Site_Crawler_Parser_All

Table of contents

Introduction

ProfileManager was designed for the aggregation of information related to corporate entities to support building business profiles. It uses the United States Securities and Exchange Commission (SEC) Central Index Key (CIK) to act as universally unique identifiers (UUIDs) and allows the user to compile a variety of information on corporate entities in an easy to use and query format because each profile is a dictionary. Assisting the accessibility of information, **Profile Manager** includes a series of mappings from CIK codes to names and back, names to aliases, and mappings from industry codes (namely The North American Industry Classification System (NAICS) and Standard Industrial Classification (SIC) codes) and descriptions of them. The hope to provide for a flexible data solution for complex business oriented applications.

Documentation

company_to_product(company, driver)

Search a company name on EWG and get all products made by the company in EWG database

Parameters

- company (str) : A company name to find products for
- driver (selenium.webdriver.Chrome) : Chrome driver after calling 'driver = setDriver()'

Returns

- dict
  - COMPANY (str) : a list of products made by the company

get_comp_name(text)

Extract relevant content of company name in a html tag

Parameters

- text (str) : raw content in a html tag

Returns

- str : a clean company name after junk texts are filtered

get_parent_child_dict(company,parent,children_list)

Build a dictionary that contains parent company, subsidiary company information for a certain company

Parameters

- company (str) : company name to build dictionary for
- parent (str) : The parent company name for the company
- children_list (list of strings) : list of subsidiary names of the company

Returns

- dict :
  - parent (str) : the parent company name
  - child (list of str) : a list of subsidiary names of the company

get_recursive_sub(company, driver)

Search "COMPANY_NAME+subsidiaries" RECURSIVELY on google chromedrivers directory to get all-level subsidiaries of a company and build a master dictionary that contains all-level subsidiary information for a company

Parameters

- company (str) : company name to find subsidiary for
- driver (selenium.webdriver.Chrome) : Chrome driver after calling 'driver = setDriver()'

Returns

- dict
  - company (str) :
  - parent (str) : the parent company of the company,'NA' if not found
  - child(list): a list of subsidiary names

get_tri_dict(tri_id, driver)

Open facility report page and scrape facility information into a dictionary

Parameters

- tri_id (str) : TRI facility id used as a unique identifier for a facility on TRI Search
- driver (selenium.webdriver.Chrome) : Chrome driver after calling 'driver = setDriver()'

Returns

- fac_dict (dict)
  - fac_name(str): Facility Name
  - tri_id(str): TRI facility ID
  - address(str): Facility Address
  - frs_id(str): FRS ID
  - mailing_name(str): Facility Mailing Name
  - mailing_address(str): Facility Mailing Address
  - duns_num(str): Facility Duns Number
  - parent_company(str): Facility's Parent Company Name
  - county(str): County
  - pub_contact(str): Public Contact Name
  - region(str): EPA Region Code
  - phone(str): Contact Number
  - latitude(str): Latitude
  - tribe(str): Tribe
  - longitude(str): Longitude
  - bia_tribal_code(str): BIA Tribal Code
  - naics(str): Naics Code
  - sic(str): SIC Code
  - last_form(str): Last Year of Report

google_sub(company, driver)

Search "COMPANY_NAME+subsidiaries" on google chromedrivers directory and scrape the knowledge graph results of subsidiary names returned by Google on the top

Parameters

- company (str) : A company name to find subsidiary for
- driver (selenium.webdriver.Chrome) : Chrome driver after calling 'driver = setDriver()'

Returns

- list : a list of subsidiary names(str)

hazard_to_company(chemical,driver)

Search NPIRS by entering a chemical name and get a list of companies that use the chemical in their products in NPIRS database

Parameters

- chemical (str) : a hazard name
- driver (selenium.webdriver.Chrome) : Chrome driver after calling 'driver = setDriver()'

Returns

- list of strings : a list of companies that use the hazard

product_to_ingredient(comp_prod_dict,driver)

Search a product name on EWG, get all ingredients in the product in EWG database, and build a master dictionary that contains information for company-products-ingredients

Parameters

- comp_prod_dict (dict) : dictionary that contains company to products information after calling 'comp_prod_dict = company_to_product(company, driver)'
- driver (selenium.webdriver.Chrome) : Chrome driver after calling 'driver = setDriver()'

Returns

- dict
  - COMPANY (str) :
  - PRODUCT (str): a list of ingredients in the company product

remove_null(comp_list)

Remove null values in a company list

Parameters

- comp_list (list of strings) : a list of companies

Returns

- str : a clean list of company names with no null values

setDriver(headless = False)

Sets a selenium webdriver object for running web-crawlers on various systems. Note: Requires chromedrivers for various platforms in a chromedrivers directory

Parameters

- headless (bool) : if True, sets a headless browser. if False (Default), sets a browser with head

Returns

- selenium.webdriver.Chrome : driver with standard option settings

wikiParser(company)

Search the Wikipedia page for a company and get wikipedia infobox together with all other contents

Parameters

- company (str) : the company you would like to query Wikipedia for

Returns

- tuple
  - dict : a dictionary of all other contents on wikipedia
  - dict : a dictionary of wikipedia infobox
  - str : page title
  - str : page url
  - beautifulsoup.table : wikipedia infobox HTML

Usage

NPIRS Engine is a site crawler for NPIRS(http://npirspublic.ceris.purdue.edu/ppis/) that gets all companies that use certain ingredients the user is looking for.

Function	Input	Processing	Output
hazard_to_company(chemical,driver)	ingredient name, chrome webdriver	search NPIRS by entering ingredient name and get company names	a list of companies that use the ingredient
setDriver()	None	set chrome driver used to automatically crawl websites	driver
get_comp_name(text)	an unfiltered string in html tags	extract relevant content	a string of the exact company name
remove_null(comp_list)	a list of company names	remove null values	a clean list of company names

Usage:

1. call `driver = setDriver()` to set chrome driver for crawling
2. call `hazard_to_company(chemical, driver)` to get a list of companies

Google Engine is a google crawler to find subsidiaries directly returned by google for a search query "COMPANY_NAME+subsidiaries".

Function	Input	Processing	
get_sub(company, driver)	company name, chrome webdriver	search "COMPANY_NAME+subsidiaries" on google	sub retu g

get_recursive_sub(company, driver)	a company name, chrome webdriver	search subsidiaries recursively on google	d that c na c and sub

Usage:

1. call `driver = setDriver()` to set chrome driver for crawling
2. call `master_google_sub = get_recursive_sub(company,driver)` to get a all-level-down subsidiaries for a company

TRI Engine is a site crawler for TRI Facility(https://www.epa.gov/enviro/tri-search) that gets all facility information with a tri id the user provides

Function	Input	Processing	Output
get_tri_dict(tri_id, driver)	tri facility id, chrome webdriver	open facility report page and scrape information into a dictionary	a dictionary of facility information

Usage:

1. call `driver = setDriver()` to set chrome driver for crawling
2. call `get_tri_dict(tri_id,driver)` to get a dictionary of facility information

EWG Engine is a site crawler for EWG Skindeep Database(https://www.ewg.org/skindeep/#.W3H8HNJKiUk) that gets product and ingredient information for a company in their database

Function	Input	Processing	Outp
company_to_product(company,driver)	company name,	search company name on	dictiona o

	chrome webdriver	EWG and get all products	compa to a list produc
product_to_ingredient(comp_prod_dict,driver)	company-product dictionay, chrome webdriver	search product name and get all ingredients	dictiona compa produc ingredier

Usage:

IMPORTANT NOTE: the driver needs to be set in a NON-HEADLESS mode. The user needs to manually close pop-up ads at the beginning for the crawler to function.

1. call `driver = setDriver()` to set chrome driver for crawling
2. call `comp_prod_dict = company_to_product(company,driver)` to get a dictionary of company to products
3. call `product_to_ingredient(comp_prod_dict,driver)` to get a dictionary of company to products to ingredients