Atitit 计算商品价格与税金

涉及到的技术点，steam api where 聚合函数

投影运算的udf

格式化展示

package attilax.tax;  
  
  
import org.apache.commons.lang3.StringUtils;  
  
   
import java.text.DecimalFormat;  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
import java.util.function.Consumer;  
import java.util.stream.Collectors;  
  
public class taxrateClsV3 {  
  
 public static void main(String[] args) {  
  
 System.*out*.println(StringUtils.*rightPad*("abc", 5) + "after");  
  
 Map input1 = new HashMap() {{  
 put("loc", "CA");  
 put("shoplist", new ArrayList<Map>() {{  
 add(new HashMap() {{  
 put("item", "book");  
 put("qty", 1);  
 put("price", 17.99);  
 }});  
 add(new HashMap() {{  
 put("item", "potato chips");  
 put("qty", 1);  
 put("price", 3.99);  
 }});  
 }});  
 }};  
  
//内存数据表LIST 进行投影运算 循环 获取税率，UDF计算单项税金，并聚合运算 计算总税  
 ((List) input1.get("shoplist")).forEach(new Consumer<Map>() {  
 @Override  
 public void accept(Map item) {  
 Object itemtype = *gettype*(item.get("item"));  
 Map item\_taxrate = *selectTaxrateFrom\_taxRateTable\_where\_loc\_and\_itemtype*(input1.get("loc"), itemtype);  
 item.put("itemtype", itemtype);  
 item.put("taxrate", item\_taxrate.get("tax\_rate\_num"));  
 item.put("taxrate\_strFormat", item\_taxrate.get("tax rate"));  
 item.put("item\_tax", *get\_item\_tax*((Integer) item.get("qty"), (double) item.get("price"), item\_taxrate.get("tax\_rate\_num")));  
 item.put("item\_total", (double) item.get("price") \* Double.*parseDouble*(item.get("qty").toString()));  
 // item.put("taxrate\_numFormat", Float.parseFloat(item\_taxrate) );  
 }  
 });  
  
 double all\_sale\_tax = ((List<Map>) input1.get("shoplist")).stream().mapToDouble(i -> (double) i.get("item\_tax")).sum();  
 DecimalFormat df2 = new DecimalFormat("###.00");//这样为保持2位  
 input1.put("tax", df2.format(all\_sale\_tax + 0.05));  
  
  
 double subtotal = ((List<Map>) input1.get("shoplist")).stream().mapToDouble(i -> (double) i.get("item\_total")).sum();  
 input1.put("subtotal", df2.format(subtotal + 0));  
 double total = Double.*parseDouble*(input1.get("tax").toString()) + Double.*parseDouble*(input1.get("subtotal").toString());  
 input1.put("total", df2.format(total));  
  
 System.*out*.println(input1);  
  
 *formatShow*(input1);  
 // calcTax(input1);  
 // System.out.println(gettype("potato chips"));  
  
 }  
  
 //格式化展示  
 private static void formatShow(Map input1) {  
 String tabs = "\t\t\t\t";  
 int col\_len = 22;  
  
 System.*out*.println("item" + StringUtils.*rightPad*("", col\_len - 4) + "price" + StringUtils.*rightPad*("", col\_len - 5) + "qty");  
 ((List) input1.get("shoplist")).forEach(new Consumer<Map>() {  
  
 @Override  
 public void accept(Map map) {  
  
 String itemPad = StringUtils.*rightPad*("", col\_len - map.get("item").toString().length());  
 String pricePad = StringUtils.*rightPad*("", col\_len - map.get("price").toString().length());  
 System.*out*.println(map.get("item") + itemPad + "$" + map.get("price") + pricePad + map.get("qty"));  
 }  
 });  
 String tabs2 = tabs + tabs;  
 int col\_len2 = 40;  
 System.*out*.println("subtotal:" + StringUtils.*rightPad*("", col\_len2 - 9) + "$" + input1.get("subtotal"));  
 System.*out*.println("tax:" + StringUtils.*rightPad*("", col\_len2 - 4) + "$" + input1.get("tax"));  
 System.*out*.println("total:" + StringUtils.*rightPad*("", col\_len2 - 6) + "$" + input1.get("total"));  
  
 }  
  
 //计算总税务  
 private static Object get\_item\_tax(Integer qty, double price, Object tax\_rate\_num) {  
 if (tax\_rate\_num == null)  
 return 0d;  
 return qty \* price \* (double) tax\_rate\_num;  
  
  
 }  
  
 //查询税率 数据表选择运算  
 private static Map selectTaxrateFrom\_taxRateTable\_where\_loc\_and\_itemtype(Object loc, Object itemtype) {  
  
  
 List<Map> result = *taxRateTable*.stream().filter(map\_item -> {  
  
 return loc.equals(map\_item.get("loc").toString()) && itemtype.equals(map\_item.get("type").toString());  
 // return true;  
  
 }).collect(Collectors.*toList*());  
 return result.get(0);  
  
 }  
  
 //查询物品类型  
 private static Object gettype(Object name) {  
 List<Map> itemTypetable = new ArrayList() {  
 {  
 //loc,sale,type(food)  
 add(new HashMap() {{  
 put("item", "potato chips");  
 put("type", "food");  
  
 }});  
 add(new HashMap() {{  
 put("item", "food222");  
 put("type", "food");  
  
 }});  
 }  
 };  
 List<Map> result = itemTypetable.stream().filter(map\_item -> {  
  
 return name.equals(map\_item.get("item").toString());  
 // return true;  
  
 }).collect(Collectors.*toList*());  
 if (result.size() == 0)  
 return "other";  
 return result.get(0).get("type");  
 }  
  
 private static List<Map> *taxRateTable* = new ArrayList() {{  
 //loc,sale,type(food)  
 add(new HashMap() {{  
 put("loc", "CA");  
 put("type", "other");  
 put("tax rate", "9.75%");  
 put("tax\_rate\_num", 0.0975);  
 }});  
 add(new HashMap() {{  
 put("loc", "CA");  
 put("type", "food");  
 put("tax rate", "0");  
 }});  
  
 add(new HashMap() {{  
 put("loc", "NY");  
 put("type", "other");  
 put("tax rate", " 8.875%");  
 put("tax\_rate\_num", 0.08875);  
 }});  
 add(new HashMap() {{  
 put("loc", "NY");  
 put("type", "food");  
 put("tax rate", "0");  
 }});  
 add(new HashMap() {{  
 put("loc", "NY");  
 put("type", "cloth");  
 put("tax rate", "0");  
 }});  
  
 }};  
  
}