ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗИРОВАННОГО СБОРА ИНФОРМАЦИИ HTML-САЙТА ПОСРЕДСТВОМ ТЕХНОЛОГИИ RUBY

Страчинский Н.С.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Сурков К.А. – ст. преп.

Выполнен анализ задач, решаемых автоматизированным сбором информации. Произведён анализ пригодности способов чтения данных для сбора информации HTML-сайта. Описаны преимущества расширения браузера для формирования наборов команд автоматизированного сбора информации HTML-сайта. Рассмотрена инфраструктура фоновых задач Sidekiq и синтаксического анализатора Nokogiri для выполнения поставленной задачи на языке программирования Ruby.

Процесс автоматизированного сбора информации, или же парсинг, заключается в получении и систематизировании данных из различных источников. С его помощью можно решать такие задачи, как сравнение цен и оценка конкуренции, обучение и тестирование моделей машинного обучения, наполнение каталогов информацией из открытых источников, извлечение данных о продукте или услугах третьих лиц для платформы электронной коммерции [1]. HTML-документ использует предопределённые теги и атрибуты, чтобы указать браузеру, как отображать контент.

Канонически для автоматизированного сбора информации приняты интерфейсы DOM (Document Object Model) и SAX (Simple API for XML). DOM предоставляет дерево объектов, которое содержит всю информацию о документе, он удобен для произвольного доступа к различным местам контекста данных, но при этом не экономит память, поскольку ему требуется читать весь документ целиком и сохранять дерево элементов в памяти. Интерфейс SAX же управляет событиями. Поэтому с его помощью проблематично получить произвольный доступ к различным местам контекста данных. SAX экономит память, поскольку приложение может хранить лишь ту часть документа, которая представляет интерес [2].

Среди требований к интерфейсам автоматизированного сбора информации можно выделить произвольный доступ к различным местам контекста данных для упрощения получения лишь требуемой информации. Также следует учитывать наличие в анализируемом HTML-документе как одинарных, так и парных тегов, присутствующих в контексте данных. Таким образом, можно прийти к выводу о пригодности DOM интерфейса для автоматизированного сбора информации HTML-сайта. Для произвольного доступа к объектам DOM дерева разумно использовать язык запросов XPath [3]. Для работы с рассмотренным языком запросов существует программная библиотека для анализа XML и HTML документов, работающая на языке программирования Ruby – Nokogiri. Данный синтаксический анализатор позволяет читать и изменять DOM дерево.

Набор команд для чтения информации с помощью XPath следует задать и хранить. Среди уверенных пользователей компьютера достаточно людей, навыков которых не хватает для написания кода, поэтому большую популярность набирают nocode-подходы. Для разработки ИТ-продуктов с данным подходом требуется ПО, позволяющее визуально моделировать события.

Таким образом, программное средство автоматизированного сбора информации должно взаимодействовать с HTML-сайтом и при этом иметь пользовательский интерфейс, подходящий для визуального моделирования событий. Для вышеупомянутых целей в качестве ПО как можно лучше подходит расширение браузера. Данный тип программ не только является веб-приложением с пользовательским интерфейсом, но и способен встраивать код и выполнять его в открытом в браузере сайте. Следовательно, пользователь имеет возможность в режиме реального времени взаимодействовать с DOM-объектами для выбора требуемых для сбора HTML-элементов.

Составленный в пользовательском интерфейсе набор команд должен постоянно выполняться и собирать информацию для сохранения актуальности данных. Для данной цели предусмотрена инфраструктура Sidekiq, позволяющая управлять фоновыми задачами, которые можно реализовать на языке программирования Ruby.

Таким образом, можно сделать вывод о целесообразности создания программного средства автоматизированного сбора информации HTML-сайта, пользовательский интерфейс которого будет представлять из себя расширение для браузера, а внутренняя реализация – программный код, который в очереди считывает и запускает набор команд, предоставленный пользователем.

Список использованных источников:

1 What is Web Scraping used for? [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.webharvy.com/articles/web-scraper-use-cases.html.

2. XML Parsing, SAX/DOM [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ranger.uta.edu/~cli/pubs/2009/XMLParsing_ChengkaiLi.pdf.

3. Парсим сайт при помощи XPath [Электронный ресурс]. – Электронные данные. – Режим доступа: https://siteclinic.ru/blog/seo-instrumenty/parsim-sajt-s-xp