Introduction

This Python script utilizes Selenium and BeautifulSoup to scrape content from the Singapore Civil Defence Force (SCDF) Fire Code website. The script recursively collects links from the main page and its sub-pages, saves the HTML content of each page, and organizes them into a directory.

Setup

Installing Dependencies

Make sure you have the necessary dependencies installed:

```
pip install selenium beautifulsoup4
```

Web Driver Configuration

Download the ChromeDriver executable and set the **chrome_driver_path** variable in the script to the correct path.

Script Overview

The script consists of several functions:

- **get_relative_links(driver, url, exclude_keywords=[])**: Retrieves relative links from a given page, excluding specified keywords.
- **get_links_recursive(driver, url, exclude_keywords=[], links=[])**: Recursively retrieves links from the main page and its sub-pages.
- **save_page_content(driver, url, save_directory)**: Navigates to a given URL, extracts content, and saves it to a file with a prefixed filename.

Function Descriptions

- **get_relative_links:**
    - Parameters:
    - **driver**: Selenium WebDriver instance.
    - **url**: URL of the page.
    - **exclude_keywords**: List of keywords to exclude from the links.
  - Returns: List of relative links.
- **get_links_recursive:**
    - Parameters:
    - **driver**: Selenium WebDriver instance.
    - **url**: URL of the page.
    - **exclude_keywords**: List of keywords to exclude from the links.
    - **links**: List to store collected links.
  - Returns: List of all collected links.
- **save_page_content:**
    - Parameters:
    - **driver**: Selenium WebDriver instance.
    - **url**: URL of the page.
    - **save_directory**: Directory to save the HTML content.

    - Returns: The saved file path.

Usage

- Configure the **chrome_driver_path** variable with the correct path to the ChromeDriver executable.
- Run the script.

Customization

File Naming

The script prefixes each file with the text extracted from the **<span itemprop="name">** tag. Modify the script if you need a different file-naming convention.

Excluded Keywords

Define keywords in the **exclude_keywords** list to filter out specific links.

Best Practices

- Use dynamic waits for page elements to ensure the script adapts to variations in page loading times.
- Regularly check for updates to the web page structure and adjust XPaths or selectors accordingly.

Troubleshooting

- Ensure the ChromeDriver path is correctly set.
- Check for updates to Selenium, BeautifulSoup, and ChromeDriver.
- Review error messages for insights.