???

? ??

??????? **??? (i) ????????Excel ? CSV?**????? SKU
??

? ????

1. **??????**
   - ???????????????? 1????????????
2. **SKU ????**
   - ?? `sku_i.json` ???????????????? SKU????????????
3. **????**
   - ?? `plist.json` ???????
     - ??????????? ? ???????
     - ?? ? ???????
   - ???? **P11-000-05** ? ????????
4. **??????**
   - ????????? + ????????????????
   - ??????????????????????
5. **???????**
   - ???`IYYYYMMDD001`
   - ???????????? `sku_transfer_log.json`??????
6. **????**
   - ????????

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

? ????

- **????**
  - ??? ?? Excel / CSV
- **SKU ???**
  - `\\nas-lianruey\office\sku\sku_i.json`
- **????**
  - `\\nas-lianruey\office\sku\plist.json`
- **?????**
  - `\\nas-lianruey\office\sku\app\sku_transfer_log.json`

? ????

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

- ?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

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

```
# 詢問加幾天，預設為1
days_to_add = simpledialog.askinteger("預交貨日設定", "請輸入要加幾天（預設為1）", initialvalue=1)
if not days_to_add:
    days_to_add = 1

# 選取檔案（支援 Excel 與 CSV）
file_path = filedialog.askopenfilename(
    title="選擇 Excel 或 CSV 檔案",
    filetypes=[("Excel/CSV Files", "*.xls *.xlsx *.csv")]
)
if not file_path:
    print("□ □□□□□□□□□□□□")
    exit()

# SKU 對照表路徑
sku_json_path = r"\\nas-lianruey\office\sku\sku_i.json"
if not os.path.isfile(sku_json_path):
    print(f"□ □□□ SKU 對照表檔案: {sku_json_path}")
    exit()

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

# 取得溫層資料
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

# 讀取資料
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)

# 刪除前3列，只留表頭與資料
if len(records) > 3:
    records = records[3:]

# 預交貨日
next_date = (datetime.today() + timedelta(days=days_to_add)).strftime("%Y%m%d")

# 讀取 log 紀錄: 新轉單編號流水號
log_file_path = r"\\nas-lianruey\office\sku\app\sku_transfer_log.json"
if os.path.isfile(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")
prefix = f"I{today_key}"
start_seq = transfer_log.get(today_key, 0) + 1
transfer_seq_counter = start_seq
transfer_id_map = OrderedDict()

# 找出欄位位置
header = records[0]
try:
    order_index = header.index("訂單編號")
    item_index = header.index("原廠商品編號")
    qty_index = header.index("數量")
    price_index = header.index("單價(含稅)")
    address_index = header.index("收貨地址")
    name_index = header.index("收件人")
```

```
    # 若無"轉單編號"欄位，補上
    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 = "I" + order_id[:7] if len(order_id) >= 7 else "I" + order_id
            records[i].append(mo_id)
        header = records[0]

except ValueError as e:
    print("□ □□□□□□□□", e)
    exit()

# 加入新欄位
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"]

# 處理資料列
new_records = [header]
temp_by_order = defaultdict(list)
transfer_index = header.index("轉單編號")

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
    mo_id = row[transfer_index]

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

    # 分錄備註
    address = str(row[address_index])
    address_clean = re.sub(r"[0-9\s]", "", address)[:3]
    recipient = str(row[name_index])
    raw_remark = address_clean + recipient
    remark = re.sub(r"[^\u4e00-\u9fffA-Za-z0-9]", "", raw_remark)
```

```
    if product_id in sku_map:
        for sku in sku_map[product_id]:
            try:
                new_qty = int(sku["新數量"]) * order_qty
            except (ValueError, TypeError):
                new_qty = sku["新數量"]
            new_sku = sku["新SKU"]
            new_temp = sku_temp_map.get(new_sku, "冷凍")
            if new_sku == "P11-000-05":
                new_temp = "常溫"
            temp_by_order[mo_id].append(new_temp)
            new_row = row + [new_sku, new_qty, sku["新進價"], next_date, new_temp, remark,
transfer_id_map[mo_id]]
            new_records.append(new_row)
    else:
        default_temp = "冷凍"
        temp_by_order[mo_id].append(default_temp)
        new_row = row + [
            product_id,
            order_qty,
            row[price_index] if price_index < len(row) else "",
            next_date,
            default_temp,
            remark,
            transfer_id_map[mo_id]
        ]
        new_records.append(new_row)

# 統一溫層與清除重複備註
final_records = [new_records[0]]
temp_col_index = header.index("溫層")
remark_col_index = header.index("分錄備註")
sku_col_index = header.index("新SKU")
seen_transfer_ids = set()

for row in new_records[1:]:
    mo_id = row[transfer_index]
    sku = row[sku_col_index]
    if sku == "P11-000-05":
        row[temp_col_index] = "常溫"
```

```
    else:
        row[temp_col_index] = "常溫" if all(t == "常溫" for t in temp_by_order[mo_id]) else "冷凍"

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

    final_records.append(row)

# 更新 log 檔
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)

# 儲存檔案
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、溫層與分錄備註、新轉單編號處理完成，檔案儲存為: {new_path}")
```

Revision #1
Created 1 August 2025 03:03:53 by Wayne
Updated 1 August 2025 03:05:23 by Wayne