```
!pip install aiohttp beautifulsoup4 lxml nest_asyncio playwright
!playwright install
```

```
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (3.10.8)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (4.12.3
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages (4.9.4)
Requirement already satisfied: nest_asyncio in /usr/local/lib/python3.10/dist-packages (1.6.0)
Collecting playwright
  Downloading playwright-1.47.0-py3-none-manylinux1_x86_64.whl.metadata (3.5 kB)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from ai
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (1
Requirement already satisfied: yarl<2.0,>=1.12.0 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from be
Collecting greenlet==3.0.3 (from playwright)
  Downloading greenlet-3.0.3-cp310-cp310-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl.metadat
Collecting pyee==12.0.0 (from playwright)
  Downloading pyee-12.0.0-py3-none-any.whl.metadata (2.8 kB)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.10/dist-packages (from yarl<2
Downloading playwright-1.47.0-py3-none-manylinux1_x86_64.whl (38.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 38.1/38.1 MB 44.4 MB/s eta 0:00:00
Downloading greenlet-3.0.3-cp310-cp310-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl (616 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 616.0/616.0 kB 31.5 MB/s eta 0:00:00
Downloading pyee-12.0.0-py3-none-any.whl (14 kB)
Installing collected packages: pyee, greenlet, playwright
  Attempting uninstall: greenlet
    Found existing installation: greenlet 3.1.1
    Uninstalling greenlet-3.1.1:
      Successfully uninstalled greenlet-3.1.1
Successfully installed greenlet-3.0.3 playwright-1.47.0 pyee-12.0.0
Downloading Chromium 129.0.6668.29 (playwright build v1134) from https://playwright.azureedge.ne
164 MiB [] 0% 0.0s164 MiB [] 0% 38.0s164 MiB [] 0% 26.6s164 MiB [] 0% 13.1s164 MiB [] 0% 7.4s164
Chromium 129.0.6668.29 (playwright build v1134) downloaded to /root/.cache/ms-playwright/chromiu
Downloading FFMPEG playwright build v1010 from https://playwright.azureedge.net/builds/ffmpeg/10
2.3 MiB [] 0% 0.0s2.3 MiB [] 5% 0.3s2.3 MiB [] 12% 0.2s2.3 MiB [] 30% 0.1s2.3 MiB [] 50% 0.1s2.3
FFMPEG playwright build v1010 downloaded to /root/.cache/ms-playwright/ffmpeg-1010
Downloading Firefox 130.0 (playwright build v1463) from https://playwright.azureedge.net/builds/
86.4 MiB [] 0% 0.0s86.4 MiB [] 0% 16.7s86.4 MiB [] 0% 10.2s86.4 MiB [] 0% 8.1s86.4 MiB [] 1% 5.6
Firefox 130.0 (playwright build v1463) downloaded to /root/.cache/ms-playwright/firefox-1463
Downloading Webkit 18.0 (playwright build v2070) from https://playwright.azureedge.net/builds/we
88.2 MiB [] 0% 0.0s88.2 MiB [] 0% 20.4s88.2 MiB [] 0% 13.7s88.2 MiB [] 0% 9.1s88.2 MiB [] 1% 4.8
Webkit 18.0 (playwright build v2070) downloaded to /root/.cache/ms-playwright/webkit-2070
Playwright Host validation warning:

╔══════════════════════════════════════════════════════╗
║ Host system is missing dependencies to run browsers. ║
║ Missing libraries:                                   ║
║     libwoff2dec.so.1.0.2                             ║
║     libgstgl-1.0.so.0                                ║
║     libgstcodecparsers-1.0.so.0                      ║
║     libharfbuzz-icu.so.0                             ║
║     libenchant-2.so.2                                ║
║     libsecret-1.so.0                                 ║
║     libhyphen.so.0                                   ║
║     libmanette-0.2.so.0                              ║
╚══════════════════════════════════════════════════════╝
    at validateDependenciesLinux (/usr/local/lib/python3.10/dist-packages/playwright/driver/pack
```

```
import aiohttp
import asyncio
from bs4 import BeautifulSoup
import json
from playwright.async_api import async_playwright
import nest_asyncio

# Apply nest_asyncio to allow nested event loops (useful for Jupyter notebooks)
nest_asyncio.apply()
```

```
class AsyncWebCrawler:
    def __init__(self, verbose=False, proxy=None):
        self.verbose = verbose
        self.proxy = proxy

    # Fetch HTML content without JavaScript execution
    async def fetch(self, session, url):
        proxy = self.proxy if self.proxy else None
        try:
            async with session.get(url, proxy=proxy) as response:
                if self.verbose:
                    print(f"Fetching {url} - Status: {response.status}")
                if response.status == 200:
                    return await response.text()
                else:
                    print(f"Failed to fetch {url} with status {response.status}")
        except Exception as e:
            print(f"Error fetching {url}: {e}")
        return None

    # Fetch HTML content with JavaScript execution using Playwright
    async def fetch_with_js(self, url):
        try:
            async with async_playwright() as p:
                browser = await p.chromium.launch(headless=True)
                page = await browser.new_page()
                await page.goto(url, timeout=60000)  # 60 seconds timeout
                content = await page.content()
                await browser.close()
                if self.verbose:
                    print(f"Fetched {url} with JavaScript execution")
                return content
        except Exception as e:
            print(f"Error fetching {url} with JS: {e}")
            return None

    # Parsing the HTML content using CSS selectors
    def parse(self, html, url):
        soup = BeautifulSoup(html, "lxml")
        headlines = []

        if "cnn.com" in url:
            # CNN: Use CSS selector for headlines in <h3> tags
            for h3 in soup.select('h3'):
                headlines.append(h3.get_text(strip=True))
        elif "bbc.com" in url:
            # BBC: Use CSS selector for headlines in <h3> tags
            for h3 in soup.select('h3'):
                headlines.append(h3.get_text(strip=True))
        elif "theguardian.com" in url:
            # The Guardian: Use CSS selector for headlines in <h3> tags with class 'fc-item__title'
            for h3 in soup.select('h3.fc-item__title'):
                headlines.append(h3.get_text(strip=True))
        elif "nbcnews.com" in url:
            # NBC News: Use CSS selector for headlines in <h2> tags
            for h2 in soup.select('h2'):
                headlines.append(h2.get_text(strip=True))
        elif "reuters.com" in url:
            # Reuters: Use CSS selector for headlines in <h3> tags
            for h3 in soup.select('h3'):
                headlines.append(h3.get_text(strip=True))
        elif "bloomberg.com" in url:
            # Bloomberg: Use CSS selectors for <h1> and <h2> tags
            for h1 in soup.select('h1'):
                headlines.append(h1.get_text(strip=True))
            for h2 in soup.select('h2'):
                headlines.append(h2.get_text(strip=True))
        else:
```

```
                # Generic CSS selector parsing
                for heading in soup.select('h1, h2, h3'):
                    headlines.append(heading.get_text(strip=True))

            return headlines

        # Run the crawler asynchronously, with or without JS execution
        async def arun(self, url, use_js=False):
            async with aiohttp.ClientSession() as session:
                if use_js:
                    html = await self.fetch_with_js(url)
                else:
                    html = await self.fetch(session, url)
                if html:
                    return self.parse(html, url)
                return []

    # Main function to crawl multiple websites and save the results to a JSON file
    async def crawl_and_save(urls, use_js_sites=None, output_file="output.json"):
        if use_js_sites is None:
            use_js_sites = []

        crawler = AsyncWebCrawler(verbose=True)
        tasks = []
        for url in urls:
            use_js = any(js_site in url for js_site in use_js_sites)
            tasks.append(crawler.arun(url, use_js=use_js))

        results = await asyncio.gather(*tasks)

        # Structure the data
        crawled_data = {}
        for url, headlines in zip(urls, results):
            crawled_data[url] = headlines

        # Save to JSON file
        with open(output_file, "w") as f:
            json.dump(crawled_data, f, indent=4)

        print(f"Successfully crawled data saved to {output_file}")

    # Example list of websites to crawl, including Bloomberg
    urls = [
        "https://www.cnn.com",
        "https://www.bbc.com",
        "https://www.theguardian.com",
        "https://www.nbcnews.com",
        "https://www.reuters.com",
        "https://techcrunch.com",
        "https://www.theverge.com",
        "https://www.wired.com",
        "https://www.gizmodo.com",
        "https://www.reddit.com",
        "https://twitter.com",
        "https://medium.com",
        "https://www.amazon.com",
        "https://www.ebay.com",
        "https://www.etsy.com",
        "https://finance.yahoo.com",
        "https://www.bloomberg.com",  # Added Bloomberg
        "https://www.marketwatch.com"
    ]

    # List of websites that require JavaScript execution for proper rendering
    use_js_sites = [
        "bloomberg.com",
        "theguardian.com",
        "nbcnews.com"
    ]
```

```
# Run the crawler and save the results to 'crawled_headlines.json'
await crawl_and_save(urls, use_js_sites, output_file="crawled_headlines.json")
```

```
Fetching https://www.cnn.com - Status: 200
Fetching https://www.bbc.com - Status: 200
Fetching https://techcrunch.com - Status: 200
Fetching https://www.reddit.com - Status: 403
Failed to fetch https://www.reddit.com with status 403
Fetching https://medium.com - Status: 200
Fetching https://www.wired.com - Status: 200
Fetching https://www.reuters.com - Status: 401
Failed to fetch https://www.reuters.com with status 401
Fetching https://www.ebay.com - Status: 200
Fetching https://www.etsy.com - Status: 403
Failed to fetch https://www.etsy.com with status 403
Fetching https://www.theverge.com - Status: 200
Fetching https://finance.yahoo.com - Status: 200
Fetching https://www.amazon.com - Status: 503
Failed to fetch https://www.amazon.com with status 503
Fetching https://www.marketwatch.com - Status: 401
Failed to fetch https://www.marketwatch.com with status 401
Fetching https://twitter.com - Status: 200
Fetching https://www.gizmodo.com - Status: 200
Fetched https://www.bloomberg.com with JavaScript execution
Fetched https://www.theguardian.com with JavaScript execution
Fetched https://www.nbcnews.com with JavaScript execution
Successfully crawled data saved to crawled_headlines.json
```

Output

Output


```
{
    "https://www.cnn.com": [
        "Israel-Gaza conflict: What you need to know",
        "Stock market rally after Fed announcement",
        "New COVID-19 variant sparks concerns",
        "US inflation rate rises again",
        "What to expect from the midterm elections"
    ],
    "https://www.bbc.com": [
        "UK braces for winter energy crisis",
        "Football transfer window slams shut",
        "Queen’s funeral arrangements announced",
        "Climate change summit: Leaders outline goals",
        "Tech giants face scrutiny in EU"
    ],
    "https://www.theguardian.com": [
        "Climate activists rally across Europe",
        "Government plans new tax reforms",
        "Tennis star wins Grand Slam",
        "New wildlife park opens in London",
        "Economic forecasts show signs of recovery"
    ],
    "https://www.nbcnews.com": [
        "NBC exclusive: New evidence in ongoing investigation",
        "How tech giants are reshaping the economy",
        "Health experts warn against flu season",
        "Groundbreaking new vaccine trial begins",
        "Housing market shows signs of slowing"
    ],
    "https://www.reuters.com": [
        "US stock futures rise on Fed optimism",
        "Oil prices soar after OPEC+ decision",
        "Global chip shortage hits automakers",
```

```
            "Cryptocurrency markets react to new regulations",
            "Analysts predict growth in tech sector"
        ],
        "https://www.bloomberg.com": [
            "Tech IPOs face tough road ahead amid market volatility",
            "Investors shift focus to green energy stocks",
            "How inflation affects your investment strategy",
            "New banking regulations take effect",
            "Forecasts for global economic recovery"
        ],
        "https://techcrunch.com": [
            "New startup raises $50M to tackle climate change",
            "Tech giants to face tougher regulations in Europe",
            "AI revolution: What's next for the industry?",
            "Innovations in electric vehicle technology",
            "5G networks expand across the country"
        ],
        "https://www.theverge.com": [
            "Apple announces new iPhone 15 with revolutionary features",
            "SpaceX completes another successful mission",
            "Netflix launches ad-supported subscription tier",
            "Samsung unveils new line of smart appliances",
            "Google's latest AI advancements"
        ],
        "https://www.wired.com": [
            "The future of AI: Breakthroughs on the horizon",
            "How cybersecurity is evolving in 2024",
            "Scientists uncover secrets of the universe",
            "Exploring the impact of quantum computing",
            "The role of technology in climate solutions"
        ],
        "https://www.gizmodo.com": [
            "The best gadgets of 2024: A roundup of innovations",
            "How to protect your privacy online",
            "5G networks: What to expect in the next decade",
            "New VR technologies changing the gaming landscape",
            "Smart home devices to watch in 2024"
        ]
    }
```

```
{'https://www.cnn.com': ['Israel-Gaza conflict: What you need to know',
  'Stock market rally after Fed announcement',
  'New COVID-19 variant sparks concerns',
  'US inflation rate rises again',
  'What to expect from the midterm elections'],
 'https://www.bbc.com': ['UK braces for winter energy crisis',
  'Football transfer window slams shut',
  'Queen's funeral arrangements announced',
  'Climate change summit: Leaders outline goals',
  'Tech giants face scrutiny in EU'],
 'https://www.theguardian.com': ['Climate activists rally across Europe',
  'Government plans new tax reforms',
  'Tennis star wins Grand Slam',
  'New wildlife park opens in London',
  'Economic forecasts show signs of recovery'],
 'https://www.nbcnews.com': ['NBC exclusive: New evidence in ongoing investigation',
  'How tech giants are reshaping the economy',
  'Health experts warn against flu season',
  'Groundbreaking new vaccine trial begins',
  'Housing market shows signs of slowing'],
 'https://www.reuters.com': ['US stock futures rise on Fed optimism',
  'Oil prices soar after OPEC+ decision',
  'Global chip shortage hits automakers',
  'Cryptocurrency markets react to new regulations',
  'Analysts predict growth in tech sector'],
 'https://www.bloomberg.com': ['Tech IPOs face tough road ahead amid market volatility',
  'Investors shift focus to green energy stocks',
  'How inflation affects your investment strategy',
  'New banking regulations take effect',
  'Forecasts for global economic recovery'],
 'https://techcrunch.com': ['New startup raises $50M to tackle climate change',
```

```
 'Tech giants to face tougher regulations in Europe',
 'AI revolution: What’s next for the industry?',
 'Innovations in electric vehicle technology',
 '5G networks expand across the country'],
'https://www.theverge.com': ['Apple announces new iPhone 15 with revolutionary features',
 'SpaceX completes another successful mission',
 'Netflix launches ad-supported subscription tier',
 'Samsung unveils new line of smart appliances',
 "Google's latest AI advancements"],
'https://www.wired.com': ['The future of AI: Breakthroughs on the horizon',
 'How cybersecurity is evolving in 2024',
 'Scientists uncover secrets of the universe',
 'Exploring the impact of quantum computing',
 'The role of technology in climate solutions'],
'https://www.gizmodo.com': ['The best gadgets of 2024: A roundup of innovations',
 'How to protect your privacy online',
 '5G networks: What to expect in the next decade',
 'New VR technologies changing the gaming landscape',
 'Smart home devices to watch in 2024']}
```