableUpdateByExample="false"
52         enableDeleteByExample="false" enableSelectByExample="false"
53         selectByExampleQueryId="false"></table>
54
55     <table tableName="uac_user_token" domainObjectName="UacUserToken"
56         enableCountByExample="false" enableUpdateByExample="false"
57         enableDeleteByExample="false" enableSelectByExample="false"
58         selectByExampleQueryId="false"></table>
59
60     <table tableName="mq_message_data" domainObjectName="MqMessageData"
61         enableCountByExample="false" enableUpdateByExample="false"
62         enableDeleteByExample="false" enableSelectByExample="false"
63         selectByExampleQueryId="false"></table>
64
65     -->
66
67     <table tableName="agent" domainObjectName="UacAgent"
68         enableCountByExample="false" enableUpdateByExample="false"
69         enableDeleteByExample="false" enableSelectByExample="false"
70         selectByExampleQueryId="false"></table>
71
72     <table tableName="store" domainObjectName="UacStore"
73         enableCountByExample="false" enableUpdateByExample="false"
74         enableDeleteByExample="false" enableSelectByExample="false"
75         selectByExampleQueryId="false"></table>
76
77 </context>
78 </generatorConfiguration>

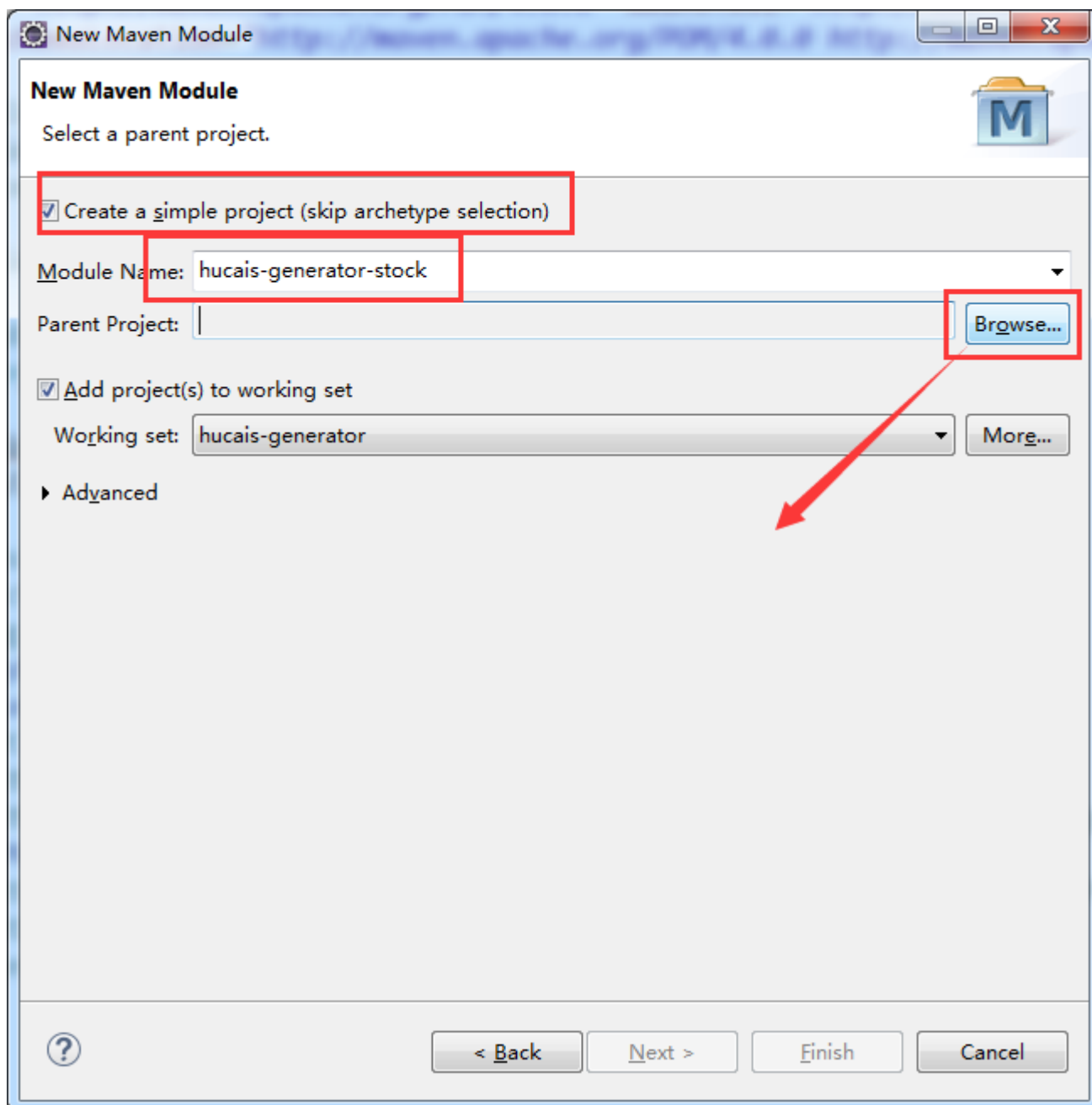
```

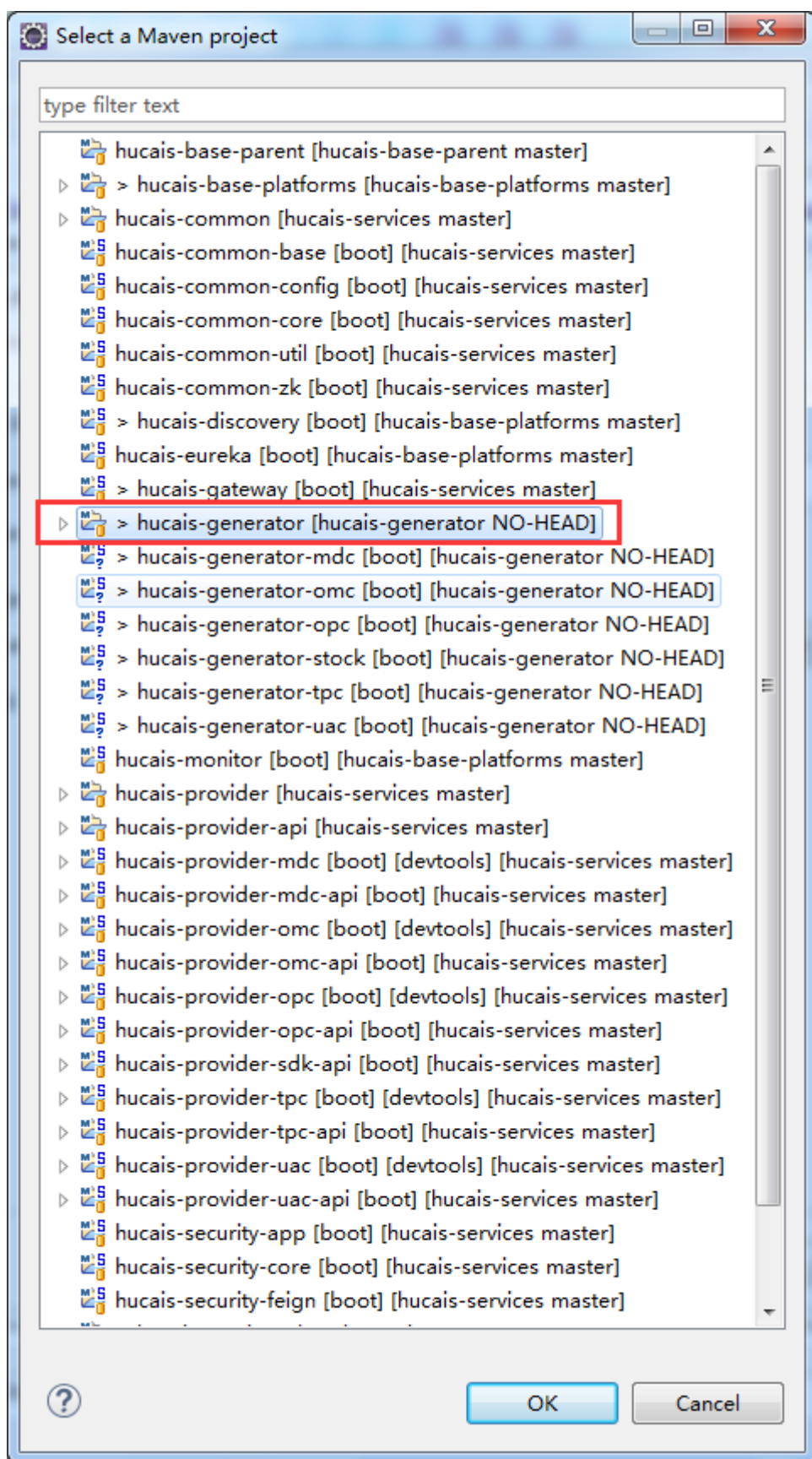
4.2 如果hucais-generator-stock项目不存在就新建项目

- 新建maven Module

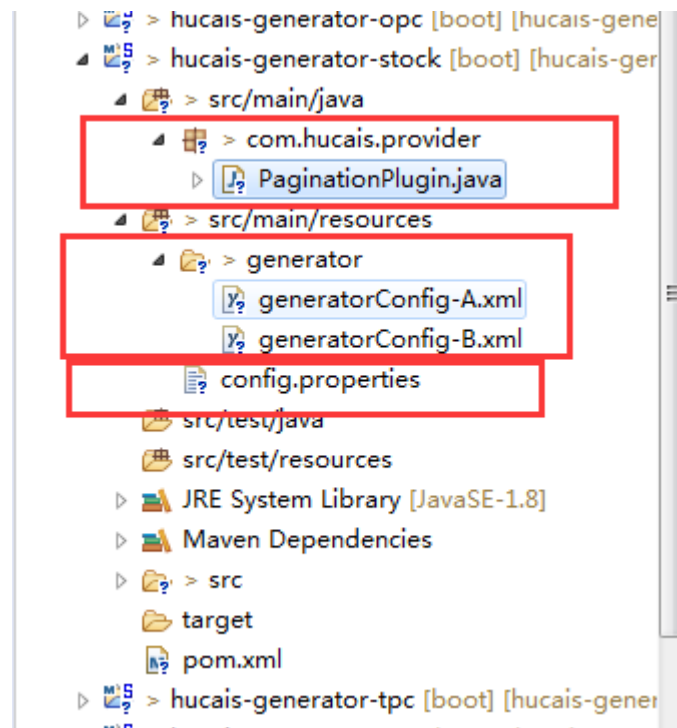


- 输入Module Name : hucais-generator-stock
- 点击Browse , 选择Parent Project :





- 点击Finish完成
- 参考其他generator代码生成项目，拷贝如下的文件夹和文件



- 然后修改config.properties配置文件

```
1 # \u6570\u636E\u5E93\u914D\u7F6E
2 jdbc.driverClass = com.mysql.jdbc.Driver
3 jdbc.url=jdbc:mysql://hucais-db-mysql:3306/hc_stock_platform?
  useUnicode=true&characterEncoding=UTF-8&useSSL=false
4 jdbc.user=root
5 jdbc.password=hucais2018
6
7 #c3p0
8 jdbc.maxPoolSize=50
9 jdbc.minPoolSize=10
10 jdbc.maxStatements=100
11 jdbc.testConnection=true
12
13 # \u901A\u7528Mapper\u914D\u7F6E
14 mapper.plugin = tk.mybatis.mapper.generator.MapperPlugin
15 mapper.Mapper = tk.mybatis.mapper.common.Mapper
```

- 修改generator-A.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE generatorConfiguration
3     PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
4     "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
5
6 <generatorConfiguration>
7     <properties resource="config.properties" />
8
9     <context id="Mysql" targetRuntime="MyBatis3Simple"
```

```

10     defaultModelType="flat">
11     <property name="beginningDelimiter" value="" />
12     <property name="endingDelimiter" value="" />
13
14     <plugin type="${mapper.plugin}">
15         <property name="mappers" value="${mapper.Mapper}" />
16     </plugin>
17
18     <jdbcConnection driverClass="${jdbc.driverClass}"
19         connectionURL="${jdbc.url}" userId="${jdbc.user}"
password="${jdbc.password}">
20     </jdbcConnection>
21
22     <javaModelGenerator targetPackage="${targetModelPackage}"
23         targetProject="${targetJavaProject}" />
24
25     <sqlMapGenerator targetPackage="${targetXMLPackage}"
26         targetProject="${targetResourcesProject}" />
27
28     <javaClientGenerator targetPackage="${targetMapperPackage}"
29         targetProject="${targetJavaProject}" type="XMLMAPPER" />
30
31     <!-- 设置要生成代码的表 -->
32     <!--<table tableName="tbl_rmc_warehouse_area"
domainObjectName="RmcWarehouseArea"> -->
33     <!--<generatedKey column="id" sqlStatement="Mysql" identity="true"/> -->
34     <!--</table> -->
35     <!--<table tableName="tbl_rmc_device_detail"
domainObjectName="RmcDeviceDetail"> -->
36     <!--<generatedKey column="id" sqlStatement="Mysql" identity="true"/> -->
37     <!--</table> -->
38
39     <table tableName="store_stock" domainObjectName="Storestock">
40         <generatedKey column="id" sqlStatement="Mysql" identity="true" />
41     </table>
42
43 </context>
44 </generatorConfiguration>

```

- 修改generator-B.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE generatorConfiguration
3      PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
4      "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
5  <generatorConfiguration>
6  <properties resource="config.properties"/>
7  <!-- 数据库驱动 -->
8      <context id="DB2Tables" targetRuntime="MyBatis3">
9          <commentGenerator>
10              <property name="suppressDate" value="true"/>
11              <!-- 是否去除自动生成的注释 true:是 : false:否 -->
12              <property name="suppressAllComments" value="false"/>

```

```

13         </commentGenerator>
14         <!-- 数据库链接URL, 用户名、密码 -->
15         <jdbcConnection driverClass="${jdbc.driverClass}"
connectionURL="${jdbc.url}" userId="${jdbc.user}" password="${jdbc.password}">
16         </jdbcConnection>
17         <javaTypeResolver>
18             <property name="forceBigDecimals" value="false"/>
19         </javaTypeResolver>
20         <!-- 生成模型的包名和位置 -->
21         <javaModelGenerator targetPackage="com.hucais.provider.model.domain"
targetProject="E:\workspace\hucais\hucais-generator\hucais-generator-
stock\src\main\java">
22             <property name="enableSubPackages" value="true"/>
23             <property name="trimStrings" value="true"/>
24         </javaModelGenerator>
25         <!-- 生成映射文件的包名和位置 -->
26         <sqlMapGenerator targetPackage="mapper"
targetProject="E:\workspace\hucais\hucais-generator\hucais-generator-
stock\src\main\resources">
27             <property name="enableSubPackages" value="true"/>
28         </sqlMapGenerator>
29         <!-- 生成DAO的包名和位置 -->
30         <javaClientGenerator type="XMLMAPPER"
targetPackage="com.hucais.provider.mapper" targetProject="E:\workspace\hucais\hucais-
generator\hucais-generator-stock\src\main\java">
31             <property name="enableSubPackages" value="true"/>
32         </javaClientGenerator>
33         <!-- 要生成哪些表 -->
34         <!--
35         <table tableName="uac_action" domainObjectName="UacAction"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false"></table>
36         <table tableName="uac_group" domainObjectName="UacGroup"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false"></table>
37         <table tableName="uac_log" domainObjectName="UacLog"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false"></table>
38         <table tableName="uac_role" domainObjectName="UacRole"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false"></table>
39
40         <table tableName="uac_user_token" domainObjectName="UacUserToken"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false"></table>

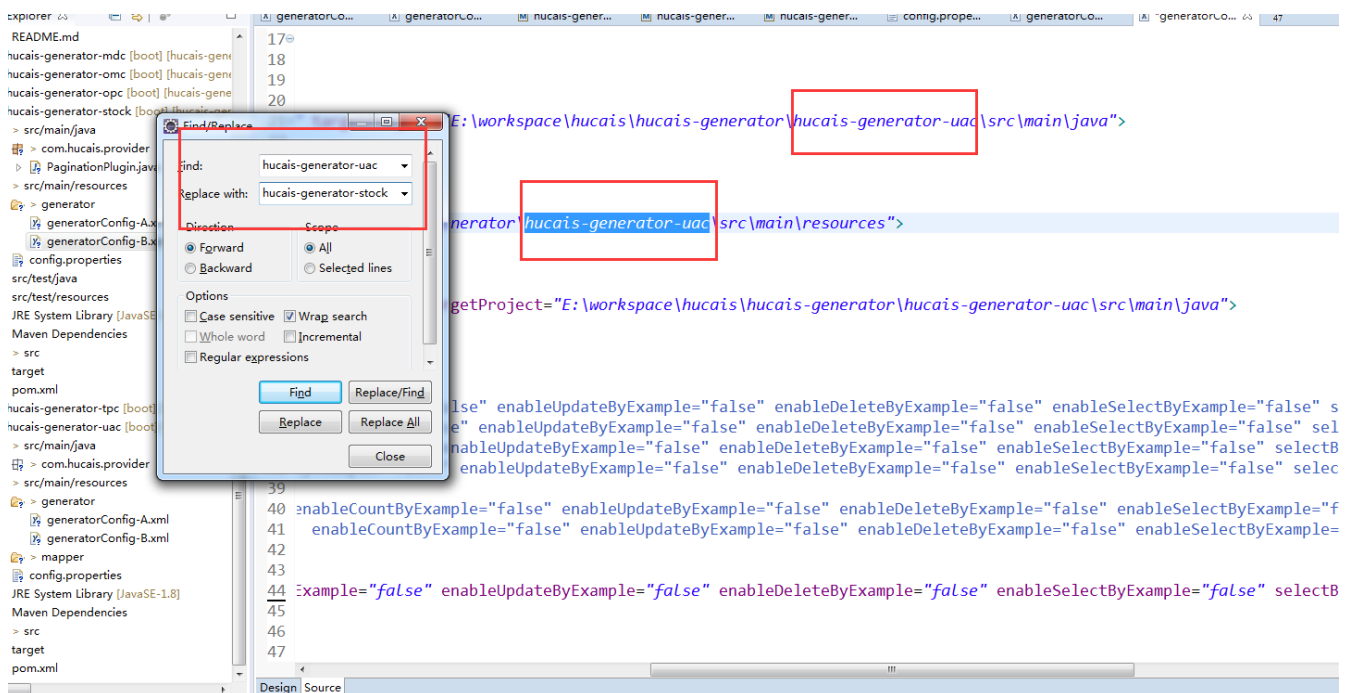
```

```

41      <table tableName="mq_message_data" domainObjectName="MqMessageData"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false"></table>
42  -->
43
44      <table tableName="store_stock" domainObjectName="Storestock"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false"></table>
45
46  </context>
47  </generatorConfiguration>

```

- 统一替换hucais-generator-uac为hucais-generator-stock

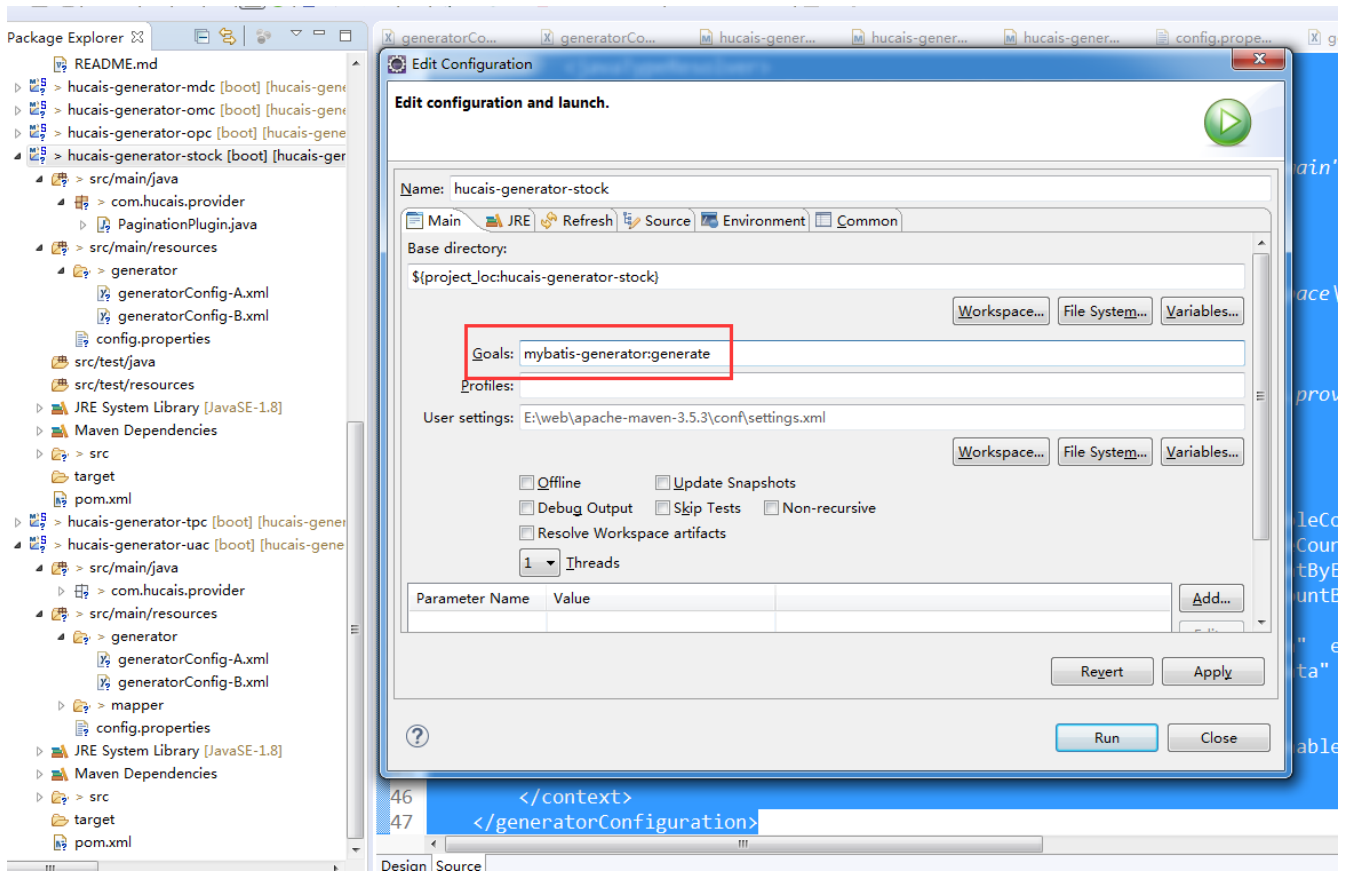


4.3 生成store_stock库存JAVA代码和Mapper.xml

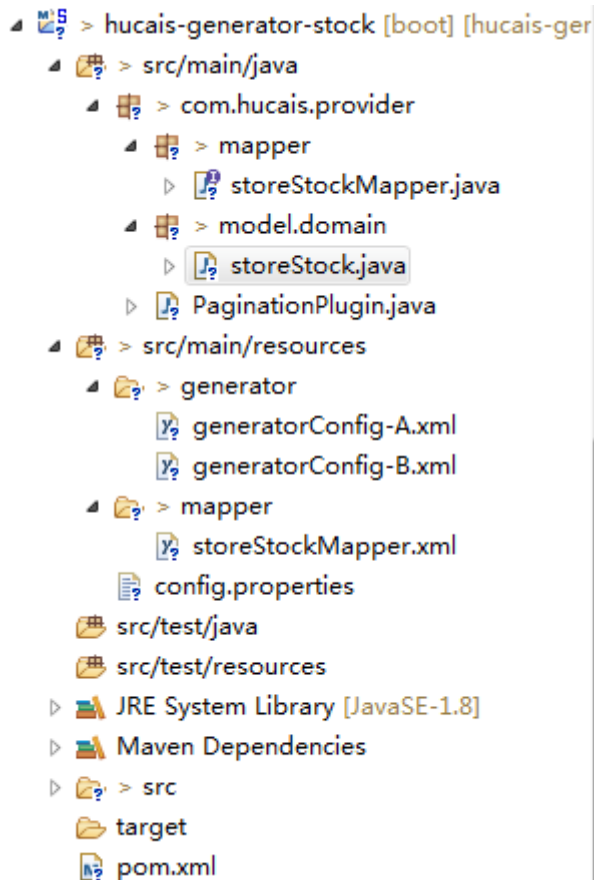
通过 Maven 插件集成的，所以运行插件使用下面的命令：

```
1 mvn mybatis-generator:generate
```

or 通过Eclipse的maven-->build



- 查看代码生成结构



5、编写接口文档 (showdoc.cc)

- 参考showdocc接口文档编写

7、创建库存微服务

- hucais-provider-stock微服务

6、编写mybatis/**Mapper.xml文件

- 参考其他mapper.xml增加新增修改删除方法

7、编写Mapper文件

- 修改Mapper添加**Mapper.xml文件中sql的对应方法

8、编写service接口和serviceImp实现

8.1 编写本地的service接口和方法

- 编写service方法，实现主要的业务逻辑
- 实现接口

8.2 编写远程调用的接口

- 比如hucais-provider-mdc服务的：`hucais-provider-mdc\src\main\java\com\hucais\provider\service\OpcRpcService.java`

```
1 package com.hucais.provider.service;
2
3
4 import com.hucais.provider.model.dto.oss.ElementImgUrlDto;
5 import com.hucais.provider.model.dto.oss.OptBatchGetUrlRequest;
6 import com.hucais.provider.model.dto.oss.OptGetUrlRequest;
7 import com.hucais.provider.model.dto.robot.ChatRobotMsgDto;
8 import com.hucais.wrapper.wrapper;
9 import org.springframework.stereotype.Component;
10
11 import javax.annotation.Resource;
12 import java.util.List;
13
14 /**
15  * The class Opc rpc service.
```



```

16  *
17  * @author hucais.com@gmail.com
18  */
19  @Component
20  public class OpcRpcService {
21
22      @Resource
23      private DingtalkFeignApi dingtalkFeignApi;
24      @Resource
25      private OpcOssFeignApi opcOssFeignApi;
26
27      public boolean sendChatRobotMsg(ChatRobotMsgDto chatRobotMsgDto) {
28          wrapper<Boolean> result = dingtalkFeignApi.sendChatRobotMsg(chatRobotMsgDto);
29          return result.getResult();
30      }
31
32      public List<ElementImgUrlDto> listFileUrl(OptBatchGetUrlRequest urlRequest) {
33          wrapper<List<ElementImgUrlDto>> result =
34          opcOssFeignApi.listFilesUrl(urlRequest);
35          return result.getResult();
36      }
37
38      public String getFileUrl(final OptGetUrlRequest request) {
39          wrapper<String> result = opcOssFeignApi.getFileUrl(request);
40          return result.getResult();
41      }
42  }

```

- 调用了FeignApi客户端：dingtalkFeignApi 和 opcOssFeignApi
- dingtalkFeignApi 和 opcOssFeignApi 都属于 hucais-provider-opc

9、编写controller

- 编写Controller接口，【springMVC restApi接口】

10、编写Feign客户端接口和Feign客户端实现

- 编写FeginApi接口
比如说：hucais-provider-opc服务的 opcOssFeignApi 接口

```

1  package com.hucais.provider.service;
2
3  import com.hucais.provider.model.dto.oss.*;
4  import com.hucais.provider.service.hystrix.OpcOssFeignApiHystrix;
5  import com.hucais.security.feign.OAuth2FeignAutoConfiguration;
6  import com.hucais.wrapper.wrapper;
7  import feign.codec.Encoder;
8  import feign.form.spring.SpringFormEncoder;
9  import org.springframework.cloud.netflix.feign.FeignClient;
10 import org.springframework.context.annotation.Bean;
11 import org.springframework.web.bind.annotation.PostMapping;

```

```

12 import org.springframework.web.bind.annotation.RequestBody;
13 import org.springframework.web.bind.annotation.RequestParam;
14 import org.springframework.web.multipart.MultipartFile;
15
16 import java.util.List;
17
18 /**
19  * The interface Opc oss feign api.
20  *
21  * @author hucais.com@gmail.com
22  */
23 @FeignClient(value = "hucais-provider-opc", configuration =
24 OAuth2FeignAutoConfiguration.class, fallback = OpcOssFeignApiHystrix.class)
25 public interface OpcOssFeignApi {
26
27     /**
28      * 上传文件.
29      *
30      * @param optUploadFileReqDto the opt upload file req dto
31      *
32      * @return the wrapper
33      */
34     @PostMapping(value = "/api/opc/oss/uploadFile")
35     wrapper<OptUploadFileRespDto> uploadFile(@RequestBody OptUploadFileReqDto
36 optUploadFileReqDto);
37
38     /**
39      * 获取附件完整路径.
40      *
41      * @param optGetUrlRequest the opt get url request
42      *
43      * @return the file url
44      */
45     @PostMapping(value = "/api/opc/oss/getFileUrl")
46     wrapper<String> getFileUrl(@RequestBody OptGetUrlRequest optGetUrlRequest);
47
48     /**
49      * List file url wrapper.
50      *
51      * @param urlRequest the url request
52      *
53      * @return the wrapper
54      */
55     @PostMapping(value = "/api/opc/oss/listFileUrl")
56     wrapper<List<ElementImgUrlDto>> listFileUrl(@RequestBody OptBatchGetUrlRequest
57 urlRequest);
58
59     /**
60      * Upload file with feign wrapper.
61      *
62      * @param file the file
63      *
64      * @return the wrapper

```

```

62     */
63     @PostMapping(value = "/api/opc/oss/handleFileUpload")
64     wrapper<OptUploadFileRespDto> handleFileUpload(@RequestPart(value = "file")
65     MultipartFile file);
66
67     /**
68     * 删除过期的文件.
69     */
70     @PostMapping(value = "/api/opc/oss/deleteExpireFile")
71     void deleteExpireFile();
72
73     /**
74     * The class Multipart support config.
75     *
76     * @author hucais.com @gmail.com
77     */
78     class MultipartSupportConfig {
79         /**
80         * Feign form encoder encoder.
81         *
82         * @return the encoder
83         */
84         @Bean
85         public Encoder feignFormEncoder() {
86             return new SpringFormEncoder();
87         }
88     }
89

```

- OpcOssFeignApi的实现客户端 OpcAttachmentFeignClient

```

1  - package com.hucais.provider.web.rpc;
2  import com.hucais.base.exception.BusinessException;
3  import com.hucais.core.support.BaseController;
4  import com.hucais.provider.exceptions.OpcBizException;
5  import com.hucais.provider.model.domain.OptAttachment;
6  import com.hucais.provider.model.dto.oss.*;
7  import com.hucais.provider.service.OpcAttachmentService;
8  import com.hucais.provider.service.OpcOssFeignApi;
9  import com.hucais.provider.service.OpcOssService;
10 import com.hucais.wrapper.WrapMapper;
11 import com.hucais.wrapper.Wrapper;
12 import com.qiniu.common.QiniuException;
13 import io.swagger.annotations.Api;
14 import io.swagger.annotations.ApiOperation;
15 import org.springframework.http.MediaType;
16 import org.springframework.web.bind.annotation.RequestBody;
17 import org.springframework.web.bind.annotation.RequestPart;
18 import org.springframework.web.bind.annotation.RestController;
19 import org.springframework.web.multipart.MultipartFile;
20 import javax.annotation.Resource;
21 import java.util.List;

```

```

22  /**
23  - The class Opc attachment feign client.
24  *
25  - @author hucais.com@gmail.com
26  */
27  @RestController
28  @Api(value = "API - OpcAttachmentFeignClient", produces =
MediaType.APPLICATION_JSON_UTF8_VALUE)
29  public class OpcAttachmentFeignClient extends BaseController implements
OpcOssFeignApi {
30      @Resource
31      private OpcAttachmentService opcAttachmentService;
32      @Resource
33      private OpcOssService opcOssService;
34      @Override
35      @ApiOperation(httpMethod = "POST", value = "上传文件")
36      public wrapper<OptUploadFileRespDto> uploadFile(@RequestBody OptUploadFileReqDto
optUploadFileReqDto) throws OpcBizException {
37          OptUploadFileRespDto result;
38          try {
39              logger.info("rpcUploadFile - RPC上传附件信息. optUploadFileReqDto={}",
optUploadFileReqDto);
40              result = opcAttachmentService.rpcUploadFile(optUploadFileReqDto);
41          } catch (BusinessException ex) {
42              logger.error("RPC上传附件信息, 出现异常={}", ex.getMessage(), ex);
43              return Wrapper.wrap(ex);
44          } catch (Exception e) {
45              logger.error("RPC上传附件信息, 出现异常={}", e.getMessage(), e);
46              return Wrapper.error();
47          }
48          return Wrapper.wrap(Wrapper.SUCCESS_CODE, "操作成功", result);
49      }
50      @Override
51      @ApiOperation(httpMethod = "POST", value = "获取附件完整路径")
52      public wrapper<String> getFileUrl(@RequestBody OptGetUrlRequest optGetUrlRequest)
{
53          String result;
54          try {
55              logger.info("getFileUrl - 获取附件完整路径. optGetUrlRequest={}",
optGetUrlRequest);
56              result = opcAttachmentService.rpcGetFileUrl(optGetUrlRequest);
57          } catch (BusinessException ex) {
58              logger.error("RPC获取附件完整路径, 出现异常={}", ex.getMessage(), ex);
59              return Wrapper.wrap(ex);
60          } catch (Exception e) {
61              logger.error("RPC获取附件完整路径, 出现异常={}", e.getMessage(), e);
62              return Wrapper.error();
63          }
64          return Wrapper.ok(result);
65      }
66      @Override
67      public wrapper<List<ElementImgUrlDto>> listFileUrl(@RequestBody
OptBatchGetUrlRequest urlRequest) {

```

```

68         logger.info("getFileUrl - 批量获取url链接. urlRequest={}", urlRequest);
69         List<ElementImgUrlDto> result = opcAttachmentService.listFilesUrl(urlRequest);
70         return wrapMapper.ok(result);
71     }
72     @Override
73     public wrapper<OptUploadFileRespDto> handleFileUpload(@RequestPart(value = "file")
MultipartFile file) {
74         return null;
75     }
76     @Override
77     public void deleteExpireFile() {
78         // 1.查询过期的文件
79         List<OptAttachment> list = opcAttachmentService.listExpireFile();
80         // 2.删除这些文件
81         for (final OptAttachment attachment : list) {
82             try {
83                 opcAttachmentService.deleteFile(attachment.getPath() +
attachment.getName(), attachment.getBucketName(), attachment.getId());
84             } catch (QiniuException e) {
85                 logger.info("删除文件失败, attachmentId={}", attachment.getId(), e);
86             }
87         }
88     }
89 }
90
91
92

```

11、添加Hystrix熔断器

- 编写OpcOssFeignApiHystrix

```

1  package com.hucais.provider.service.hystrix;
2
3  import com.hucais.provider.model.dto.oss.*;
4  import com.hucais.provider.service.OpcOssFeignApi;
5  import com.hucais.wrapper.Wrapper;
6  import org.springframework.stereotype.Component;
7  import org.springframework.web.multipart.MultipartFile;
8
9  import java.util.List;
10
11 /**
12  * The class Opc oss feign api hystrix.
13  *
14  * @author hucais.com@gmail.com
15  */
16 @Component
17 public class OpcOssFeignApiHystrix implements OpcOssFeignApi {
18     @Override
19     public wrapper<OptUploadFileRespDto> uploadFile(final OptUploadFileReqDto
optUploadFileReqDto) {

```

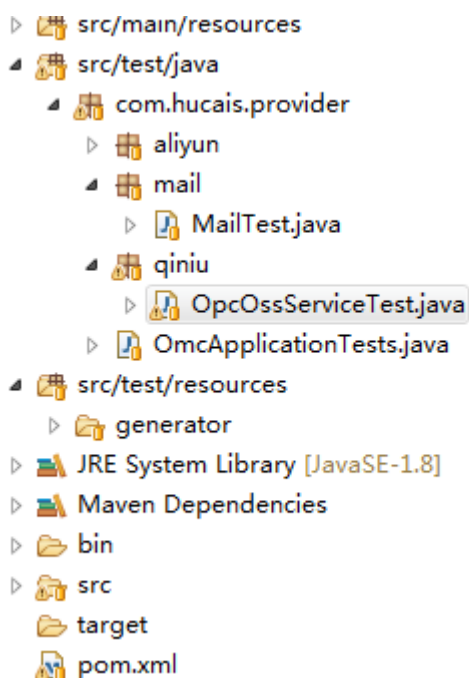
```

20         return null;
21     }
22
23     @Override
24     public wrapper<String> getFileUrl(final OptGetUrlRequest optGetUrlRequest) {
25         return null;
26     }
27
28     @Override
29     public wrapper<List<ElementImgUrlDto>> listFileUrl(final OptBatchGetUrlRequest
urlRequest) {
30         return null;
31     }
32
33     @Override
34     public wrapper<OptUploadFileRespDto> handleFileUpload(final MultipartFile file) {
35         return null;
36     }
37
38     @Override
39     public void deleteExpireFile() {
40
41     }
42 }
43

```

12、单元测试

- 编写单元测试



```
1 package com.hucais.provider.qiniu;
2
3 import com.hucais.config.properties.PaascloudProperties;
4 import com.hucais.provider.OmcApplicationTests;
5 import com.hucais.provider.service.OpcOssService;
6 import com.qiniu.common.QiniuException;
7 import org.junit.Test;
8
9 import javax.annotation.Resource;
10 import java.io.File;
11 import java.io.FileInputStream;
12 import java.io.IOException;
13
14 public class OpcOssServiceTest extends OmcApplicationTests {
15
16     @Resource
17     private OpcOssService optOssService;
18     @Resource
19     private PaascloudProperties hucaisProperties;
20
21     @Test
22     public void uploadFileTest() throws InterruptedException, IOException {
23
24         File file = new File("C:\\Users\\Lenovo\\Pictures\\love\\love6.jpg");
25
26         FileInputStream inputStream = new FileInputStream(file);
27
28         // DefaultPutRet defaultPutRet = optOssService.uploadFile(inputStream,
29         // "love6.jpg", "picture", "hucais-oss-bucket");
30         // logger.info("发送 生产数据 defaultPutRet={}", defaultPutRet);
31     }
32
33     @Test
34     public void deleteFileTest() throws QiniuException {
35
36         optOssService.deleteFile("picture/2017-08-
37         03/2a55cf5b4fab42d0984c7fa4c3d1f95f.jpg", "hucais-oss-bucket");
38         logger.info("删除成功");
39     }
40
41     @Test
42     public void batchDeleteFileTest() throws QiniuException {
43         String[] arr = new String[]{"love1.jpg", "image/jpg/love3.jpg"};
44
45         optOssService.batchDeleteFile(arr, "hucais-oss-bucket");
46         logger.info("删除成功");
47     }
48
49     @Test
50     public void getFileUrlTest() throws QiniuException {
```

```

51     String fileUrl =
    optOssService.getFileUrl(hucaisProperties.getQiniu().getOss().getPrivateHost(),
    "alipay/QRCode14e5d60fb30d4f7ea83ed3ec936ec54d.png", 3600L);
52     logger.info("获取File url成功 fileUrl={}", fileUrl);
53 }
54
55 @Test
56 public void getFileUrlTest2() throws QiniuException {
57
58     String fileUrl =
    optOssService.getFileUrl(hucaisProperties.getQiniu().getOss().getPrivateHost(),
    "alipay/QRCode/5be9c68b531f471bb85948a4954d5e6f.png");
59     logger.info("获取File url成功 fileUrl={}", fileUrl);
60 }
61 }
62 }

```

13、PostMan接口测试

- 编写PostMan接口调试

14、前端测试联调

14.1 首先修改hucais-gateway路由地址为本地服务地址

- 修改bootstrap.yml文件

```

1  spring:
2    cloud:
3      config:
4        fail-fast: true
5        discovery:
6          service-id: hucais-discovery
7          enabled: true
8          label: ${spring.profiles.active}
9          profile: ${spring.profiles.active}
10         username: admin
11         password: admin
12
13     eureka:
14       client:
15         serviceUrl:
16           defaultZone: http://root:root@hucais-eureka:8761/eureka/
17       instance:
18         instance-id:
19           ${spring.application.name}:${spring.cloud.client.ipAddress}:${server.port}
20         prefer-ip-address: true
21
22     zuul:
23       host:
24         socket-timeout-millis: 60000
25         connect-timeout-millis: 60000

```



```

25 forceOriginalQueryStringEncoding: true
26 routes:
27   uac-route:
28     path: /uac/**
29   #本地调试使用
30   #   url: http://127.0.0.1:8000
31     sensitiveHeaders:
32   #   strip-prefix: false
33     service-id: hucais-provider-uac
34   mdc-route:
35     path: /mdc/**
36   #   url: http://127.0.0.1:8030
37     sensitiveHeaders:
38     service-id: hucais-provider-mdc
39   omc-route:
40     path: /omc/**
41   #   url: http://127.0.0.1:8050
42     sensitiveHeaders:
43     service-id: hucais-provider-omc
44   opc-route:
45     path: /opc/**
46   #   url: http://127.0.0.1:8070
47     sensitiveHeaders:
48     service-id: hucais-provider-opc
49   tpc-route:
50     path: /tpc/**
51   #   url: http://127.0.0.1:8090
52     sensitiveHeaders:
53     service-id: hucais-provider-tpc
54   add-proxy-headers: true
55   add-host-header: true
56

```

- 修改你需要调试的微服务的url为本地的ip

14.2 前端开发人员修改HOST地址

- 比如你需要联调的开放人员的IP地址是192.172.21.56，修改Host

```

1 192.172.21.56 dev-api.hucais.com
2 192.172.21.56 api.hucais.com

```

15、更新代码并提交到develop分支

- git pull代码
- git commit
- git push origin develop

