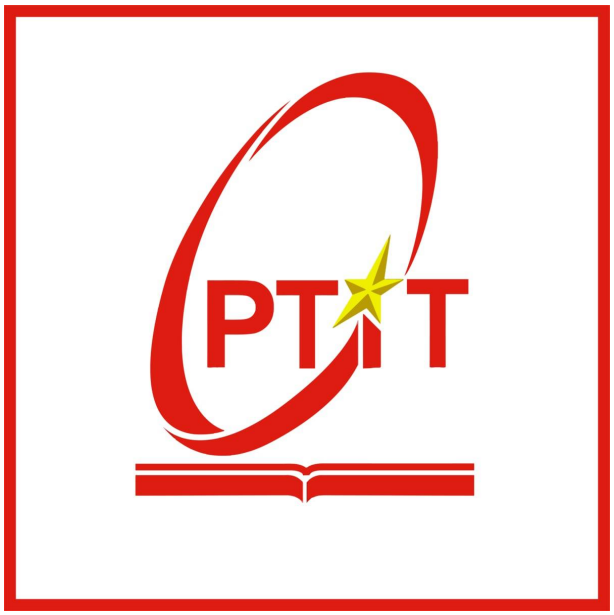
BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN I



BÁO CÁO BÀI TẬP LỚN

MÔN: LẬP TRÌNH PYTHON

Giảng viên hướng dẫn: Kim Ngọc Bách
Sinh viên: Nguyễn Việt Anh
Mã sinh viên: B23DCCE009
Lớp: D23CQCE06-B
Niên khóa: 2023 - 2028
Hệ đào tạo: Đại học chính quy

Hà Nội, Tháng 5/2025

Mục lục

Giới thiệu Tổng quan

Chương trình Python này được thiết kế để tự động thu thập (cào) dữ liệu thống kê của cầu thủ bóng đá từ trang web `fbref.com` (cụ thể là cho giải Ngoại hạng Anh mùa 2024-2025), sau đó xử lý, làm sạch, kết hợp dữ liệu từ nhiều bảng khác nhau và cuối cùng xuất ra một file CSV duy nhất.

---

1 Import các Thư viện Cần thiết

- `os` (được import dưới tên `operating_system`): Tương tác với hệ điều hành, cụ thể là tạo thư mục.
- `io.StringIO` (được import dưới tên `string_input_output`): Đọc chuỗi HTML như một file, cần thiết cho `pandas.read_html`.
- `time` (được import dưới tên `execution_timer`): Tạm dừng thực thi (`sleep`) và lấy thời gian hiện tại.
- `pandas` (được import dưới tên `data_manipulator`): Thư viện mạnh mẽ cho thao tác và phân tích dữ liệu, chủ yếu dùng DataFrame.
- `bs4.BeautifulSoup` (được import dưới tên `html_content_parser`): Phân tích cú pháp tài liệu HTML và trích xuất dữ liệu.
- `selenium.webdriver` (được import dưới tên `browser_automation`): Tự động hóa trình duyệt web (trong trường hợp này là Chrome).
- `selenium.webdriver.chrome.options.Options` (được import dưới tên `ChromeLaunchOptions`
  Cấu hình các tùy chọn khi khởi chạy Chrome (ví dụ: chạy ở chế độ `headless`).
- `selenium.webdriver.chrome.service.Service` (được import dưới tên `ChromeBrowserServic`
  Quản lý dịch vụ ChromeDriver.
- `webdriver_manager.chrome.ChromeDriverManager` (được import dưới tên `ChromeBrowserDri`
  Tự động tải và quản lý phiên bản ChromeDriver phù hợp.

---

2 Các Hàm Xử lý Dữ liệu Phụ trợ

2.1 `transform_age_string_to_numeric(age_text_representation)`

- Chuyển đổi chuỗi tuổi có định dạng "NĂM-NGÀY"(ví dụ: "30-150") thành một số thập phân (ví dụ: $30 + \frac{150}{365}$).
- Trả về "N/A"nếu có lỗi hoặc định dạng không hợp lệ.

2.2 sanitize_nationality_string(nationality_raw_string)

- Trích xuất mã quốc gia (thường là 3 chữ cái cuối, ví dụ: "ENG"từ "England ENG") từ chuỗi quốc tịch.
- Trả về "N/A"nếu có lỗi hoặc chuỗi rỗng.

---

3 Khởi tạo và Cấu hình Selenium WebDriver (setup_automat

- Hàm này thiết lập một trình duyệt Chrome tự động.
- `ChromeLaunchOptions`: Cấu hình để chạy trình duyệt ở chế độ `headless` (không có giao diện người dùng), tắt GPU, không dùng sandbox, giảm mức độ log, và một số tùy chọn khác để chạy ổn định hơn trong môi trường tự động.
- `ChromeBrowserDriverManager().install()`: Tự động tải về và cài đặt phiên bản ChromeDriver tương thích với trình duyệt Chrome hiện có trên máy.
- Trả về một đối tượng `driver` của Selenium, đại diện cho trình duyệt đã được khởi tạo.

---

4 Định nghĩa các Nguồn Dữ liệu (DATA_SCRAPING_TARGETS)

Đây là một danh sách (`list`) các từ điển (`dictionary`). Mỗi từ điển chứa:

- `source_url`: Đường link URL đến trang web chứa bảng dữ liệu cần lấy. Tất cả đều trỏ đến các trang thống kê khác nhau của giải Ngoại hạng Anh mùa 2024-2025 trên `fbref.com`.
- `html_table_id`: ID của thẻ `<table>` trong HTML của trang web đó, dùng để xác định chính xác bảng cần lấy dữ liệu.

---

5 Định nghĩa các Cột Dữ liệu Cần thiết và Ánh xạ Đổi tên

- `REQUIRED_DATA_FIELDS`: Danh sách tên các cột mà người dùng muốn có trong DataFrame cuối cùng.
- `COLUMN_HEADER_MAPPING_RULES`: Một từ điển lớn.
  - Mỗi khóa (`key`) của từ điển này là một `html_table_id` (tương ứng với các bảng sẽ được cào).
  - Giá trị (`value`) tương ứng là một từ điển khác, dùng để ánh xạ (đổi tên) các tên cột gốc (thường khó hiểu hoặc có ký tự đặc biệt, ví dụ: `Unnamed: 1`, `Performance.1`) từ bảng HTML thành các tên cột thân thiện và dễ hiểu hơn (ví dụ: `Player`, `Gls per 90`).

---

6 Định nghĩa Kiểu Dữ liệu cho các Cột sau khi Xử lý

- `INTEGER_TYPE_COLUMNS`: Danh sách các tên cột sẽ được chuyển đổi sang kiểu số nguyên (`Int64` - cho phép giá trị rỗng).
- `FLOAT_TYPE_COLUMNS`: Danh sách các tên cột sẽ được chuyển đổi sang kiểu số thực (`float`).

---

7 Hàm Chính Thực thi Việc Cào và Xử lý Dữ liệu

7.1 `extract_table_data_from_webpage(browser, page_url, table_html_id`

1. Sử dụng `browser.get(page_url)` để điều khiển trình duyệt đã khởi tạo truy cập vào `page_url`.
2. `execution_timer.sleep(3.2)`: Tạm dừng 3.2 giây để trang web có thời gian tải đầy đủ nội dung (đặc biệt là các nội dung động được JavaScript tải).
3. `browser.page_source`: Lấy toàn bộ mã HTML của trang sau khi đã tải xong.
4. `html_content_parser(web_page_source_code, "html.parser")`: Sử dụng `BeautifulSoup` để phân tích mã HTML này.
5. `parsed_html_document.find("table", {"id": table_html_id_attribute})`: Tìm thẻ `<table>` có `id` tương ứng.
6. Nếu không tìm thấy bảng, in cảnh báo và trả về `None`.
7. `data_manipulator.read_html(string_input_output(str(html_table_object)), header=0)` Sử dụng `pandas` để đọc trực tiếp bảng HTML (đã được chuyển thành chuỗi và đưa vào `StringIO`) thành một danh sách các DataFrame. `header=0` chỉ định dòng đầu tiên là header.
8. Lấy DataFrame đầu tiên từ danh sách (`list_of_dataframes[0]`).
9. `extracted_dataframe.rename(columns=rename_rules_for_table)`: Đổi tên các cột dựa trên `COLUMN_HEADER_MAPPING_RULES`.
10. `extracted_dataframe.loc[:, ~extracted_dataframe.columns.duplicated(keep='first')]` Loại bỏ các cột trùng lặp, giữ cột đầu tiên.
11. Nếu có cột `Age`, áp dụng hàm `transform_age_string_to_numeric` để chuẩn hóa dữ liệu tuổi.
12. Trả về DataFrame đã xử lý.

7.2 merge_multiple_dataframes(dictionary_of_dataframes)

- Nếu không có DataFrame nào trong `dictionary_of_dataframes`, sử dụng `data_manipulator.` để tạo một DataFrame rỗng.
- Nếu không, sử dụng `data_manipulator.merge` để gộp DataFrame hiện tại vào `aggregated_dataframe` dựa trên cột `Player` bằng phương thức `outer` merge (giữ lại tất cả các cầu thủ từ cả hai bảng).
- Xử lý các cột có thể bị trùng tên sau khi gộp bằng cách thêm hậu tố `_original`, `_new`, rồi loại bỏ các cột `_new` và đổi tên lại các cột `_original`.
- Trả về `aggregated_dataframe` đã được gộp.

7.3 perform_final_data_cleanup_and_formatting(master_player_datafram

- Nhận vào DataFrame tổng hợp.
- Chọn lại các cột theo `REQUIRED_DATA_FIELDS` để đảm bảo thứ tự và chỉ giữ lại các cột mong muốn.
- Lặp qua các cột:
  - Chuyển đổi kiểu dữ liệu của các cột sang số nguyên (`Int64`) hoặc số thực (`float`) theo định nghĩa trong `INTEGER_TYPE_COLUMNS` và `FLOAT_TYPE_COLUMNS`, sử dụng `errors="coerce"`.
  - Lọc bỏ những cầu thủ có số phút thi đấu (`Minutes`) ít hơn hoặc bằng 90 (nếu cột `Minutes` tồn tại và là kiểu số).
  - Chuẩn hóa cột `Nation` bằng hàm `sanitize_nationality_string`.
- Trả về DataFrame đã được làm sạch và định dạng cuối cùng.

7.4 export_dataframe_to_predetermined_csv_file(dataframe_to_save)

- Nhận vào DataFrame cuối cùng.
- Định nghĩa đường dẫn cố định để lưu file: `D:/python project/report/csv/result.csv`.
- `operating_system.makedirs(fixed_directory_path, exist_ok=True)`: Tạo thư mục `D:/python project/report/csv` nếu nó chưa tồn tại. `exist_ok=True` để không báo lỗi nếu thư mục đã có sẵn.
- `dataframe_to_save.to_csv(...)`: Lưu DataFrame ra file CSV tại đường dẫn trên, với các tùy chọn:
  - `index=False`: Không ghi chỉ số của DataFrame vào file.
  - `encoding="utf-8-sig"`: Dùng encoding UTF-8 với BOM (Byte Order Mark), giúp Excel mở file CSV có tiếng Việt tốt hơn.
  - `na_rep="N/A"`: Thay thế các giá trị `NaN` (rỗng) trong DataFrame bằng chuỗi "N/A"trong file CSV.
- In thông báo thành công hoặc lỗi.

---

8 Điểm vào Chính của Chương trình (`if __name__ == "__main__":`)

1. `web_browser_instance = setup_automated_browser_instance()`: Khởi tạo trình duyệt tự động.
2. `collected_dataframes_storage = {}`: Tạo một từ điển rỗng để lưu trữ các DataFrame cào được từ mỗi URL.
3. Lặp qua từng `target_config` trong `DATA_SCRAPING_TARGETS`:
   - In thông báo đang xử lý target nào.
   - Gọi `extract_table_data_from_webpage` để lấy dữ liệu từ URL và ID bảng tương ứng.
   - Nếu lấy được DataFrame và DataFrame đó không rỗng, lưu nó vào `collected_dataframes_st` với `key` là `html_table_id`.
4. Kiểm tra xem có dữ liệu nào được cào không (`collected_dataframes_storage` có rỗng không):
   - Nếu có, gọi `merge_multiple_dataframes` để gộp tất cả các DataFrame đã thu thập được.
   - Nếu không, in cảnh báo.
5. Gọi `perform_final_data_cleanup_and_formatting` để làm sạch và định dạng DataFrame tổng hợp.
6. Gọi `export_dataframe_to_predetermined_csv_file` để lưu DataFrame cuối cùng ra file CSV.
7. `web_browser_instance.quit()`: Đóng trình duyệt tự động.
8. In thông báo hoàn tất chương trình.

Tóm lại Quy trình ETL

Chương trình thực hiện một quy trình ETL (Extract, Transform, Load) đơn giản:

- **Extract (Trích xuất)**: Lấy dữ liệu thô từ nhiều bảng trên trang web `fbref.com` bằng `Selenium` và `BeautifulSoup`.
- **Transform (Biến đổi)**: Làm sạch dữ liệu (đổi tên cột, chuẩn hóa tuổi, quốc tịch), chuyển đổi kiểu dữ liệu, gộp các bảng dữ liệu lại với nhau, và lọc theo một số điều kiện.
- **Load (Tải)**: Lưu trữ dữ liệu đã xử lý vào một file CSV tại một đường dẫn cố định.