**Egyptian Customs Tariff Web Scraper ETL Documentation**

This Python script scrapes tariff data from two Egyptian government websites and combines the results into a comprehensive dataset. The script consists of two main scraping functions and a data processing pipeline.

1. . **scrape\_customs\_data() Function**

**Purpose:**

Scrapes HS code items from the Egyptian Customs Authority website (customs.gov.eg).

Functionality:

* Iterates through chapters 1-99 and pages 1-100 for each chapter
* Constructs URLs using base\_url with page and chapter parameters
* Makes HTTP requests to each URL and parses the HTML response
* Finds tariff data tables with class='table'
* Extracts item codes from table cells, removing any forward slashes
* Stops scraping a chapter when no more data is found (empty table or no rows)
* Returns a pandas DataFrame containing:
* Chapter number
* Page number
* Item code (HS code)

**Parameters:** None

**Returns:**

A pandas DataFrame with the scraped customs data

1. **scrape\_fei\_data(hscode) Function**

**Purpose:**

Scrapes detailed tariff information for specific HS codes from the FEI website (fei.org.eg).

Functionality:

* Takes an HS code as input
* Constructs the target URL with the HS code parameter
* Makes HTTP request and parses HTML response
* Extracts multiple data fields using safe extraction methods:
* Description
* Unit
* Custom Fee
* VAT
* Agreement details (with special processing)
* Implements robust error handling for each field
* Processes agreement details to keep only Egypt-related content
* Returns a dictionary with all extracted fields

**Parameters:**

* hscode: The HS code to look up (string)

**Returns:**

A dictionary containing all scraped tariff details for the HS code

1. Main Execution Flow

**Processing Steps:**

1. Calls scrape\_customs\_data() to get initial HS code list
2. For each unique HS code, calls scrape\_fei\_data() to get details
3. Combines both datasets into a final DataFrame
4. Adds quotes around Item values to preserve leading zeros in Excel
5. Saves the final dataset to CSV
6. Prints progress updates and sample data

**Output Data Insights:**

* Egypt maintains 8,795 unique HS codes for tariff/VAT/agreement classification
* As of 2025, only 3,143 HS codes have been updated with published tariff/VAT/agreement information
* The remaining 5,652 HS codes await official publication by Egyptian customs
* The code is designed for seamless future reruns to update the output file when new HS code information becomes available
* In the output file, 'N/A' indicates data that is 'Not Available' on the official websites.

**Data Processing Features:**

* Combines data from two different government sources
* Handles pagination and chapter-wise scraping
* Implements comprehensive error handling
* Processes agreement details to focus on Egypt-specific content
* Formats output for Excel compatibility

**Output File:**

* egyptian\_tariff\_data\_quoted\_again.csv
* Contains all scraped fields with quoted Item values

**Error Handling:**

* Both functions include try-except blocks
* Returns default 'N/A' values for missing fields
* Prints error messages while continuing execution
* Uses timeout for HTTP requests

**Performance Notes:**

* Processes data in batches (prints progress every 10 HS codes)
* Skips empty pages/chapters efficiently
* Uses BeautifulSoup for reliable HTML parsing

Usage Instructions:

1. Run the script directly
2. The final output will be saved as a CSV file
3. Progress is printed to console
4. Sample data is displayed at the end

**Dependencies:**

* requests (for HTTP requests)
* BeautifulSoup (for HTML parsing)
* pandas (for data manipulation)
* re (for regular expressions)

***Note: The script is designed to handle the specific HTML structure of the target websites as of the implementation date. Changes to the website structure may require updates to the script logic.***