Python Packages Used for Web Scraping:

1. Selenium
2. BeautifulSoup
3. Requests

APIs used: Oxylabs API

It offers real-time scraping functionality for accessing Lazada reviews, retrieving data in a structured format via proxy services. However, it only provides a 7-day free trial, so the authentication credentials must be updated once they expire for the code to continue working.

Link for reference:

1. GitHub - oxylabs/lazada-scraper: The Lazada Scraper API on Github is a powerful tool designed to retrieve product details, offers, reviews, and seller information from the Lazada e-commerce platform.
2. Lazada Scraper API | Free Trial (oxylabs.io)

URLs for the data source

1. [Free Shipping] KF94 face Mask mask kf94 50pcs malaysia Made in Korea Original 50PCS Washable Cloth Korea k f94 kf95 facemask viral With Design Kf94 Mask Original 50 Pcs Single Facialmask murah【Local Stock】| Lazada
2. KF94 Korea Mask 4 LAYER Disposable Earloop Face Mask KF94 Pelitup Muka/ KF94 (10PCS) | Lazada
3. KF94 Korea Earloop Headloop 4 PLY Disposable Earloop Face Mask KF94 Pelitup Muka/ KF94 韩国口罩 | Lazada

Table with brief descriptions of the data fields of the raw data.

Name	The buyer's name who left a review about the product.
Review	The comment or feedback provided by the buyer regarding the product.
SkuInfo	Details about the product category, such as color and quantity (e.g., number of pieces in a mask pack).
Date	The date when the buyer left the review.
StarCount	The number of stars (out of 5) given by the buyer to rate the product.

In [ ]:
```
## Author: Wong Yee En
```

Web Scraping

In [2]:
```
# Import the LazadaScrapeWithoutAPI class from the Python file
from lazada_scrape_selenium import LazadaScrapeWithoutAPI

# URL of the Lazada product page to scrape
url = 'https://www.lazada.com.my/products/kf94-korea-mask-4-layer-disposable-earloo

# Initialize the scraper object
scraper = LazadaScrapeWithoutAPI(url, max_pages=3)

# Scrape the reviews
scraper.scrape_reviews()

# Export the reviews to JSON and CSV files
scraper.export_to_json('lazada_reviews.json')
scraper.export_to_csv('lazada_reviews.csv')

# Clear reviews if needed (for a new scraping session)
scraper.clear_reviews()
```

1***1 Comfortable fit and stylish design, Easy to wear and fold, Effective four-laye
r filtration, color_family:White 11 May 2024 5
Mui L. Perfect size for adults, Comfortable fit and stylish design, Convenient earlo
op design, Variety of colors to choose from, color_family:Black 1 day ago 5
1***5 Waterproof and breathable mask, Variety of colors to choose from, Effective fo
ur-layer filtration, Perfect size for adults, color_family:White 1 week ago 5
Mui L. Perfect size for adults, Convenient earloop design, Comfortable fit and styli
sh design, color_family:White 1 day ago 5
ow K. abit more transparent or thinner material than usual. also smaller. fast deliv
ery. placed on Aug 5, got on Aug 7. color_family:White 07 Aug 2024 4
I***y Variety of colors to choose from,
Perfect size for adults, Easy to wear and fold. color_family:Black 3 weeks ago 5
Chan M. Convenient earloop design, High-quality materials used, color_family:White 1
week ago 5
J***y good value for money
delivery also ok
Thank you Lazada color_family:Black 3 weeks ago 5
NG T. Convenient earloop design, Convenient earloop design, color_family:White 3 we
eks ago 5
Coco M. Very cheap at less than 10c each of these mask. The quality is actually quit
e thin, just normal use, I don’t think suitable for patients. By the way it’s value
buy. color_family:Black 08 May 2024 5
1***1 Secure nose clip for a snug fit, Waterproof and breathable mask, Convenient ea
rloop design, color_family:Black 11 May 2024 5
P***. Easy to wear and fold, Price affordable
Perfect size for adults, color_family:Black 19 Jun 2024 5
zin M. I buy White 8 Not good ❌❌❌❌
black colour 👌👌👌👌👌👍👍👍👍

white Don't buy color_family:White 19 Aug 2023 5
ong T. this face mask good quality and good condition ,
fast delivery. color_family:Black 22 May 2024 5
D***. Goods received in good conditions. Well packed and fast delivery. color_famil
y:White 24 Apr 2024 5

API

In [3]:
```
# Import the LazadaScrapeWithAPI class
from lazada_scrape_api import LazadaScrapeWithAPI

# List of Lazada product URLs
urls = [
    'https://www.lazada.com.my/products/kf94-korea-earloop-headloop-4-ply-disposabl
    'https://www.lazada.com.my/products/kf94-korea-mask-4-layer-disposable-earloop-
]

# Authentication credentials for Oxylabs API
auth_credentials = ('YeeEn_J1wov', 'Yeeenlenglui00~')

# Initialize the scraper object
scraper = LazadaScrapeWithAPI(auth_credentials)

# Scrape reviews from the list of URLs
```

```
scraper.scrape_reviews(urls)

# Export the scraped data to JSON and CSV
scraper.export_to_json('reviews_api_example.json')
scraper.export_to_csv('reviews_api_example.csv')

# Clear reviews
scraper.clear_reviews()
```

In [ ]:

```python
## Author: Wong Yee En
import requests
from bs4 import BeautifulSoup
import json
import csv

class LazadaScrapeWithAPI:
    def __init__(self, auth_credentials):
        """
        Initialize the LazadaScrapeWithAPI class with authentication credentials for the Oxylabs API.
        """
        self.auth_credentials = auth_credentials
        self.all_reviews = []

    def scrape_reviews(self, urls):
        """
        Scrape reviews from a list of Lazada product URLs using the Oxylabs API.

        Args:
        urls (list): List of Lazada product URLs to scrape.

        Returns:
        None
        """
        for url in urls:
            # Set up proxy with requests
            payload = {
                'source': 'universal',
                'url': url,
                'render': 'html',
                'user_agent_type': 'desktop',
                'context': [{'key': 'follow_redirects', 'value': True}],
            }

            response = requests.post(
                'https://realtime.oxylabs.io/v1/queries',
                auth=self.auth_credentials,
                json=payload,
            )

            html_content = response.json()['results'][0]['content']

            # Parse the HTML content with BeautifulSoup
            soup = BeautifulSoup(html_content, 'html.parser')

            # Extract reviews for each URL
            nameContainers = soup.findAll('div', class_='middle')
            containers = soup.findAll('div', class_='item-content')
            dateContainers = soup.findAll('div', class_='top')

            for nameContainer, container, dateContainer in zip(nameContainers, containers, dateContainers):
                name = nameContainer.find('span')
                review = container.find('div', class_='content')
                skuInfo = container.find('div', class_='skuInfo')
```

```
                date = dateContainer.find('span', class_='title right')
                stars = dateContainer.find('div', class_='container-star starCtn left').find_all('img', class_='star', src='//img.lazcdn.com/g/tps/tfs/TB19ZvEgfDH8KJjy1XcXXcpdXXa-64-64.png')
                star_count = len(stars)

                if name and review and skuInfo and date and stars:
                    name_text = name.get_text(strip=True)
                    review_text = review.get_text(strip=True)
                    skuInfo_text = skuInfo.get_text(strip=True)
                    date_text = date.get_text(strip=True)
                    self.all_reviews.append((name_text, review_text, skuInfo_text, date_text, star_count))

    def export_to_json(self, file_name):
        """
        Export the scraped data to a JSON file.

        Args:
        file_name (str): The name of the JSON file to export the data.

        Returns:
        None
        """
        with open(file_name, 'w', encoding='utf-8') as f:
            json.dump(self.all_reviews, f, ensure_ascii=False, indent=4)

    def export_to_csv(self, file_name):
        """
        Export the scraped data to a CSV file.

        Args:
        file_name (str): The name of the CSV file to export the data.

        Returns:
        None
        """
        with open(file_name, 'w', newline='', encoding='utf-8') as f:
            writer = csv.writer(f)
            writer.writerow(['Name', 'Review', 'SkuInfo', 'Date', 'StarCount'])
            for name, review, skuInfo, date, star_count in self.all_reviews:
                writer.writerow([name, review, skuInfo, date, star_count])

    def clear_reviews(self):
        """
        Clear the stored reviews.

        Returns:
        None
        """
        self.all_reviews = []

```

```python
## Author: Wong Yee En
import time
from selenium import webdriver
from bs4 import BeautifulSoup
import json
import csv
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

class LazadaScrapeWithoutAPI:
    def __init__(self, url, max_pages=5):
        """
        Initialize the LazadaScrapeWithoutAPI class.

        Args:
        url (str): The URL of the Lazada product page to scrape.
        max_pages (int): The maximum number of pages to scrape reviews from.
        """
        self.url = url
        self.max_pages = max_pages
        self.reviews = []
        self.driver = None

    def initialize_driver(self):
        """
        Initialize the Selenium WebDriver with Chrome options.
        """
        options = webdriver.ChromeOptions()
        options.add_argument("--start-maximized")
        self.driver = webdriver.Chrome(options=options)
        self.driver.get(self.url)
        body = self.driver.find_element(By.TAG_NAME, 'body')
        for _ in range(10):  # Adjust the range for more/less increments
            body.send_keys(Keys.ARROW_DOWN)  # Scroll a small amount down
            body.send_keys(Keys.ARROW_DOWN)
            body.send_keys(Keys.ARROW_DOWN)
            time.sleep(2)  # Adjust sleep duration for slower or faster scrolling

    def scrape_reviews(self):
        """
        Scrape the reviews from the Lazada product page.
        """
        self.initialize_driver()

        for i in range(0, self.max_pages):
            soup = BeautifulSoup(self.driver.page_source, 'html.parser')

            nameContainers = soup.findAll('div', class_='middle')
            containers = soup.findAll('div', class_='item-content')
            dateContainers = soup.findAll('div', class_='top')

            for nameContainer, container, dateContainer in zip(nameContainers, containers,
dateContainers):
                name = nameContainer.find('span')
```

```
                review = container.find('div', class_='content')
                skuInfo = container.find('div', class_='skuInfo')
                date = dateContainer.find('span', class_='title right')
                stars = dateContainer.find('div', class_='container-star starCtn left').find_all('img', class_='star', src='//img.lazcdn.com/g/tps/tfs/TB19ZvEgfDH8KJjy1XcXXcpdXXa-64-64.png')
                star_count = len(stars)

                if name and review and skuInfo and date and stars:
                    name_text = name.get_text(strip=True)
                    review_text = review.get_text(strip=True)
                    skuInfo_text = skuInfo.get_text(strip=True)
                    date_text = date.get_text(strip=True)
                    self.reviews.append((name_text, review_text, skuInfo_text, date_text, star_count))
                    print(name_text, review_text, skuInfo_text, date_text, star_count)

            time.sleep(2)

            # Try to click the next button to go to the next page
            try:
                next_button = self.driver.find_element(By.CSS_SELECTOR, "button[class='next btn next-btn-normal next-btn-medium next-pagination-item next']")
                next_button.click()
            except:
                print("No more pages.")
                break

            time.sleep(3)

        self.driver.quit()

    def export_to_json(self, file_name):
        """
        Export the scraped reviews to a JSON file.

        Args:
        file_name (str): The name of the JSON file to export the data.
        """
        with open(file_name, 'w', encoding='utf-8') as f:
            json.dump(self.reviews, f, ensure_ascii=False, indent=4)

    def export_to_csv(self, file_name):
        """
        Export the scraped reviews to a CSV file.

        Args:
        file_name (str): The name of the CSV file to export the data.
        """
        with open(file_name, 'w', newline='', encoding='utf-8') as f:
            writer = csv.writer(f)
            writer.writerow(['Name', 'Review', 'SkuInfo', 'Date', 'StarCount'])
            for name, review, skuInfo, date, star_count in self.reviews:
                writer.writerow([name, review, skuInfo, date, star_count])

    def clear_reviews(self):
        """
```

```
109         Clear the stored reviews.
110         """
111         self.reviews = []
112
```

Data Ingestion

In [14]:
```
## Author: Wong Yee En
```

In [4]:
```
from pyspark.sql import SparkSession

spark = SparkSession\
        .builder\
        .appName("Ingestion")\
        .getOrCreate()
```

```
24/09/07 00:53:36 WARN SparkSession: Using an existing Spark session; only runtime S
QL configurations will take effect.
```

In [5]:
```
sc = spark.sparkContext
sc.addFile("../de_classes/data_storage/hadoop_file_handler.py")
```

```
24/09/07 00:53:36 WARN SparkContext: The path ../de_classes/data_storage/hadoop_file
_handler.py has been added already. Overwriting of added paths is not supported in t
he current version.
```

In [6]:
```
# Import the HadoopFileHandler class
from hadoop_file_handler import HadoopFileHandler

# Create an instance of HadoopFileHandler
handler = HadoopFileHandler()

df = handler.read_csv("data/raw/" + "*.csv")

output_path = "data/merged/merged_reviews.csv"
handler.write_csv(df, output_path)
```

```
24/09/07 00:53:37 WARN SparkSession: Using an existing Spark session; only runtime S
QL configurations will take effect.
```

In [7]:
```
df = handler.read_csv('data/merged/merged_reviews.csv')
```

In [8]:
```
total_rows = df.count()
print(f"Total number of rows: {total_rows}")
```

```
Total number of rows: 5767
```

In [9]:
```
df.show(truncate=True)
```

```
+------------+--------------------+--------------------+------------+---------+
|        Name|              Review|             SkuInfo|        Date|StarCount|
+------------+--------------------+--------------------+------------+---------+
|       1***1|Comfortable fit a...|  color_family:White| 11 May 2024|        5|
|     Coco M.|Very cheap at les...|  color_family:Black| 08 May 2024|        5|
|       1***1|Secure nose clip ...|  color_family:Black| 11 May 2024|        5|
|       P***.|Easy to wear and ...|  color_family:Black| 19 Jun 2024|        5|
|      ong T.|this face mask go...|  color_family:Black| 22 May 2024|        5|
|      zin M.|I buy White 8 Not...|  color_family:White| 19 Aug 2023|        5|
|       D***.|Goods received in...|  color_family:White| 24 Apr 2024|        5|
|       G***n|overall ok. U get...|  color_family:Black| 17 Apr 2024|        5|
|      nur N.|CANTIKKK GILE 😭 ...|   Color Family:Pink| 15 Feb 2022|        5|
|      Lim C.|Good packing, rec...|color_family:Navy...| 15 Dec 2023|        5|
|   Joanne L.|Goods received in...|  color_family:Black| 08 Dec 2023|        4|
|        Vivi|High-quality mate...|  color_family:White|  1  day ago|        5|
|  susilawati|maaf boss baru se...|  color_family:Black| 20 Oct 2023|        5|
|digitalvault|So far good, I wi...|  color_family:Black|2  weeks ago|        5|
|       ****2|Pretty decent mas...|  color_family:Black|2  weeks ago|        5|
|       R***.|High-quality mate...|  color_family:White| 1  week ago|        5|
|     Suki S.|Received in short...|    color_family:Red| 02 Oct 2023|        5|
|     Suki S.|Received in short...| color_family:Purple| 02 Oct 2023|        5|
|     Suki S.|Received in short...|   color_family:Pink| 02 Oct 2023|        5|
|      Jay L.|High-quality mate...|  color_family:White| 29 Jun 2024|        5|
+------------+--------------------+--------------------+------------+---------+
only showing top 20 rows
```

In [10]:
```
spark.stop()
```