

? ? ? ? ?

????????????

- ?SKU
- ???
- ???
- ?????
- ??
- ????
- ?????

?? ? ? ? ? ?

1. ?????
2. ?????????????????????? 1??
3. ???????Excel ? CSV??
4. ?????? SKU ??????????????????????
5. ????????????????

? ? ? ? ?

- ??? **SKU** ?????????????????????????????????
- ?????????????????????

?? ? ? ? ? ?

```
import os
import json
import pyexcel as pe
import requests
import re
from tkinter import Tk, filedialog, simpledialog
from datetime import datetime, timedelta
from collections import defaultdict, OrderedDict

# 窗口 tkinter 窗口
root = Tk()
root.withdraw()
```

```

# 0000000001
days_to_add = simpdialog.askinteger("000000", "0000000000001", initialValue=1)
if not days_to_add:
    days_to_add = 1

# 00000000 Excel 0 CSV0
file_path = filedialog.askopenfilename(
    title="00 Excel 0 CSV 00",
    filetypes=[("Excel/CSV Files", "*.xls *.xlsx *.csv")]
)
if not file_path:
    print("0 000000000000")
    exit()

# SKU 000
sku_json_path = r"\\nas-lianruey\office\sku\sku_ma.json"
if not os.path.isfile(sku_json_path):
    print(f"0 000 SKU 000000{sku_json_path}")
    exit()

with open(sku_json_path, "r", encoding="utf-8") as f:
    sku_map = json.load(f)

# 0000
plist_url = r'\\nas-lianruey\office\sku\plist.json'

try:
    with open(plist_url, "r", encoding="utf-8") as f:
        temp_data = json.load(f)
except FileNotFoundError:
    print(f"0 000000000000{plist_url}")
    exit()
except json.JSONDecodeError as e:
    print(f"0 000000000000{e}")
    exit()

sku_temp_map = {}
for item in temp_data:
    sku = item.get("sku")

```

```

temp = item.get("temp", "").strip() or ""
if sku:
    sku_temp_map[sku] = temp

# 读取
ext = os.path.splitext(file_path)[1].lower()
if ext == ".csv":
    records = pe.get_array(file_name=file_path, encoding="utf-8-sig")
else:
    records = pe.get_array(file_name=file_path)

next_date = (datetime.today() + timedelta(days=days_to_add)).strftime("%Y%m%d")
header = records[0]

# 初始化字典
try:
    order_index = header.index("订单号")
    item_index = header.index("商品名称")
    qty_index = header.index("数量")
    address_index = header.index("地址")
    name_index = header.index("名称")

    if "日期" not in header:
        header.append("日期")
        for i in range(1, len(records)):
            order_id = str(records[i][order_index])
            mo_id = order_id[:12] if len(order_id) >= 12 else order_id
            records[i].append(mo_id)
        header = records[0]

except ValueError as e:
    print("读取失败", e)
    exit()

# 写入
header += ["SKU", "日期", "数量", "地址", "名称", "日期"]

# 写入 log 文件
log_file_path = r"\\nas-lianruey\office\sku\app\sku_transfer_log.json"
if os.path.exists(log_file_path):

```

```

with open(log_file_path, "r", encoding="utf-8") as f:
    transfer_log = json.load(f)
else:
    transfer_log = {}

today_key = datetime.today().strftime("%Y%m%d")
today_prefix = f"M{today_key}"
start_seq = transfer_log.get(today_key, 0) + 1
transfer_seq_counter = start_seq
transfer_id_map = OrderedDict()

# 初始化
new_records = [header]
temp_by_order = defaultdict(list)

for row in records[1:]:
    row = row + [""] * (len(header) - 7 - len(row))
    product_id = str(row[item_index])
    try:
        order_qty = int(row[qty_index])
    except (ValueError, TypeError):
        order_qty = 1

    transfer_id = row[header.index("ID")]
    if transfer_id not in transfer_id_map:
        transfer_id_map[transfer_id] = f"{today_prefix}{transfer_seq_counter:03d}"
        transfer_seq_counter += 1

    address = str(row[address_index])
    address_cleaned = re.sub(r"[0-9\s]", "", address)[:3]
    name = str(row[name_index])
    remark = address_cleaned + name

    if product_id in sku_map:
        for sku in sku_map[product_id]:
            try:
                new_qty = int(sku["qty"]) * order_qty
            except (ValueError, TypeError):
                new_qty = sku["qty"]
            new_sku = sku["SKU"]

```

```

        temp_value = sku_temp_map.get(new_sku, "")
        temp_by_order[transfer_id].append(temp_value)
        new_row = row + [new_sku, new_qty, sku[""], next_date, temp_value, remark,
transfer_id_map[transfer_id]]
        new_records.append(new_row)
    else:
        temp_by_order[transfer_id].append("")
        new_records.append(row + ["", "", "", next_date, "", remark,
transfer_id_map[transfer_id]])

# 
final_records = [new_records[0]]
temp_index = header.index("")
transfer_index = header.index("")
remark_index = header.index("")
seen_transfer_ids = set()

for row in new_records[1:]:
    transfer_id = row[transfer_index]
    row[temp_index] = "" if all(t == "" for t in temp_by_order[transfer_id]) else ""
    if transfer_id in seen_transfer_ids:
        row[remark_index] = ""
    else:
        seen_transfer_ids.add(transfer_id)
    final_records.append(row)

# log
transfer_log[today_key] = transfer_seq_counter - 1
with open(log_file_path, "w", encoding="utf-8") as f:
    json.dump(transfer_log, f, ensure_ascii=False, indent=2)

# 
base_dir = os.path.dirname(file_path)
name_part, ext_part = os.path.splitext(os.path.basename(file_path))
timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
new_filename = f"{name_part}_SKU_{timestamp}.xls"
new_path = os.path.join(base_dir, new_filename)

pe.save_as(array=final_records, dest_file_name=new_path)
print(f" SKU {new_path}")

```

Revision #1

Created 1 August 2025 03:05:30 by Wayne

Updated 1 August 2025 03:17:19 by Wayne