**Country Heads Web Scraping**

This project is dedicated to extracting and organizing comprehensive data about country heads and key government officials across various nations. The data was gathered using web scraping techniques in Python and has been structured to ensure accuracy and consistency.

Below is a summary of the fields included in the final dataset.

**Fields in the final dataset**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **ISO2CODE** | The ISO 3166-1 alpha-2 code for the country |
| **Country\_Head\_Name\_EN** | The name of the country in English |
| **Country\_Head\_Name\_AR** | The name of the country in Arabic |
| **Designation\_EN** | The official title or position (e.g., President, King, Interior Minister, Defense Minister) in English. |
| **Person\_Name\_EN** | The name of the individual holding the position in English |
| **Person\_Name\_AR** | The name of the individual holding the position in Arabic. |
| **Assumed\_Office\_Date** | The date on which the individual assumed office. |
| **Image\_URL** | The URL linking to the image of the individual |
| **ID** | A unique identifier for each country |
| **Designation\_ID** | A unique identifier for each official title or position |
| **Person\_ID** | A unique identifier for each individual holding a position |

Notes:

* Some of these fields were collected from Wikipedia pages, while others were created as unique identifiers to ensure data integrity.
* Not all Arabic names are included due to the absence of Arabic versions on Wikipedia for some entries.

**Libraries Used**

This project utilizes several Python libraries to handle various tasks such as web scraping, data manipulation, and URL encoding. Below is a brief overview of each library and its role in the project:

|  |  |
| --- | --- |
| **Library** | **Description** |
| **urllib.parse** | Specifically, the `quote` function is used to encode URLs by converting special characters into percent-encoded formats, ensuring that the URLs are properly formatted for web requests. |
| **Pandas** (`**pd**`) | A powerful data manipulation library used to create and manage DataFrames. It plays a crucial role in organizing, analyzing, and exporting the collected data. |
| **re** | The regular expression library is used for string manipulation, allowing for pattern matching and text extraction tasks essential for cleaning and processing data. |
| **requests** | A simple and elegant HTTP library for making web requests. It is used to send requests to web pages and retrieve the HTML content needed for data extraction. |
| **BeautifulSoup** from `**bs4**` | A library for parsing HTML and XML documents. It is used to navigate and extract data from the HTML content retrieved by `requests`, making the web scraping process more manageable. |

**Main Functions Used**

This section outlines the key functions used in the project, including their inputs and outputs. All URLs used are from Wikipedia:

|  |  |  |
| --- | --- | --- |
| **Function** | **Input** | **Output** |
| **get\_country\_data** | A Wikipedia URL containing country ISO codes. | Retrieves the country's ISO2CODE, the name of the country in English (`Country\_Head\_Name\_EN`), and the URL for the country (`Country\_URL`). |
| **get\_arabic\_wikipedia\_link** | The English Wikipedia link for a person or a country. | Obtains the corresponding Arabic Wikipedia link. |
| **get\_country\_name\_arabic** | The Arabic Wikipedia URL for a person or country. | Retrieves the Arabic name of the country or person. |
| **fetch\_government\_data** | The Wikipedia English URL for the country. | Collects government-related data, including the official title in English (`Designation\_EN`) and the name of the person in English (`Person\_Name\_EN`). |
| **get\_person\_data\_main** | The Wikipedia URL for the person’s page. | Retrieves the assumed office date (`Assumed\_Office\_Date`) and the image URL (`Image\_URL`) for the person. |
| **extract\_defense\_ministers\_from\_table** | The Wikipedia link for defense ministers and the index of the relevant table. | Extracts data about defense ministers, including the country name (`State`), the name of the defense minister (`Defense\_Minister\_Name`), and the URL for the defense minister page (`Defense\_Minister\_url`). |
| **get\_person\_minister\_data** | The Wikipedia URL for the defense/interior minister’s page. | Retrieves the image link for the defense/interior minister. |
| **extract\_interior\_ministers\_from\_table** | The Wikipedia link for interior ministers and the index of the relevant table. | Extracts data about interior ministers, including the country name (`State`), the name of the interior minister (`Interior\_Minister\_Name`), and the URL for the interior minister (`Interior\_Minister\_url`). |

**Transformations Functions Used**

This section describes the auxiliary functions used in the project, which assist in data processing and manipulation:

|  |  |
| --- | --- |
| **Function** | **Purpose** |
| **assign\_country\_url** | Updates certain country URLs in the dataset as required by the team. |
| **expand\_data** | Appends the `Person\_Name\_EN` list and `Person\_URL` list to the DataFrame, expanding it with additional rows for each person. |
| **assign\_designation\_id** | Adds a `Designation\_ID` column to the DataFrame, assigning unique identifiers to each official designation. |
| **has\_date** | Checks if the `Assumed\_Office\_Date` is a valid date, ensuring that the date data is properly formatted and accurate. |
| **aggregate\_designations** | Merges duplicated names into a single row. For example, if a person holds multiple titles (e.g., "President: X, Defense Minister: X"), they will be combined into one record (e.g., "President/Defense Minister: X"). |
| **add\_defense\_ministers** | Iterates over the defense ministers' data and appends matching records to the DataFrame. Specifically:  It loops through the data of defense ministers and checks for any matches between the `State` (from the first DataFrame) and the `Country\_Head\_Name\_EN` (from the second DataFrame). If a match is found, the corresponding defense minister data is appended to the new DataFrame.  The second DataFrame is then combined with the appended data from the new. The combined DataFrame is sorted by `ID` and `Designation\_ID`, and finally saved to an Excel sheet. |

**Workflow Diagram**

The diagram illustrates the workflow of the project, depicting how data is processed and integrated. Each step is represented by a node, with arrows showing the flow of data between them.

* Blue: Indicates fields that are included in the final data.
* Orange: Represents fields that are excluded from the final data.

