????

? ??

????? **?? Excel ? CSV ??** ????????? **SKU ??????????????** ? **???????**????????

? ????

1. **??????**
   - ???????????????? 2????????????
2. **SKU ????**
   - ???SKU / ??? / ??????????????????????
3. **????**
   - ???????? `https://ec.zfun.com.tw/plist.json` ?? SKU ???
   - ???????????????? ????????????
4. **?????**
   - ??? `HYYYYMMDD001` ???????
   - ?????? `\\nas-lianruey\office\sku\app\sku_transfer_log.json` ??????
5. **??????**
   - ???????? + ??????
   - ????????????????????
6. **????**
   - ????????????????

```
□□□□_SKU□□_20250801_143000.xls
```

? ????

- **????**
  - Excel ? CSV ??????????????
- **??????**
  - `https://ec.zfun.com.tw/plist.json` ???? `wu` / ??? `wu2266228` ?
- **???????**
  - `\\nas-lianruey\office\sku\app\sku_transfer_log.json`

? ????

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

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

?? ????

1. ?????
2. ???????????
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

# 關閉 tkinter 主視窗
root = Tk()
root.withdraw()

# 詢問加幾天，預設為2
```

```
days_to_add = simpledialog.askinteger("預交貨日設定", "請輸入要加幾天（預設為2）", initialvalue=2)
if not days_to_add:
    days_to_add = 2

# 選取檔案
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()

# 讀取 SKU 溫層資料
plist_url = 'https://ec.zfun.com.tw/plist.json'
username = 'wu'
password = 'wu2266228'
try:
    response = requests.get(plist_url, auth=(username, password))
    response.raise_for_status()
    temp_data = response.json()
except requests.exceptions.RequestException as e:
    print("□ □□□□□□□□□", e)
    exit()

# 建立 SKU 對應溫層字典
sku_temp_map = {}
for item in temp_data:
    sku = str(item.get('sku', '')).strip()
    temp = item.get('temp', '').strip() or '冷凍'
    if sku:
        sku_temp_map[sku] = temp

# 根據副檔名讀取資料
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)
```

```
if not records or len(records) < 2:
    print("□ □□□□□□□□")
    exit()

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

# 找出必要欄位
header = records[0]
def idx(col):
    try:
        return header.index(col)
    except Exception:
        print(f"□ □□□□□□□{col}")
        exit()

order_index      = idx("訂單單號")
item_index       = idx("產品名稱")
qty_index        = idx("訂貨數量")
address_index    = idx("收貨地址")
name_index       = idx("收貨人姓名")
phone_index      = idx("收貨人電話")
vendor_sku_index = idx("廠商貨號")
unit_price_index = idx("單價(元)")
# 下方三個欄位若原本沒有也OK，會補上
new_sku_index    = header.index("新SKU")     if "新SKU"     in header else -1
new_qty_index    = header.index("新數量")    if "新數量"    in header else -1
new_price_index  = header.index("新進價")    if "新進價"    in header else -1

# 新增必要欄位
if "新SKU" not in header:
    header.append("新SKU")
    new_sku_index = len(header) - 1
if "新數量" not in header:
    header.append("新數量")
    new_qty_index = len(header) - 1
if "新進價" not in header:
    header.append("新進價")
    new_price_index = len(header) - 1
if "新預交貨日" not in header:
```

```
    header.append("新預交貨日")
if "溫層" not in header:
    header.append("溫層")
if "分錄備註" not in header:
    header.append("分錄備註")
if "新轉單編號" not in header:
    header.append("新轉單編號")

# 準備 LOG
log_path = r"\\nas-lianruey\office\sku\app\sku_transfer_log.json"
today_key = datetime.today().strftime("%Y%m%d")
today_prefix = f"H{today_key}"

if os.path.exists(log_path):
    with open(log_path, "r", encoding="utf-8") as f:
        log_data = json.load(f)
else:
    log_data = {}

seq_start = log_data.get(today_key, 0) + 1
transfer_seq_counter = seq_start
transfer_id_map = OrderedDict()

# 主處理
temp_by_order_id = defaultdict(list)
rows_by_order_id = defaultdict(list)
final_records = [header]

for row in records[1:]:
    row = row + [""] * (len(header) - len(row))  # 補足欄位

    # 基本資料
    vendor_sku = str(row[vendor_sku_index]).strip()
    order_id = str(row[order_index]).strip()
    mo_id = "H" + order_id[:13] if len(order_id) >= 13 else "H" + order_id
    mo_id = mo_id.strip()

    if mo_id not in transfer_id_map:
        transfer_id_map[mo_id] = f"{today_prefix}{transfer_seq_counter:03d}"
        transfer_seq_counter += 1
```

```
# 新SKU、新數量、新進價的自動填入
order_qty = str(row[qty_index]).strip()
unit_price = str(row[unit_price_index]).strip()

new_sku = str(row[new_sku_index]).strip() if new_sku_index >= 0 else ""
if not new_sku:
    new_sku = vendor_sku
    row[new_sku_index] = new_sku

new_qty = str(row[new_qty_index]).strip() if new_qty_index >= 0 else ""
if not new_qty:
    new_qty = order_qty
    row[new_qty_index] = new_qty

new_price = str(row[new_price_index]).strip() if new_price_index >= 0 else ""
if not new_price:
    new_price = unit_price
    row[new_price_index] = new_price

# 電話分拆
phone_raw = str(row[phone_index]).strip()
if "/" in phone_raw:
    new_phone, new_mobile = map(str.strip, phone_raw.split("/", 1))
else:
    new_phone, new_mobile = phone_raw, ""

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

# 溫層判斷
temp = sku_temp_map.get(new_sku, "冷凍")
print(f"[廠商貨號比對] 訂單號: {order_id}, 廠商貨號: {vendor_sku}, 新SKU: {new_sku}, 溫層: {temp}")

# 組成新 row
row_idx_map = {k: header.index(k) for k in header}
row_out = row.copy()
```

```
    row_out[row_idx_map["新SKU"]]     = new_sku
    row_out[row_idx_map["新數量"]]    = new_qty
    row_out[row_idx_map["新進價"]]    = new_price
    row_out[row_idx_map["新預交貨日"]]= next_date
    row_out[row_idx_map["溫層"]]      = temp
    row_out[row_idx_map["分錄備註"]]  = remark
    row_out[row_idx_map["新轉單編號"]]= transfer_id_map[mo_id]
    # 也可以視情況加入 new_phone、new_mobile 欄位

    temp_by_order_id[mo_id].append(temp)
    rows_by_order_id[mo_id].append(row_out)

# 決定 unified_temp
temp_index = header.index("溫層")
remark_index = header.index("分錄備註")
for mo_id, rows in rows_by_order_id.items():
    print(f"\n[訂單統計] mo_id: {mo_id}，此單所有SKU溫層: {temp_by_order_id[mo_id]}")
    unified_temp = "常溫" if all(t == "常溫" for t in temp_by_order_id[mo_id]) else "冷凍"
    print(f"[訂單統計] 統一判定此單 unified_temp = {unified_temp}")
    if unified_temp == "冷凍":
        print(f"[警告] 此訂單被判成冷凍，明細如下:")
        for t, row in zip(temp_by_order_id[mo_id], rows):
            print(f"    廠商貨號: {row[header.index('新SKU')]}, 溫層: {t}")
    for i, row in enumerate(rows):
        row[temp_index] = unified_temp
        if i > 0:
            row[remark_index] = ""
        final_records.append(row)

# 儲存 LOG
log_data[today_key] = transfer_seq_counter - 1
with open(log_path, "w", encoding="utf-8") as f:
    json.dump(log_data, 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"□ □□□□□□□□□□□□{new_path}")
```

Revision #1
Created 1 August 2025 02:58:15 by Wayne
Updated 1 August 2025 03:00:20 by Wayne