

Автоматизированный сбор новостных данных с использованием Selenium WebDriver и Python: Яндекс.Дзен

Языков Кирилл Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается разработка системы автоматизированного сбора новостных данных с веб-сайта Яндекс.Дзен. Представлен подход к парсингу динамически загружаемого контента с использованием библиотеки Selenium WebDriver и языка программирования Python. Описывается практическая реализация парсинга новостных заголовков и URL-адресов новостей, а также методы обработки ошибок и обеспечения надежности извлечения данных. Проведен анализ структуры HTML-кода целевого ресурса, что позволило разработать эффективные CSS-селекторы для идентификации и извлечения необходимых элементов. Статья представляет собой case study, демонстрирующий возможности применения Selenium WebDriver и Python для автоматизации сбора информации из веб-источников.

Ключевые слова: парсинг, веб-скрапинг, Selenium WebDriver, Python, автоматизация, новостные данные, Яндекс.Дзен, CSS-селекторы, обработка ошибок.

Automated news data collection using Selenium WebDriver and Python: Yandex Zen

Yazykov Kirill Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article discusses the development of an automated system for collecting news data from the Yandex Zen website. An approach to parsing dynamically loaded content using the Selenium WebDriver library and the Python programming language is presented. It describes the practical implementation of parsing news headlines and news URLs, as well as methods for error handling and ensuring reliable data extraction. The structure of the HTML code of the target resource was analyzed, which made it possible to develop effective CSS selectors for identifying and extracting the necessary elements. The article is a case study demonstrating the possibilities of using Selenium WebDriver and Python to automate the collection of information from web sources.

Keywords: parsing, web scraping, Selenium WebDriver, Python, automation, news data, Yandex.Zen, CSS selectors, error handling.

1 Введение

1.1 Актуальность

В эпоху информационного изобилия автоматизированный сбор данных из веб-источников становится все более востребованным инструментом для анализа данных, мониторинга новостей, исследования трендов и принятия решений на основе данных. Автоматизация процесса позволяет значительно снизить трудозатраты и повысить эффективность сбора информации, обеспечивая актуальность и оперативность получаемых данных. Платформа Яндекс.Дзен является популярным новостным агрегатором, что делает актуальным изучение методов парсинга данных именно с этого ресурса.

1.2 Обзор исследований

Для ознакомления с языком программирования Python и его применением в разработке графических интерфейсов, рекомендуем обратиться к следующим источникам. В.А. Машков и В.И. Литвиненко рассматривают применение Python для задач самодиагностики на системном уровне [1]. Г.Д. Бухарова и П.С. Комельских подчеркивают важность и необходимость включения Python в учебный процесс [2]. Г.С. Сейдаметов демонстрирует особенности использования Python в подготовке инженеров-программистов [3], а Э.А. Усеинов иллюстрирует применение объектно-ориентированного программирования в рамках изучения Python [4]. Разработка приложений с графическим пользовательским интерфейсом (GUI) на Python с использованием Tkinter также активно исследовалась. Например, А.С. Матвеева в своей статье рассматривает применение паттерна «Команда» для создания модульных и структурированных приложений на Tkinter, демонстрируя повышение гибкости, расширяемости, надежности и удобства использования [5].

1.3 Цель исследования

Целью данного исследования является разработка и реализация системы автоматизированного сбора новостных данных с веб-сайта Яндекс.Дзен с использованием Selenium WebDriver и языка программирования Python. Для достижения цели были поставлены следующие задачи:

- Разработать программный код на Python с использованием библиотеки Selenium WebDriver для автоматизации процесса парсинга.
- Реализовать методы обработки ошибок для повышения надежности извлечения данных.
- Провести тестирование разработанной системы и оценить ее эффективность.

2 Материалы и методы

В качестве материалов использован язык программирования Python и библиотека Selenium WebDriver для автоматизации управления веб-браузером. Для работы с веб-сайтом Яндекс.Дзен был проведен анализ его HTML-структуры с применением инструментов разработчика веб-браузера. Для идентификации элементов использовались CSS-селекторы, которые были тщательно подобраны и протестированы. Для обработки ошибок и обеспечения надежности извлечения данных применялись методы обработки исключений в Python. Программный код был реализован в среде разработки (например, VisualCode) с использованием библиотеки webdriver-manager для автоматической установки драйвера веб-браузера.

3 Результаты и обсуждения

1. Анализ структуры HTML и выбор селекторов (Подготовка к парсингу)

Прежде чем приступить к написанию кода, необходимо провести анализ структуры HTML-кода целевого веб-сайта. Это необходимо для идентификации элементов, содержащих интересующие данные (в нашем случае, заголовки новостей и URL-адреса). Правильный выбор селекторов (CSS-селекторы или XPath) является критическим для корректного извлечения информации. Ошибки в селекторах могут привести к неверным результатам или сбою работы парсера.

Действия:

- Открываем сайт Яндекс.Дзен в браузере и, используя инструменты разработчика (обычно активируются нажатием клавиши F12 или щелчком правой кнопкой мыши и выбором “Inspect”), анализируем HTML-код главной страницы.
- Определяем теги и атрибуты, содержащие заголовки новостей. В нашем случае, заголовки располагались внутри элементов `<p class="dzen-desktop--card-top-avatar__text-Pu">`. С точки зрения HTML-структуры, заголовок является текстом внутри тега `<p>`, имеющего определенный класс.
- Определяем элементы ссылок. Ссылки, будут внутри тегов `<a>`.
- Определяем селекторы, которые будут использоваться для поиска этих элементов.

2. Установка необходимых библиотек и настройка WebDriver (Настройка среды разработки)

Для работы с Selenium WebDriver и Python необходимо установить соответствующие библиотеки. Selenium позволяет управлять браузером, а Python обеспечивает среду для написания кода. Для удобства и автоматизации управления браузером (в данном случае, Chrome), используется webdriver-manager (рис. 1).

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
import time

url = "https://dzen.ru/"

try:
    driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
except Exception as e:
    print(f"Ошибка при инициализации WebDriver: {e}")
    print("Убедитесь, что у вас установлен Chrome (или другой выбранный браузер) и драйвер для него.")
    exit()
```

Рисунок 1 – Код импорта библиотек и браузера Google

3. Открытие веб-сайта и добавление задержки (Загрузка страницы) (рис. 2)

Необходимо загрузить целевую веб-страницу в браузере, управляемом WebDriver. Задержки (`time.sleep()`) необходимы для обеспечения полной загрузки страницы, включая динамически загружаемый контент, что является особенностью современных веб-сайтов. Недостаточная задержка может привести к тому, что парсер не найдет элементы, которые загружаются асинхронно.

```
try:
    driver.get(url)
    print(f"Открываем сайт: {url}")
    time.sleep(3)
except Exception as e:
    print(f"Ошибка при открытии сайта: {e}")
    driver.quit()
    exit()
```

Рисунок 2 – Код открытия сайт и добавления задержки

4. Поиск элементов на странице (Поиск заголовков и URL) (рис. 3)

После загрузки страницы необходимо найти элементы, содержащие интересные нас данные. Для этого используется метод `driver.find_elements()`, который позволяет найти элементы по заданному селектору (CSS или XPath). Использование селекторов, соответствующих структуре HTML, является ключевым для успешного парсинга.

```
try:
    headlines = driver.find_elements(By.CSS_SELECTOR, "p.dzen-desktop--card-top-avatar__text-Pu")
    print(f"Найдено {len(headlines)} заголовков.")

    link_elements = driver.find_elements(By.CSS_SELECTOR, "a.dzen-desktop--card-top-avatar__link-Lh")
    print(f"Найдено {len(link_elements)} ссылок.")

except Exception as e:
    print(f"Ошибка при поиске заголовков или ссылок: {e}")
    driver.quit()
    exit()
```

Рисунок 3 – Код поиска заголовков и URL адресов

5. Извлечение данных (Получение заголовков и URL) (рис. 4)

После нахождения элементов необходимо извлечь из них интересные данные. В данном случае извлекается текст заголовков и атрибут href (URL ссылки). Для этого используются методы `.text` (для текста) и `.get_attribute('href')` (для получения значения атрибута). Важно применять методы обработки ошибок, чтобы предотвратить сбои в работе скрипта при возникновении проблем с данными.

```
def get_text_safe(element):
    try:
        return element.text.strip()
    except:
        return ""

def get_attribute_safe(element, attribute_name):
    try:
        return element.get_attribute(attribute_name)
    except:
        return ""

if headlines and link_elements:
    for i in range(len(headlines)):
        try:
            headline_text = get_text_safe(headlines[i])
            link_element = link_elements[i]
            link_url = get_attribute_safe(link_element, 'href')
            if headline_text and link_url:
                print(f"Заголовок: {headline_text}")
                print(f"URL: {link_url}")
                print("-" * 20)
        except IndexError:
            print(f"Ошибка: Не удалось сопоставить заголовок и URL по индексу {i}.")
            continue
    else:
        print("Заголовки или ссылки не найдены. Проверьте CSS-селекторы.")
```

Рисунок 4 – Код для извлечения данных

6. Закрытие браузера (Завершение работы) (рис. 5)

После завершения парсинга необходимо закрыть экземпляр браузера, управляемый WebDriver, чтобы освободить ресурсы.

```
driver.quit()
print("Работа завершена.")
```

Рисунок 5 – Код закрытия браузера

7. Результат работы (рис. 6)

```

Найдено 14 сопок.
Заголовок: Над российскими регионами за ночь уничтожили 39 беспилотников
URL: https://dzen.ru/news/story/83b6d71-085c-53da-9242-b6dd8024144?lang=ru&from=main_portal&fan=1&annot_type=trust&t=175899832&persistent_id=318338354&c14url=85e97328c75106ac1ef9737ee5a5a6dd&st=175100069
&story=47a1c99-3b3d-547f-a6dc-e5ba0a08f2f7

Заголовок: Захарова: ВСУ понесут наказание за атаку на китайского журналиста
URL: https://dzen.ru/news/story/c542b0f1-cd93-52e9-a559-594a1aeca3ab?lang=ru&from=main_portal&fan=1&annot_type=trust&t=175899832&persistent_id=3183379187&c14url=27e70c4ff547fbac0546eb2d2106d0&st=175100069
&story=4b33caa4-77f9-552f-aa09-516c3d7c0e15

Заголовок: Euractiv: Макрон с опозданием прибыл на саммит ЕС в расстроенном виде
URL: https://dzen.ru/news/story/d03cc300-448c-5720-ac60-f5348136112?lang=ru&from=main_portal&fan=1&annot_type=trust&t=175899832&persistent_id=318337421&c14url=e2fc56fa6e93b401edcd29b1e61b&st=175100069
&story=9dd5a0e5-e520-53b7-be10-c90076a069a6

Заголовок: Губернатор Федорченко: БПЛА атаковали предприятие в Самарской области
URL: https://dzen.ru/news/story/91f50e2b-30ab-58d5-b075-1d8e9e764712?lang=ru&from=main_portal&fan=1&annot_type=trust&t=175899832&persistent_id=3183383492&c14url=a4ec9e4a16915ff2552cf82b5b3985&st=175100069
&story=45095087-2e02-54d8-92e3-1b239ff9cc4a

Заголовок: Алаудинов: СБУ хочет убить офицеров спецназа «Аомат», включая группу «Аидар»
URL: https://dzen.ru/news/story/6b834a5c-c920-5b0c-acd3-8f2a174c0556?lang=ru&from=main_portal&fan=1&annot_type=trust&t=175899832&persistent_id=3183381621&c14url=14aa72971d441f3b9d03a8de8a407674&st=175100069
&story=ac1a1af4-d44e-55de-ad10-e9f0d52c7b3f

Заголовок: Против России совершили грубую провокацию
URL: https://dzen.ru/news/story/5f753201-e847-54a7-98d3-106ba79797bd?c14url=9596f5b351aaf88f6425663154270916&fan=1&lang=ru&opertop=1&persistent_id=3183374783&rubric=personal_feed&story=98456ae9-e685-5e18-b31d
-bd88aef1ba2&st=175899832&tail_polnote&st=5

Заголовок: Скончался Рюкши Шуленин: режиссер и член Союза, снимавший свыше 30 фильмов
URL: https://dzen.ru/news/story/c71882b9-3981-5b1c-bf84-bf9f8cb780a?c14url=09f48b948745023d624aa69767f782&fan=1&lang=ru&opertop=1&persistent_id=3183372170&rubric=personal_feed&story=f1b009ff-4249-5714-8ba9
-ea047d52527&st=175899832&tail_polnote&st=5

Заголовок: Глава ФАС Школьников: Рынок маркетплейсов грозит окончательная монополизация
URL: https://dzen.ru/news/story/ffa5f930-ca98-5599-a52d-c0e762d223ae?c14url=b99249c63a3b0bf324fafb57664729&fan=1&lang=ru&opertop=1&persistent_id=3183371083&rubric=personal_feed&story=36617ef-8d68-5cc5-a4b8
-e39abccf33d&st=175899832&tail_polnote&st=5

Заголовок: «Известия»: Июль принесет в разные регионы РФ ненастную погоду
URL: https://dzen.ru/news/story/d056b45e-33ae-5f18-b63b-6b7b440c713b?c14url=8845d87080b90bc5784e185d727b6b18&fan=1&lang=ru&opertop=1&persistent_id=3183380159&rubric=personal_feed&story=4f2e3315-e99e-56cb-8719
-0f4cd51594&st=175899832&tail_polnote&st=5

Заголовок: Новостка Королева показала результат операций на лице
URL: https://dzen.ru/news/story/0c51b708-9ac5-5fa3-ab7c-5d0bb170217?c14url=f753636990d6c61ba3dc3da8ae2c18&fan=1&lang=ru&opertop=1&persistent_id=3183371560&rubric=personal_feed&story=7351161a-034e-593a-9e3a
-9e7a2933205&st=175899832&tail_polnote&st=5

Заголовок: Определелись все участники плей-офф клубного чемпионата мира по футболу
URL: https://sportsdzen.ru/news/story/989a5738-c052-5b0c-ba05-1eed4c234ba?c14url=55e7c3a307947c7b663104563bee91338&fan=1&lang=ru&opertop=1&persistent_id=3180161557&rubric=personal_feed&story=8347034e-2f63-553
-c4ae-b750eb800f4&st=175899832&tail_polnote&st=5

Заголовок: Названы самые распространенные виды рака в Якутии
URL: https://dzen.ru/news/story/477c75a1-b6a5-5034-bdb5-99841bd8287d?c14url=f47f18d7db20e11990b358a917a2748&fan=1&lang=ru&opertop=1&persistent_id=3183383554&rubric=personal_feed&story=efc9a435-70b9-5cb8-9d0f
-bb1ad31904&st=175899832&tail_polnote&st=5

```

Рисунок 6 – Результат работы

Выводы

В результате проведенного исследования была разработана работоспособная система автоматизированного сбора новостных данных с веб-сайта Яндекс.Дзен. Было продемонстрировано применение Selenium WebDriver и Python для эффективного парсинга динамически загружаемого контента. Разработанный программный код обеспечивает надежный и автоматизированный сбор новостных заголовков и URL-адресов новостей. В процессе реализации были выявлены и решены проблемы, связанные со структурой HTML-кода и динамической загрузкой контента. Предложенные методы обработки ошибок повышают устойчивость системы к изменениям в структуре сайта. Полученные результаты могут быть использованы для дальнейшей разработки систем сбора данных, анализа новостных трендов и мониторинга информации из веб-источников. Необходимо отметить, что структура веб-сайтов может меняться, поэтому разработанный код нуждается в регулярном обновлении CSS-селекторов для поддержания работоспособности системы.

Библиографический список

1. Машков В.А., Литвиненко В.И. Использование языка программирования python 3 и системы компьютерной алгебры sympy на факультативных занятиях по теории чисел // Электротехнические и компьютерные системы. 2017. С. 48-54
2. Бухарова Г.Д., Комельских П.С. Важность и необходимость внедрения языка программирования python в процесс обучения студентов // Новые информационные технологии в образовании Материалы VII международной научно-практической конференции. 2014. С. 40-42.
3. Сейдаметов Г.С. Особенности использования языка программирования python в подготовке будущих инженеров-программистов // International

scientific review. 2019. №. 9(10). С. 50-51

4. Усеинов Э.А. Объектно-ориентированное программирование в рамках дисциплины «язык программирования python» // Ученые записки крымского инженерно-педагогического университета. 2012. №. 34. С. 157-160.
5. Матвеева А. С. Реализация паттерна поведения «Команда» в графическом интерфейсе пользователя на Python // Постулат. 2025. №1.