??

? ?? ?? SKU Excel ???? ????

? ??

????????? **?? (Shopee) ?????????? Excel?**?
????? **SKU ??????????????????????** ? **???????**?
??????????????????

? ????

1. **?? Excel ??**
   - ?????? `365507` ???????????
2. **SKU ????**
   - ?? `sku_sp.json` ??????????????????? SKU?
   - ????????????
3. **????**
   - ?? `plist.json` ?????
     - ????????? ? ??????
     - ?? ? ????????
4. **??????**
   - ?????????? + ???????
   - ??????????????????????
5. **???????**
   - ??? `SYYYYMMDD001`
   - ???????????? `sku_transfer_log.json` ??????
6. **????**
   - ????????

```
□□□□_SKU□□_YYYYMMDD_HHMMSS.xls
```

? ????

- **????**
  - ???? Excel ???.xls ? .xlsx?
- **SKU ???**
  - `\\nas-lianruey\office\sku\sku_sp.json`
- **????**

    - `\\nas-lianruey\office\sku\plist.json`
- **?????**
    - `\\nas-lianruey\office\sku\app\sku_transfer_log.json`

? ????

??????????

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

?? ????

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

? ????

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

?? ?????

```
import os
import io
import json
import msoffcrypto
import pyexcel as pe
import requests
import re
from tkinter import Tk, filedialog
```

```
from datetime import datetime, timedelta
from collections import defaultdict, OrderedDict

# □□ tkinter □□□
root = Tk()
root.withdraw()

# □□□□ Excel
file_path = filedialog.askopenfilename(
    title="□□□□□ Excel □□",
    filetypes=[("Excel Files", "*.xls *.xlsx")]
)
if not file_path:
    print("□ □□□□□□□□□□□□□")
    exit()

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

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

# □□ SKU □□□□□□
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"□ □□□□□□□□□□{plist_url}")
    exit()
except json.JSONDecodeError as e:
    print(f"□ □□□□□□□□□□□{e}")
    exit()

sku_temp_map = {}
for item in temp_data:
```

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

# 解密 Excel
decrypted = io.BytesIO()
with open(file_path, "rb") as f:
    office_file = msoffcrypto.OfficeFile(f)
    office_file.load_key(password="365507")
    office_file.decrypt(decrypted)
decrypted.seek(0)

# 讀取為表格陣列
file_ext = os.path.splitext(file_path)[1].lower()
file_type = "xlsx" if file_ext == ".xlsx" else "xls"
records = pe.get_array(file_type=file_type, file_content=decrypted.read())

# 日期加一天
next_date = (datetime.today() + timedelta(days=1)).strftime("%Y%m%d")

# 新轉單編號 log 讀取
log_path = r"\\nas-lianruey\office\sku\app\sku_transfer_log.json"
if os.path.exists(log_path):
    with open(log_path, "r", encoding="utf-8") as f:
        transfer_log = json.load(f)
else:
    transfer_log = {}

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

# 找出欄位
header = records[0]
try:
    order_index = header.index("訂單編號")
    if "轉單編號" not in header:
        header.append("轉單編號")
```

```
        for i in range(1, len(records)):
            order_id = str(records[i][order_index])
            mo_id = "S" + order_id[:14] if len(order_id) >= 14 else "S" + order_id
            records[i].append(mo_id)
        header = records[0]
    transfer_index = header.index("轉單編號")
    item_index = header.index("商品選項貨號")
    qty_index = header.index("數量")
    address_index = header.index("收件地址")
    name_index = header.index("收件者姓名")
except ValueError as e:
    print("□ □□□□□□□□", e)
    exit()

# 加欄位
header += ["新SKU", "新數量", "新進價", "新預交貨日", "溫層", "分錄備註", "新轉單編號"]

# 展開資料列
new_records = [header]
temp_by_order_id = defaultdict(list)

for row in records[1:]:
    row = row + [""] * (len(header) - 7 - len(row))  # 填滿欄位
    product_id = str(row[item_index])
    mo_id = row[transfer_index]

    try:
        original_qty = int(row[qty_index])
    except (ValueError, TypeError):
        original_qty = 1

    # 建立分錄備註
    addr = str(row[address_index])
    addr_cleaned = re.sub(r"[0-9\s]", "", addr)[:3]
    name = str(row[name_index])
    remark = addr_cleaned + name

    # 建立新轉單編號
    if mo_id not in transfer_id_map:
        transfer_id_map[mo_id] = f"{prefix}{seq_counter:03d}"
        seq_counter += 1
```

```
    new_transfer_id = transfer_id_map[mo_id]

    if product_id in sku_map:
        for sku in sku_map[product_id]:
            try:
                new_qty = int(sku["新數量"]) * original_qty
            except:
                new_qty = ""
            new_sku = sku["新SKU"]
            temp = sku_temp_map.get(new_sku, "冷凍")
            temp_by_order_id[mo_id].append(temp)
            new_records.append(row + [new_sku, new_qty, sku["新進價"], next_date, temp, remark,
new_transfer_id])
    else:
        temp_by_order_id[mo_id].append("冷凍")
        new_records.append(row + ["", "", "", next_date, "冷凍", remark, new_transfer_id])

# 根據同轉單判斷是否全為常溫，且分錄備註只保留第一筆
final_records = [new_records[0]]
temp_index = header.index("溫層")
remark_index = header.index("分錄備註")
seen_transfer_ids = set()

for row in new_records[1:]:
    mo_id = row[transfer_index]
    row[temp_index] = "常溫" if all(t == "常溫" for t in temp_by_order_id[mo_id]) else "冷凍"

    if mo_id in seen_transfer_ids:
        row[remark_index] = ""
    else:
        seen_transfer_ids.add(mo_id)

    final_records.append(row)

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

# 輸出
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轉換_{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轉換、預交貨日與新轉單編號處理完成，檔案儲存為: {new_path}")
```

Revision #1
Created 1 August 2025 03:19:07 by Wayne
Updated 1 August 2025 03:22:20 by Wayne