Spring Cloud

用户购买商品的业务逻辑，这个业务逻辑由两个微服务提供支持：

1. 库存服务
2. 订单服务

1. 基础模块创建

1.1 创建表

- 执行如下SQL，创建表结构

```
CREATE TABLE `cloud_order` (
  `id` int(11) NOT NULL AUTO_INCREMENT comment '自增主键',
  `user_id` varchar(255) DEFAULT NULL comment '用户id',
  `product_code` varchar(255) DEFAULT NULL comment '商品编码',
  `count` int(11) DEFAULT 0 comment '商品数量',
  `money` int(11) DEFAULT 0 comment '商品金额',
  PRIMARY KEY (`id`)
) ;

CREATE TABLE `cloud_storage` (
  `id` int(11) NOT NULL AUTO_INCREMENT  comment '自增主键',
  `product_code` varchar(255) DEFAULT NULL comment '商品编码',
  `count` int(11) DEFAULT 0 comment '商品数量',
  PRIMARY KEY (`id`),
  UNIQUE KEY (`product_code`)
);
```

1.2 创建父工程（管理Maven依赖）：cloud-et2301

- 修改pom.xml，管理Maven依赖

```
<properties>
  <!-- 定义版本号 -->
  <cloud.version>2021.0.4</cloud.version>
  <cloud.alibaba.version>2021.0.4.0</cloud.alibaba.version>
  <mybatis-plus.version>3.4.3.1</mybatis-plus.version>
  <druid.version>1.2.11</druid.version>
  <hutool.version>5.8.0</hutool.version>
```

```xml
</properties>

<!-- Spring Boot -->
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.7.5</version>
</parent>

<!-- 管理Maven依赖 -->
<dependencyManagement>
  <dependencies>
    <!-- 管理Spring Cloud的依赖 -->
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>${cloud.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>

    <!-- 管理Spring Cloud Alibaba的依赖 -->
    <dependency>
      <groupId>com.alibaba.cloud</groupId>
      <artifactId>spring-cloud-alibaba-dependencies</artifactId>
      <version>${cloud.alibaba.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>

    <!-- mybatis-plus-boot-starter -->
    <dependency>
      <groupId>com.baomidou</groupId>
      <artifactId>mybatis-plus-boot-starter</artifactId>
      <version>${mybatis-plus.version}</version>
    </dependency>

    <!-- druid-spring-boot-starter -->
    <dependency>
      <groupId>com.alibaba</groupId>
      <artifactId>druid-spring-boot-starter</artifactId>
      <version>${druid.version}</version>
    </dependency>

    <!-- hutool-all -->
    <dependency>
      <groupId>cn.hutool</groupId>
      <artifactId>hutool-all</artifactId>
      <version>${hutool.version}</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

1.2 创建实体类模块：cloud-entity

1. 修改pom.xml文件，添加依赖

```xml
<dependencies>
  <dependency>
    <groupId>com.baomidou</groupId>
    <artifactId>mybatis-plus-boot-starter</artifactId>
    <optional>true</optional>
  </dependency>

  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <optional>true</optional>
  </dependency>
</dependencies>
```

2. 创建Order和Storage实体类

```java
@Data
@TableName("cloud_storage")
public class Storage {

  @TableId(type = IdType.AUTO)
  private Integer id;

  /** 商品编码 */
  private String productCode;

  /** 商品数量 */
  private Integer count;
}
```

```java
@Data
@TableName("cloud_order")
public class Order {

    @TableId(type = IdType.AUTO)
    private Integer id;

    /** 用户id */
    private String userId;

    /** 商品编码 */
    private String productCode;

    /** 商品数量 */
```

```
    private Integer count;

    /** 商品价格 */
    private Integer money;
}
```

1.3 创建common模块：cloud-common

1. 修改pom.xml文件，添加依赖

```
<dependencies>
  <!-- hutool-all -->
  <dependency>
    <groupId>cn.hutool</groupId>
    <artifactId>hutool-all</artifactId>
  </dependency>

  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <optional>true</optional>
  </dependency>
</dependencies>
```

2. 创建ResultVO

2. 开发库存服务：storage-service

1. 创建 `application.yml`
2. 创建启动类
3. 创建StorageMapper接口、StorageMapper.xml
4. 创建StorageService接口和StorageServiceImpl实现类
5. 创建StorageController（更新接口）

3. 开发订单服务：order-service

1. 创建 `application.yml`
2. 创建启动类
3. 创建OrderMapper接口、OrderMapper.xml
4. 创建OrderService接口和OrderServiceImpl实现类
5. 创建OrderController（添加接口）