**功能規格書（wooRequestHelpers.py）**

**專案名稱：WooCommerce API 請求輔助工具**

**版本：** 1.0

**1. 簡介**

**WooRequestsHelper** 是一個輔助類別，主要用於透過 woocommerce Python SDK 與 WooCommerce API 進行通訊。

此類別提供標準的 **GET**、**POST**、**DELETE**、**PUT** 請求處理，並內建錯誤處理機制以確保 API 回應符合預期。

**2. 需求目標**

* 提供 WooCommerce API 的請求封裝，讓測試與開發更簡單。
* 允許開發人員透過 Python 簡單地發送請求，避免重複編寫請求邏輯。
* 內建 **狀態碼驗證**，確保 API 回應符合預期值，避免錯誤請求。
* 提供基本的 **日誌記錄 (Logging)**，方便偵錯與追蹤請求。

**3. 主要功能**

**3.1 初始化 WooCommerce API**

**功能描述**

* \_\_init\_\_() 方法會根據配置檔案 get\_config() 來初始化 WooCommerce API 客戶端。
* **請求憑證**（URL、消費者金鑰、消費者密鑰）均來自於 configurations.py 配置檔。

**輸入**

| **參數名稱** | **類型** | **說明** |
| --- | --- | --- |
| 無 | N/A | 直接從 get\_config() 讀取 API 配置資訊 |

**處理流程**

1. 透過 get\_config() 取得 WooCommerce API 的 **URL、消費者金鑰、消費者密鑰**。
2. 初始化 woocommerce.API 物件。
3. 設定 API 版本為 "wc/v3"。

**輸出**

| **參數名稱** | **類型** | **說明** |
| --- | --- | --- |
| self.wcapi | API | 已初始化的 WooCommerce API 物件 |

**3.2 發送 GET 請求**

**功能描述**

* get(wc\_endpoint, params=None, expected\_status\_code=200)
* 發送 **GET** 請求至指定的 WooCommerce API 端點。
* 檢查 API **回應狀態碼** 是否符合預期值，若不符則拋出錯誤。

**輸入**

| **參數名稱** | **類型** | **必填** | **預設值** | **說明** |
| --- | --- | --- | --- | --- |
| wc\_endpoint | str | ✅ | 無 | WooCommerce API 端點 |
| params | dict | ❌ | None | 可選的查詢參數 |
| expected\_status\_code | int | ❌ | 200 | 預期的 HTTP 狀態碼 |

**處理流程**

1. 發送 GET 請求至 wc\_endpoint，並附加 params（若有）。
2. **驗證狀態碼** 是否與 expected\_status\_code 相符。
3. 若不相符，則拋出 AssertionError，顯示 API 錯誤資訊。
4. 若成功，回傳 **JSON 格式的 API 回應**。

**輸出**

| **參數名稱** | **類型** | **說明** |
| --- | --- | --- |
| response\_json | dict | API 回應的 JSON 內容 |

**3.3 發送 POST 請求**

**功能描述**

* post(wc\_endpoint, params=None, expected\_status\_code=200)
* 發送 **POST** 請求至指定 WooCommerce API 端點，並附帶資料。
* 檢查 API **回應狀態碼** 是否符合預期值，若不符則拋出錯誤。

**輸入**

| **參數名稱** | **類型** | **必填** | **預設值** | **說明** |
| --- | --- | --- | --- | --- |
| wc\_endpoint | str | ✅ | 無 | WooCommerce API 端點 |
| params | dict | ❌ | None | 要傳送的資料 |
| expected\_status\_code | int | ❌ | 200 | 預期的 HTTP 狀態碼 |

**處理流程**

1. 記錄請求內容 params（方便日誌偵錯）。
2. 發送 POST 請求至 wc\_endpoint，並附帶 params（若有）。
3. **驗證狀態碼** 是否與 expected\_status\_code 相符。
4. 若不符，則拋出 AssertionError，顯示 API 錯誤資訊。
5. 若成功，回傳 **JSON 格式的 API 回應**。

**輸出**

| **參數名稱** | **類型** | **說明** |
| --- | --- | --- |
| response\_json | dict | API 回應的 JSON 內容 |

**3.4 狀態碼驗證**

**功能描述**

* assert\_status\_code()
* 確保 API **回應狀態碼** 與預期狀態碼相符，否則拋出錯誤。

**處理流程**

1. **比較回應狀態碼** 是否等於 expected\_status\_code。
2. 若不符，則拋出 AssertionError，詳細列出錯誤資訊：
   * API 端點 (wc\_endpoint)
   * 傳送的參數 (params)
   * 實際回應 (rs.status\_code)
   * 預期回應 (expected\_status\_code)
   * 伺服器回應內容 (rs.json())

**3.5 其他方法（未實作）**

| **方法名稱** | **功能** | **狀態** |
| --- | --- | --- |
| delete() | 用於刪除 WooCommerce 資源 | **未實作** |
| put() | 用於更新 WooCommerce 資源 | **未實作** |

**4. 系統整合**

**4.1 依賴關係**

| **模組名稱** | **用途** |
| --- | --- |
| woocommerce | WooCommerce API 客戶端 |
| BDDCommon.CommonConfigs.configurations | 讀取 API 配置 |
| logging | 記錄日誌資訊 |

**4.2 設定檔案（properties.ini）**

*[woo\_commerce]*

*url = http://mylocalwp.local/*

*username = admin*

*password = admin$1…*

*consumer\_key = ck\_b0…*

*consumer\_secret = cs\_71…*

**5. 測試案例**

**5.1 測試 GET 請求**

| **測試項目** | **測試內容** | **預期結果** |
| --- | --- | --- |
| 成功取得產品列表 | get("products") | 回傳 200 狀態碼與 JSON 內容 |
| 查詢不存在產品 | get("products/9999999") | 回傳 404 |

**5.2 測試 POST 請求**

| **測試項目** | **測試內容** | **預期結果** |
| --- | --- | --- |
| 註冊新用戶 | post("customers", params=payload, expected\_status\_code=201) | 回傳 201 |
| 重複註冊相同用戶 | post("customers", params=payload, expected\_status\_code=400) | 回傳 400 |

**6. 限制與未來改進**

* **目前不支援 DELETE & PUT 方法**（未實作）。
* **應該加入異常處理**，例如 **API 伺服器錯誤** (500) 應有適當錯誤回應。
* **需要支援 OAuth2 或 Token 驗證**，提高 API 安全性。

這是一份完整的 **功能規格書**，可以用於開發與測試。🚀