

Создание парсера для перевода теста из Yandex форм в документ Word

Языков Кирилл Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается процесс разработки программы, которая будет использоваться педагогами в учебных образовательных учреждениях при помощи парсинга web-страницы. Основное внимание уделяется этапам настройки среды, библиотек и обеспечению автономной работы с любого компьютера, способное работать как приложение. Приведены примеры реализации, описаны архитектура решения и использованные инструменты. Работа может быть полезна разработчикам и исследователям, интересующимся созданием нового программного обеспечения для образовательных учреждений.

Ключевые слова: разработка программы, учебные образовательные учреждения, парсинг web-страницы, приложение.

Creating a parser to translate a test from Yandex forms into a Word document

Kirill Alexandrovich Yazykov

Sholom-Aleichem Priamursky State University

Student

Abstract

The article discusses the process of developing a program that will be used by teachers in educational institutions by parsing a web page. The main focus is on the steps of setting up the environment, libraries, and ensuring that the program can be run independently on any computer, similar to an application. The article provides examples of implementation, describes the architecture of the solution, and outlines the tools used. This work can be useful for developers and researchers interested in creating new software for educational institutions.

Keywords: software development, educational institutions, web page parsing, application.

1 Введение

1.1 Актуальность

В современном образовательном процессе широко применяются электронные формы для создания контрольных работ и тестов. Одним из популярных инструментов является сервис Яндекс.Формы, который позволяет быстро создавать и распространять тесты среди учащихся. Однако для

преподавателей в условиях современных запретов на мобильные телефоны у учеников, нужно снова переходить на бумажные носители.

1.2 Обзор исследований

Обзор библиотек на C# проведён С. В. Ивановой в C# используются такие библиотеки как AngleSharp, HtmlAgilityPack, CSQuery и другие [1]. В исследовании И. И. Карабак, К. А. Зорина, рассматриваются примеры парсинга в публичных сообществах мессенджера «Телеграм» в противовес парсингу решаются проблемы защиты информации от утечки [2]. Х.И. Эшонкулов в своём исследовании так же изучает проблему необходимости защиты от автоматизированного сбора информации с вебресурсов. [3]. А. А. Менщиков, А. В. Комарова, Ю. А. Гатчин рассмотрели вопрос потребности защиты от парсинга, получив всестороннее рассмотрение в исследовании [4]. А. И. Дубровина описала последствия копирования данных из интернет ресурсов, описание методов защиты описываются в статье [5].

1.3 Цель исследования

Целью данной работы является создание приложения, способного парсить тестовые вопросы из Yandex форм на языке программирования python.

2 Описание разработки чата

Для начала необходимо перейти на сайт <https://forms.yandex.ru/> и создать новую форму (рис. 1).

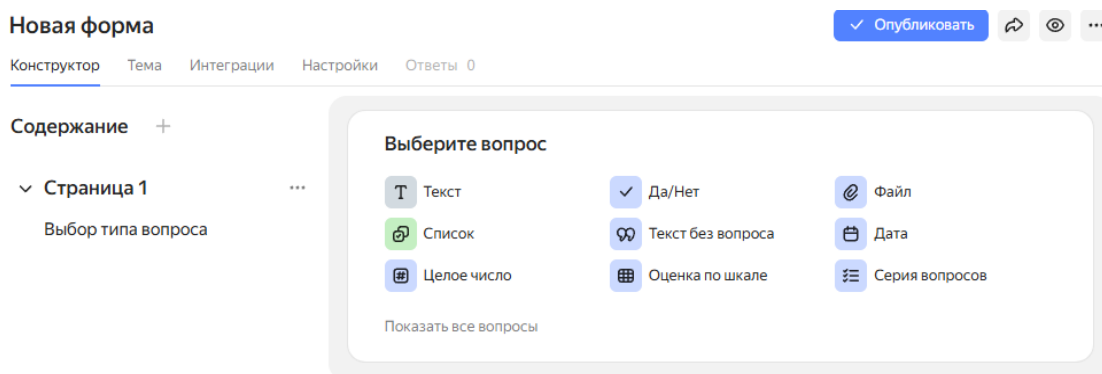


Рисунок 1 – Страница создания Яндекс формы

Далее необходимо создать несколько вопросов (рис. 2).

Содержание +

Страница 1

Какого цвета желтый лимон?

Один вариант

Обязательный

Вопрос: Какого цвета желтый лимон?

Добавить пояснение

Добавить изображение

Ответы:

Плитка

Звёзды

В один ряд

Баллы

Баллы засчитываются только за один правильный ответ.

желтого 1 балл

Идентификатор 9008949728295794

Скрытый ответ

зеленого 0 баллов

Идентификатор 1750473607147

Скрытый ответ

красного 0 баллов

Идентификатор 1750473631137

Скрытый ответ

синего 0 баллов

Идентификатор 1750473635789

Скрытый ответ

Добавить вариант

Настройки

Сортировка ответов

Без сортировки

По алфавиту

В случайном порядке для каждого пользователя

Идентификатор вопроса answer_choices_90089497282957

Скрыть вопрос

Рисунок 2

После создания формы, необходимо создать файл `parser.py`, а после открыть его через IDE PyCharm (рис. 3).

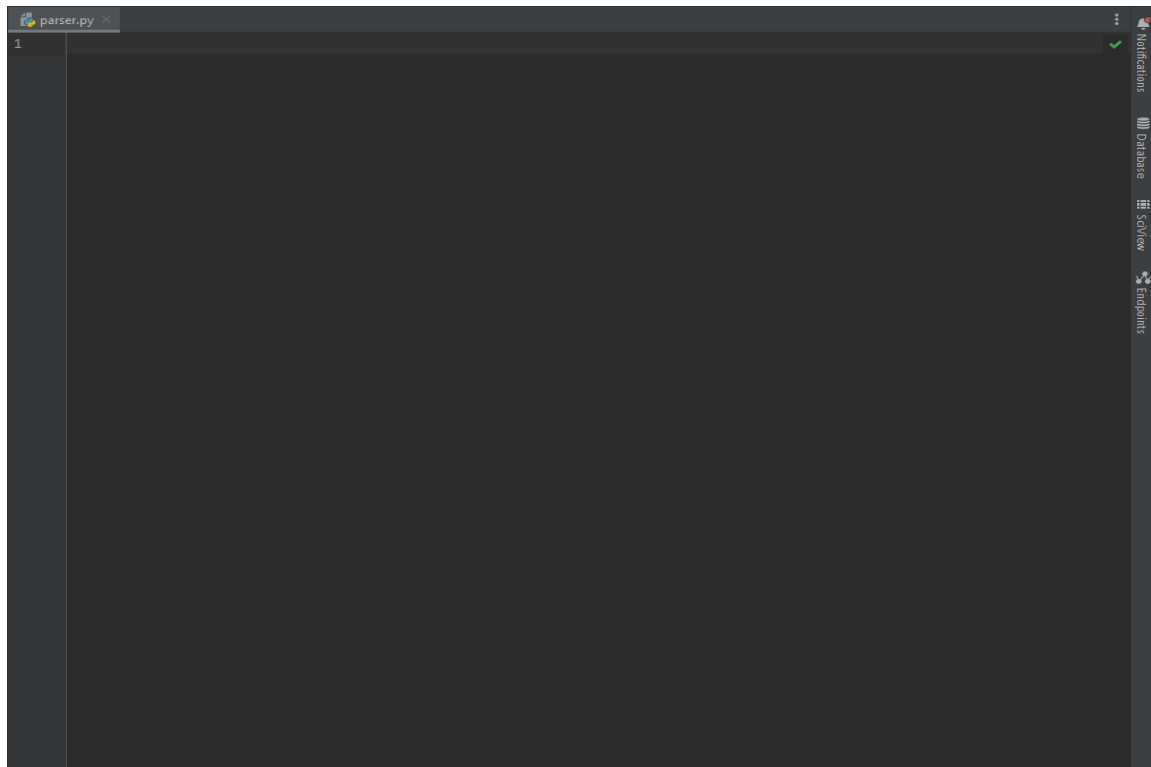


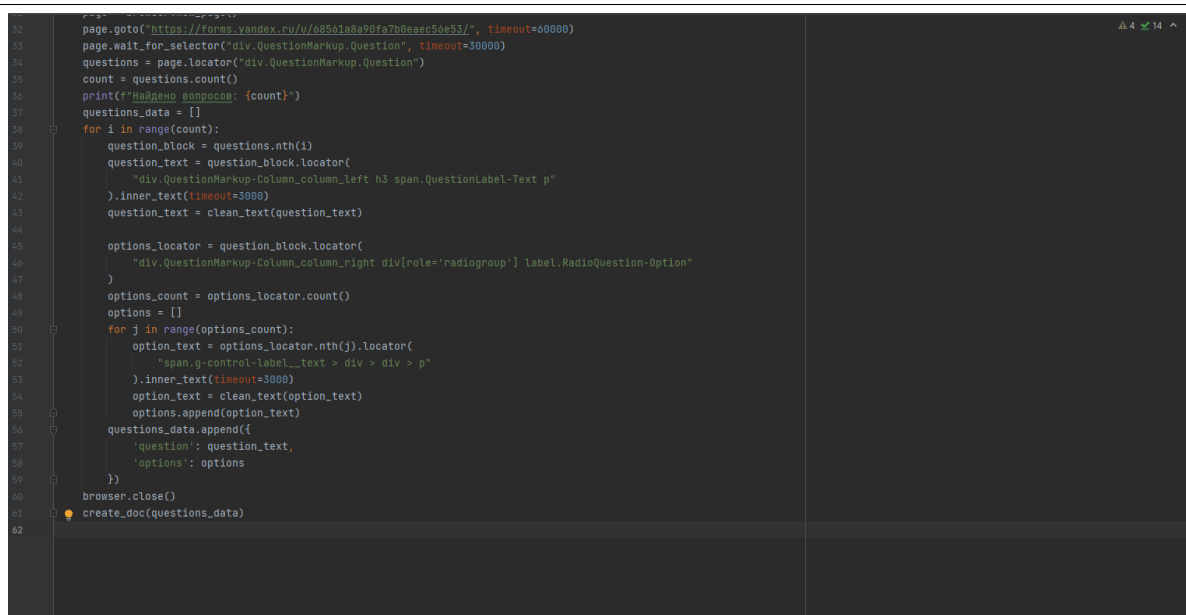
Рисунок 3 – Создание файла parser.py и открытие в PyCharm

Теперь следует установить необходимые библиотеки: `python-docx` и `playwright`.

Следующим этапом нужно написать код для парсера страницы Яндекс формы, предварительно скопировав ссылку на тест: (рис. 4-5).

```
1 from playwright.sync_api import sync_playwright
2 from docx import Document
3 from docx.shared import Pt
4 from docx.enum.text import WD_PARAGRAPH_ALIGNMENT
5
6 2 usages
7 def clean_text(text):
8     return text.strip().replace('\n', ' ')
9
10 1 usage
11 def create_doc(questions_data, filename="Тест.docx"):
12     doc = Document()
13     # Здесь прописывать шрифты для документа
14     style = doc.styles['Normal']
15     font = style.font
16     font.name = 'Arial'
17     font.size = Pt(12)
18     doc.add_heading('Тест', level=1).alignment = WD_PARAGRAPH_ALIGNMENT.CENTER
19     for idx, q in enumerate(questions_data, 1):
20         para = doc.add_paragraph()
21         run = para.add_run(f"Вопрос {idx}: {q['question']}")
22         run.bold = True
23         para.space_after = Pt(6)
24         if not q['options']:
25             doc.add_paragraph("Варианты ответа не найдены", style='Normal')
26         else:
27             for option in q['options']:
28                 doc.add_paragraph(option, style='List Bullet')
29             doc.add_paragraph()
30     doc.save(filename)
31     print(f"Файл {filename} создан.")
32 with sync_playwright() as p:
33     browser = p.chromium.launch(headless=True)
34     page = browser.new_page()
35     page.goto("https://forms.yandex.ru/u/68561a8a90fa7b0eae55e53/", timeout=60000)
36     page.wait_for_selector("div.QuestionMarkup.Question", timeout=30000)
37     questions = page.locator("div.QuestionMarkup.Question")
```

Рисунок 4 – Код для файла parser.py



```

32 page.goto("https://forms.yandex.ru/u/68561a8a96fa7b0eacc50e53/", timeout=60000)
33 page.wait_for_selector("div.QuestionMarkup.Question", timeout=30000)
34 questions = page.locator("div.QuestionMarkup.Question")
35 count = questions.count()
36 print(f"Найдено вопросов: {count}")
37 questions_data = []
38 for i in range(count):
39     question_block = questions.nth(i)
40     question_text = question_block.locator(
41         "div.QuestionMarkup-Column_column_left h3 span.QuestionLabel-Text p"
42     ).inner_text(timeout=3000)
43     question_text = clean_text(question_text)
44
45     options_locator = question_block.locator(
46         "div.QuestionMarkup-Column_column_right div[role='radiogroup'] label.RadioQuestion-Option"
47     )
48     options_count = options_locator.count()
49     options = []
50     for j in range(options_count):
51         option_text = options_locator.nth(j).locator(
52             "span.g-control-label__text > div > div > p"
53         ).inner_text(timeout=3000)
54         option_text = clean_text(option_text)
55         options.append(option_text)
56     questions_data.append({
57         'question': question_text,
58         'options': options
59     })
60 browser.close()
61 create_doc(questions_data)
62

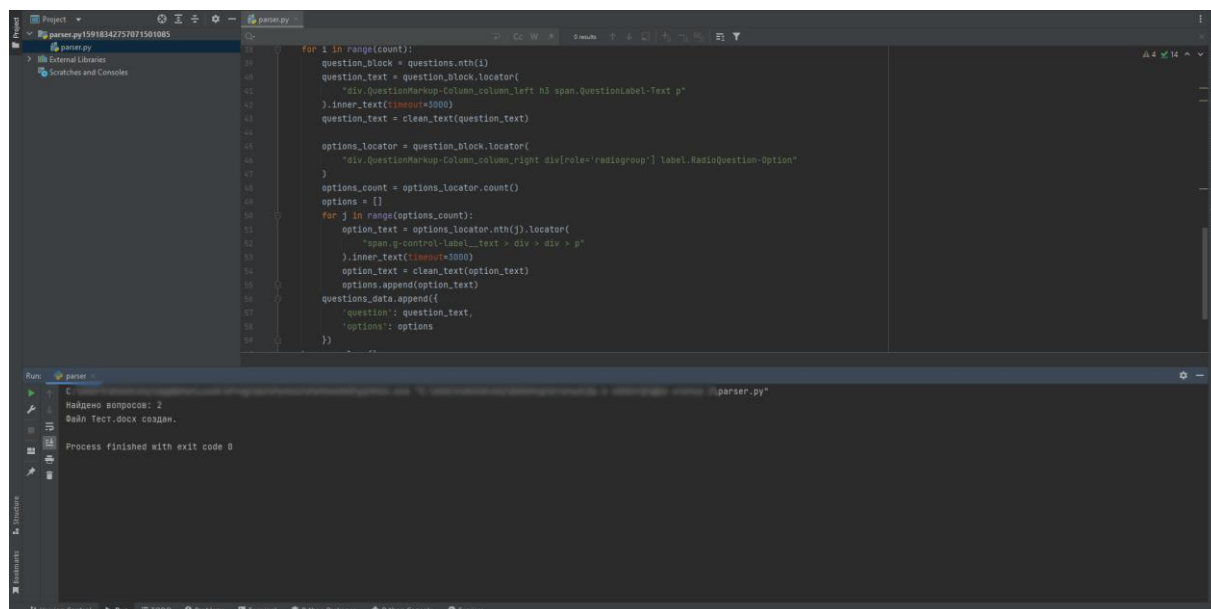
```

Рисунок 5 – Код для файла parser.py

Перед тем, как запустить код необходимо открыть командную строку в той папке, где находится проект и ввести `playwright install`.

Чтобы не срабатывала часто защита от парсинга использовался таймер на действия, предотвращая частую ошибку в консоли.

После этого можно запустить код, который спарсит все вопросы и варианты ответов в файл `Тест.docx` (рис. 6-7).



```

Run: parser
C:\Users\user>python parser.py
Найдено вопросов: 2
В файл Test.docx создан.
Process finished with exit code 0

```

Рисунок 6 – Вывод консоли IDE после запуска программы

3 Выводы

В ходе работы удалось продемонстрировать, что использование языка программирования python достаточно чтобы эффективно реализовать локальное приложение парсинга для Yandex форм. Такой подход не только

позволяет в краткие сроки собрать информацию по яндекс форме, но и облегчит преподавателям подготовку к занятиям.

Несмотря на некоторые технические сложности при настройке, локальные решения открывают новые возможности для самостоятельной разработки парсер-систем. Полученные результаты подтверждают, что автономный парсер может успешно применяться в практических задачах и является достойной альтернативой онлайн-сервисам.

Библиографический список

1. Иванова С. В. Обзор библиотек для парсинга HTML на C# // Молодежь и системная модернизация страны: сборник научных статей 4-й Международной научной конференции студентов и молодых ученых, Курск, 21–22 мая 2019 года. Курск: Юго-Западный государственный университет, 2019. С. 75-77.
1. Карабак И. И., Зорин К. А., Ажмухамедов И. М. Парсинг телеграмканалов как элемент системы автоматизированного анализа информации, полученной из сети интернет // Прикаспийский журнал: управление и высокие технологии. 2022. №. 1 (57). С. 9-17.
2. Эшонкулов Х. И. Проблемы автоматизированного сбора информации // Вестник науки и образования. 2021. № 11-2(114). С. 38-41.
3. Менщиков А. А., Комарова А. В., Гатчин Ю. А. Изучение поведения средств автоматизированного сбора информации с веб-ресурсов // Вопросы кибербезопасности. 2017. №. 3 (21). С. 49-54
4. Дубровина А. И. Парсинг, его последствия и методы предотвращения в целях защиты информации //Modern Science. 2020. №. 12-3. С. 230-233.