УДК 004.414.38
DOI 10.17513/snt.39734

МЕТОДИКА ПРОВЕДЕНИЯ СРАВНИТЕЛЬНОГО АНАЛИЗА МЕТОДОВ ПАРСИНГА ВЕБ-САЙТОВ

[1]Черепанов М.Д., [1]Жуков Н.Н., [1]Безруких А.Д., [2]Безруких Ю.А.

[1]ФГАОУ ВО «Национальный исследовательский университет ИТМО», Санкт-Петербург, e-mail: cherepp.01@gmail.com;
[2]ФГБОУ ВО «Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева», Красноярск, e-mail: expert-sib@yandex.ru

На сегодняшний день интернет является одним из крупнейших источников информации. На ранних этапах развития ИТ пользователи выполняли поиск необходимых данных и извлечение из них информации ручным методом, но данный процесс занимает большое количество времени, а также не исключено влияние человеческого фактора, из-за которого информация может быть извлечена не в полном объеме или искажена. Сейчас для автоматизации данного процесса используются программы-парсеры, благодаря чему участие человека в процессе извлечения информации сводится к минимуму. Существуют разные типы веб-ресурсов, и под каждый необходимо разрабатывать свой парсер. В интернете находится большое количество статей и руководств по созданию таких парсеров, но зачастую авторы не объясняют выбор того или иного метода парсинга, что обусловлено отсутствием информационных материалов с объяснением отличий данных методов и ситуаций, при которых они применимы. Данная методика поможет провести сравнительный анализ методов парсинга, для выявления достоинств и недостатков каждого из них, и ситуаций, при которых эффективен тот или иной метод. В рамках данной работы выполнен обзор различных веб-ресурсов в контексте загрузки данных на них, а также проведен обзор методов парсинга, которые применяются для извлечения информации с веб-ресурсов. На основе этой информации выявлены критерии для проведения сравнительного анализа методов парсинга и разработана методика его проведения.

Ключевые слова: парсинг веб-сайтов, HTTP-запросы, производительность парсера, автоматизация веб-браузера, сравнительный анализ, методика

CLASSIFICATION METHODOLOGY FOR THEMATIC MODELING RESULTS OF CANDIDATES BY TEAM ROLE

[1]Cherepanov M.D., [1]Zhukov N.N., [1]Bezrukikh A.D., [2]Bezrukikh Yu.A.

[1]ITMO University, Saint Petersburg, e-mail: cherepp.01@gmail.com;
[2]Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, e-mail: expert-sib@yandex.ru

Today, the Internet is one of the largest sources of information. In the early stages of IT development, users searched for the necessary data and extracted information from them manually, but this process takes a lot of time, and the influence of the human factor is not excluded, due to which the information may not be fully extracted or distorted. Now parser programs are used to automate this process, due to which human participation in the process of extracting information is minimized. There are different types of web resources, and for each you need to develop your own parser. There are a large number of articles and manuals on the Internet on creating such parsers, but often the authors do not explain the choice of one or another parsing method, due to the lack of information materials explaining the differences between these methods and the situations in which they are applicable. This technique will help to conduct a comparative analysis of parsing methods to identify the advantages and disadvantages of each of them, and situations in which this or that method is effective. As part of this work, a review of various web resources in the context of uploading data to them was made, as well as a review of parsing methods that are used to extract information from web resources. Based on this information, criteria for a comparative analysis of parsing methods were identified, and a methodology for its implementation was developed.

Keywords: web scraping, HTTP requests, web browser automation, parser performance, benchmarking, methodology

В настоящее время интернет является одним из крупнейших источников информации. Люди обращаются к различным веб-источникам, чтобы найти в них нужные данные, извлечь их, провести анализ и выделить необходимую информацию. Для автоматизации этого процесса используют парсеры [1].

Существуют разные типы сайтов, и поэтому под каждый тип существует определенный метод для парсинга. Интернет заполнен большим количеством статей и руководств по созданию парсеров. Но зачастую авторы не объясняют выбор того или иного метода, вследствие чего люди, пытающиеся применить выбранный метод, сталкиваются с различными проблемами и ошибками или используют не совсем эффективный метод. Это обусловлено отсутствием информационных материалов, которые бы обобщали все существующие методы, описывали отличия и ситуации, при которых эффективнее всего применим тот или иной метод.

Стоит учесть, что большинство веб-ресурсов негативно относятся к парсингу и используют различные системы защиты. Каждый метод парсинга обладает своими специфическими способами обхода таких блокировок. Необходимо понимать, что данные способы по-разному влияют на процесс парсинга, и важно знать, при каких ситуациях и с использованием какого метода парсинга каждый из этих способов применим.

Актуальность обусловлена отсутствием информационных материалов, описывающих отличия методов парсинга и ситуации, при которых они применимы.

Целью статьи является разработка методики проведения сравнительного анализа методов парсинга веб-ресурсов, для выявления ситуаций, при которых более эффективен тот или иной метод.

Материалы и методы исследования

На сегодня существует большое количество типов веб-ресурсов, например такие, как сайт-визитка, интернет-магазин, доска объявлений, блог и др. Но для парсера неважно, какой тип сайта предстоит обработать, важным является метод получения веб-страницей нужных данных, соответственно, по этому признаку можно классифицировать веб-ресурсы.

Первым вариантом может быть заранее сформированная веб-страница на стороне сервера, которая имеет в себе нужные для парсинга данные. В таком случае парсинг упрощается, так как необходим лишь список страниц, с которых будут собираться данные. Такие сайты можно классифицировать как статические в контексте получения данных. Пример такого веб-ресурса представлен на рис. 1.

При переходе между страницами на таком сайте, в инструментах разработчика видно, что для каждой страницы генерируется свой HTML-документ и все данные в него уже включены, при этом URL-адрес меняется и страница обновляется.

Следующий вариант формирования данных веб-страницы можно условно назвать «динамический». Его применяют на веб-ресурсах, данные на страницах которых могут изменяться во время просмотра. Такие типы веб-ресурсов более сложные для парсинга, так как необходимо понять условия, при которых происходит загрузка новых данных. На рис. 2 представлен пример сайта, данные на который загружаются и обновляются во время нахождения пользователя на странице.

В данном примере видно, что цена предмета обновляется автоматически, при этом в инструментах разработчика можно видеть, что данные загружаются из XHR-запроса [2].

Далее рассмотрим подробнее методы парсинга:

1. Использование официальных API [3]. Многие веб-сайты предоставляют API (Application Programming Interface) для доступа к своим данным. Этот подход позволяет получать структурированную информацию, соответствующую заданным параметрам и ограничениям, что облегчает процесс парсинга. API предоставляет разработчикам программный интерфейс для взаимодействия с веб-ресурсом и получения данных в удобном формате, часто в формате JSON или XML. При использовании официальных API разработчикам необходимо ознакомиться с документацией, предоставленной веб-сайтом, и получить доступные методы и параметры для получения нужной информации. Но не всегда целевой веб-ресурс имеет собственный API, либо присутствуют определенные платные тарифы для доступа к нему.

2. Парсинг HTML-документов. Этот метод парсинга включает получение HTML-документов с помощью HTTP-запросов, таких как GET и POST [4], к веб-ресурсам. Затем происходит анализ полученных данных для извлечения нужной информации. Для выполнения HTTP-запросов можно использовать различные инструменты и библиотеки, такие как requests в Python. Полученный HTML-код анализируется с использованием специальных инструментов для парсинга HTML, таких как BeautifulSoup или lxml [5]. С помощью этих инструментов можно найти нужные элементы на странице, извлечь текст, атрибуты, ссылки и другую информацию [6]. Но с помощью данного метода, как правило, достаточно сложно или вообще невозможно обойти защиту, применяющую встроенные на страницы Javascript функции, выполнение которых бывает обязательным для отображения каких-либо данных. Недостатком такого способа являются затраты большого количества времени на процесс парсинга, так как необходимо сперва получить HTML-документ, который иногда может иметь большой вес, а после найти нужную информацию.

3. Парсинг XHR-запросов. Некоторые веб-приложения загружают данные динамически с помощью технологии XHR (XMLHttpRequest) [2]. XHR-запросы используются для обмена данными между веб-браузером и сервером без перезагрузки страницы. Парсинг таких запросов позволяет извлекать информацию, которая не отображается непосредственно на веб-странице.

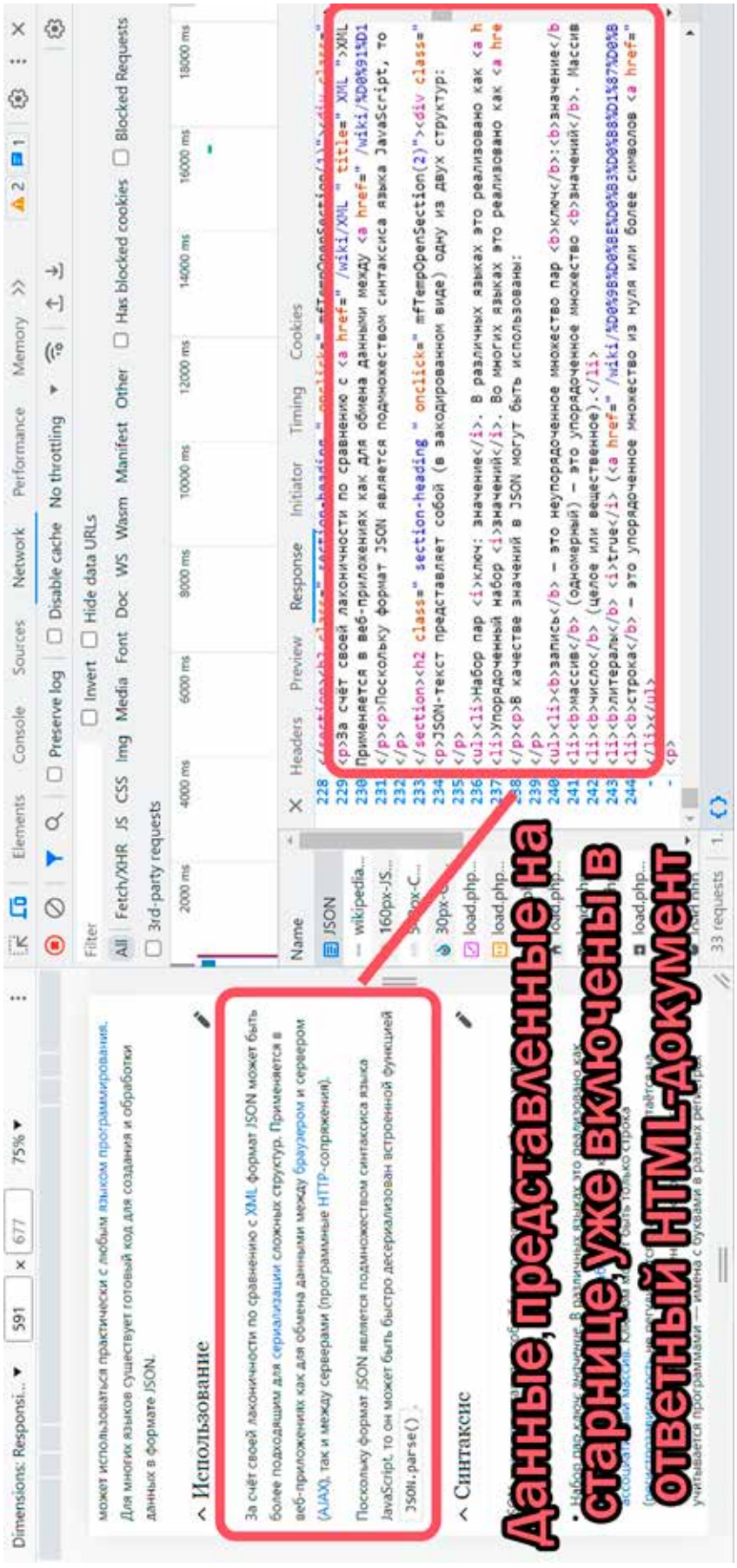

Рис. 1. Пример статического веб-ресурса

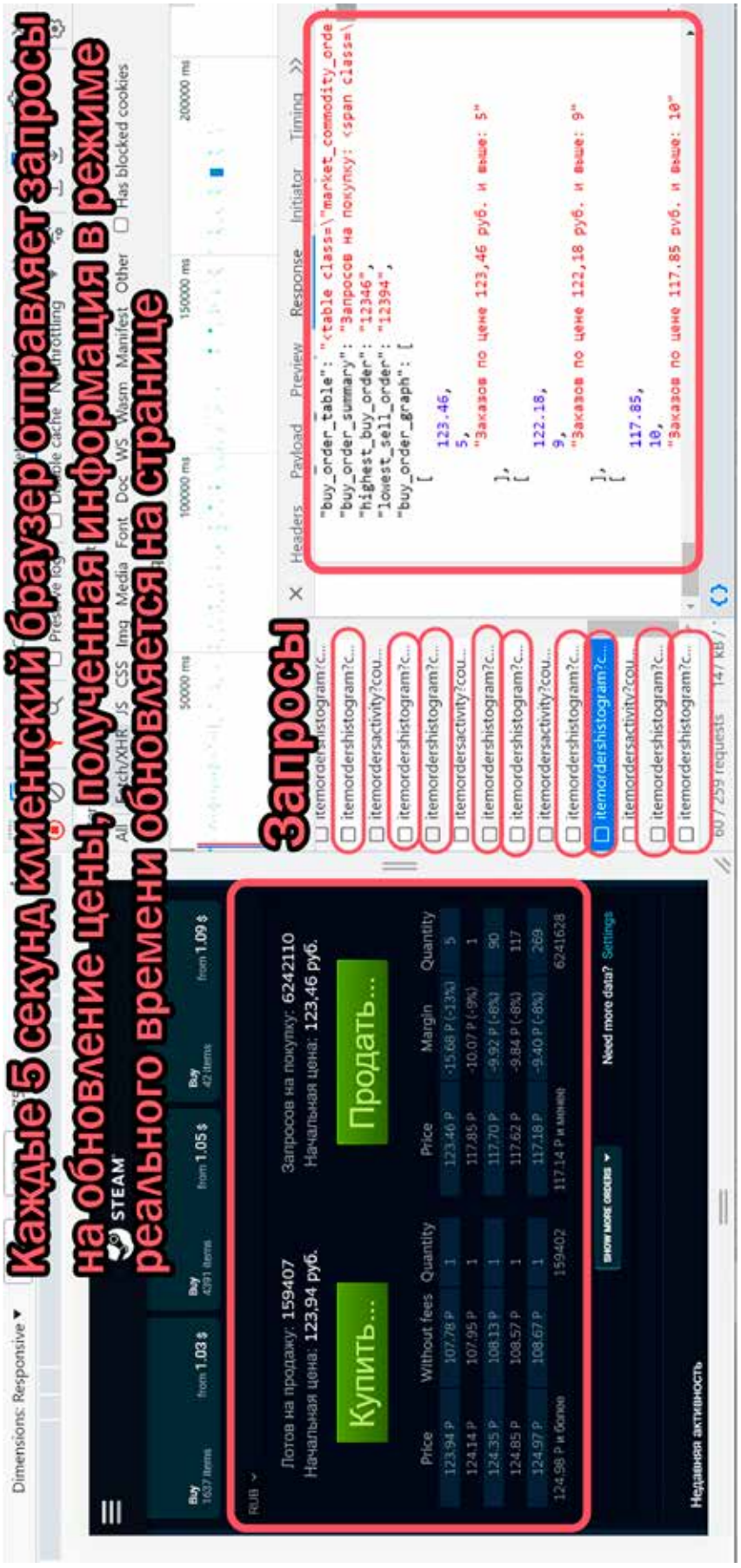

Рис. 2. Пример динамического веб-ресурса

Достоинства и недостатки методов парсинга

Классификация	Достоинства	Недостатки
API и XHR	1. Простота использования. 2. Высокая скорость получения данных. 3. Данные уже представлены в сгруппированном, отсортированном виде	1. Не всегда веб-ресурсы имеют свой собственный API. 2. Иногда доступ к API является платным
Парсинг HTML	1. Бесплатный способ	1. Большие временные затраты. 2. Легкость выявления парсера и последующая его блокировка
Парсинг XHR	1. Бесплатный способ. 2. Высокая скорость получения данных. 3. Данные уже представлены в сгруппированном, отсортированном виде	1. Легкость выявления парсера и последующая его блокировка. 2. Не всегда на веб-страницах присутствуют скрипты, генерирующие XHR запросы, которые может обработать сервер
Автоматизация веб-браузера	1. Бесплатный способ. 2. Меньше вероятность получить блокировку. 3. Возможность применения одного из вышеупомянутых методов	1. Большие временные и ресурсные затраты

Для этого необходимо проанализировать сетевой трафик, например, используя встроенные в браузер инструменты разработчика, и обнаружить XHR-запросы, отправляемые и получаемые при взаимодействии с веб-приложением. Полученные данные могут быть в формате JSON или XML, и их можно анализировать с использованием соответствующих инструментов. Этот подход более эффективный, чем предыдущий, так как зачастую такие запросы обращаются к специальным маршрутам веб-ресурса, которые получают различные параметры фильтрации данных и возвращают их в чистом виде. При этом вес передаваемых документов гораздо меньше, чем документы формата HTML, это уменьшает количество времени на загрузку, поиск и извлечение необходимых данных.

Парсеры, разработанные при помощи описанных выше методов 2 и 3, являются самыми распространенными, но и в то же время легко обнаруживаемыми, так как взаимодействие происходит через простые и однообразные HTTP-запросы, из-за чего создается нестандартная активность, которая легко распознается системами защиты от парсинга.

4. Автоматизация веб-браузера. Этот подход включает использование инструментов, таких как Selenium WebDriver, для автоматизации действий веб-браузера. Selenium WebDriver позволяет программно контролировать веб-браузер и эмулировать действия пользователя, такие как клики, заполнение форм, прокрутка страницы и другие взаимодействия. При использовании автоматизации веб-браузера можно получить доступ к данным, которые доступны только через взаимодействие с веб-ресурсом, такие как динамически загружаемые элементы или взаимодействие с JavaScript [7]. Selenium WebDriver позволяет программировать скрипты на различных языках программирования для автоматизации веб-браузера и извлечения нужной информации. Данный метод не является самым быстрым, но можно точно сказать, что он является более эффективным, чем вышеупомянутые [8], не считая метод парсинга через API, так как целевой веб-ресурс будет считать парсер, разработанный с использованием данного метода, за реального пользователя.

При рассмотрении данных методов были выявлены достоинства и недостатки каждого, которые представлены в таблице.

Каждый из этих методов имеет свои преимущества, недостатки и ограничения [9], и выбор конкретного метода зависит от требований проекта и доступности данных. Рассмотренные выше методы будут в дальнейшем использованы для выявления критериев для сравнительного анализа. Также алгоритмы применения данных методов на различные типы веб-ресурсов будут использоваться для проведения экспериментов, с целью получения данных, на основе которых будет проводиться сравнительный анализ.

Результаты исследования и их обсуждение

Так как целью работы парсера является извлечение информации, основные критерии будут связаны со скоростью и качеством выполнения этого процесса, а именно:

1. Скорость извлечения информации. Данный параметр будет измерять в секун-

дах время, которое потребуется на извлечение данных, например, со 100 страниц, HTML или JSON документов.

2. Затраченное количество вычислительных ресурсов. Данный параметр будет показывать количество затраченной оперативной памяти и вычислительной мощности процессора в процентах.

3. Блокировка парсера. Велика вероятность, что при парсинге при помощи методов парсинга HTML-, JSON-документов или автоматизации браузера может произойти блокировка со стороны целевого веб-ресурса, что заставит использовать различные способы обхода данных блокировок. Поэтому еще одним критерием будет временная величина, характеризующая время активной работы парсера до его блокировки. При этом все эксперименты должны проводиться в одинаковых условиях, то есть запросы будут направлены на один и тот же веб-ресурс, и ни один из способов обхода систем защиты от парсеров не будет применен. Данный критерий опциональный, так как, возможно, некоторые из тестируемых веб-ресурсов не будут иметь подобные системы защиты.

4. Обход систем защиты. Еще один опциональный критерий, который позволит оценить возможность обхода систем защиты от парсинга, если такие будут присутствовать. Важно заметить, что способы обхода должны быть бесплатными и общедоступными. К таким способам, например, относятся временные паузы между запросами, бесплатные прокси [10].

Анализ и оценка результатов, полученных на основе этих критериев, позволит выявить признаки веб-ресурсов, которые поддержат принятие решения о выборе того или иного метода парсинга.

Сама методика проведения сравнительного анализа для всех методов парсинга будет состоять из нескольких этапов.

На первом этапе необходимо будет выбрать несколько разных веб-ресурсов. Каждый веб-ресурс должен относиться к нескольким из этих классов или в лучшем случае ко всем:

1. Веб-ресурс, который имеет документированное API.

2. Веб-ресурс, информация на страницах которого статична, то есть генерируется на стороне сервера и не изменяется во время нахождения пользователя на ней.

3. Веб-ресурс, информация на страницах которого динамична, то есть загружается с сервера во время нахождения пользователя на странице.

4. Веб-ресурс, имеющий защиту от парсинга.

На следующем этапе будет разработан парсер для каждого выбранного веб-ресурса, используя один из методов парсинга, которые были рассмотрены выше. Поскольку каждый веб-ресурс может иметь свою уникальную структуру и особенности и не подразумевается, что парсеры будут универсальными, то потребуется разработать специальную версию парсера для каждого сайта.

Затем каждый разработанный парсер будет применен к соответствующему веб-ресурсу, и производительность каждого метода парсинга будет оценена и измерена согласно установленным критериям, описанным выше. Важно заметить, что при наличии систем защиты от парсинга на целевом веб-ресурсе, необходимо будет, если это возможно, используя доступные способы обхода таких систем, настроить парсер, чтобы не получить блокировку.

После проведения экспериментов и получения результатов необходимо провести анализ данных, целью которого будет выявить особенности каждого веб-ресурса, которые могли бы помочь с определением эффективного метода парсинга именно на данном веб-ресурсе.

Заключение

В данном исследовании был проведен обзор различных видов веб-ресурсов в контексте загрузки данных на них. Ресурсы были классифицированы на статические, где страницы с данными предварительно генерируются на стороне веб-сервера, и динамические, где загрузка данных происходит в реальном времени в момент нахождения пользователя на странице.

Также был проведен обзор методов парсинга, которые используются для извлечения информации с веб-ресурсов. Рассмотрены основные методы, такие как использование официальных API, парсинг HTML-документов, анализ XHR-запросов и автоматизация веб-браузера. Каждый метод имеет свои преимущества и ограничения, и их выбор зависит от конкретных характеристик веб-ресурса и требуемой функциональности парсера.

Далее были выявлены критерии, которые будут использоваться для проведения сравнительного анализа методов парсинга. Основные критерии включают скорость извлечения информации и затраченное количество вычислительных ресурсов. Также рассмотрены опциональные критерии, такие как время работы парсера до блокировки и возможность обхода систем защиты от парсинга.

В результате разработана методика проведения сравнительного анализа. Эксперименты, которые необходимо провести, состоят из вариации типов веб-ресурсов и методов парсинга по определенному алгоритму, который включает в себя такие этапы, как разработка парсера, тест разработанного парсера и, при необходимости, настройка парсера для обхода систем защиты с применением общедоступных способов обхода (временные паузы, прокси и другие).

Список литературы

1. Ермоленко А.В., Котелина Н.О., Старцева Е.Н., Юркина М.Н. О востребованности подготовки в области парсинга данных для web-разработчиков // Вестник Сыктывкарского университета. Серия 1. Математика. Механика. Информатика. 2021. № 1 (38). С. 56–69.

2. Сигалов Д.А., Хашаев А.А., Гамаюнов Д.Ю. Обнаружение серверных точек взаимодействия в веб-приложениях на основе анализа клиентского javascript-кода // Прикладная дискретная математика. 2021. № 53. С. 32–54.

3. Карабак И.И., Зорин К.А., Ажмухамедов И.М. Парсинг телеграм-каналов как элемент системы автоматизированного анализа информации, полученной из сети интернет // Прикаспийский журнал: управление и высокие технологии. 2022. № 1 (57). С. 9–17.

4. Меньшиков Я.С. Преимущества автоматического сбора данных в сети интернет над ручным сбором данных // Universum: технические науки. 2022. № 10 (103). URL: https://7universum.com/ru/tech/archive/item/14383 (дата обращения: 05.07.2023).

5. Вильданов Т.Э., Иванов Н.С. Анализ инструментов парсинга и веб-скрейпинга в рамках разработки арбитражной инвестиционной стратегии на рынке спортивных ставок // Скиф. Вопросы студенческой науки. 2021. № 5 (57). С. 23–33.

6. Нгуен Тхань Вьет, Кравец А.Г. Алгоритм работы веб-краулера для решения задачи сбора данных из открытых интернет-источников // Известия Санкт-Петербургского государственного технологического института (технического университета). 2019. № 51 (77). С. 115–119.

7. Корепанова А.А., Бушмелев Ф.В., Сабреков А.А. Технологии парсинга Node.js в задаче агрегации сведений и оценки параметров грузовых маршрутов посредством извлечения данных из открытых источников // Компьютерные инструменты в образовании. 2021. № 3. С. 41–56. DOI: 10.32603/2071-2340-2021-3-41-56.

8. Суханов А.А., Маратканов А.С. Анализ способов сбора социальных данных из сети Интернет // International scientific review. 2017. № 1 (32). С. 25–28.

9. Крамаров С.О., Овсянников В.А., Сахарова Л.В., Усатый Р.С., Лукьянова Г.В. Автоматизированный сбор данных ключевых финансовых показателей предприятий IT-отрасли региона // Вестник кибернетики. 2022. № 3 (47). С. 39–45. DOI: 10.34822/1999-7604-2022-3-39-45.

10. Москаленко А.А., Лапонина О.Р., Сухомлин В.А. Разработка приложения веб-скрапинга с возможностями обхода блокировок // Современные информационные технологики и ИТ-образование. 2019. Т. 15, № 2. С. 413–420. DOI: 10.25559/SITITO.15.201902413-420.