

системой пользователей не будет превышать пяти. Также снижает объём передаваемой по сети информации предложенный формат обмена: пакет включает только изменённые данные.

При изменении значений часов и форм контроля в семестрах дисциплин, а также добавления и удаления дисциплин необходим пересчёт всех итоговых показателей учебного плана, что влечёт за собой увеличение затрат времени на вычисления. Задача была решена с использованием селекторов. Селектор – это функция, которая принимает состояние Redux в качестве аргумента и возвращает данные, полученные из этого состояния для обновления визуального компонента. Селектор не пересчитывается, пока не изменится состояние части хранилища, за которым он следит. Таким образом, при изменении данных в общем хранилище приложения, селектор не допустит перерисовки связанного с ним визуального компонента, если эти изменения не затронули отображаемые с компоненте данные.

Использование сотрудниками учебного отдела данного модуля информационной системы университета позволит сформировать банк данных для обеспечения информацией другие части системы, такие как модуль распределения нагрузки по кафедрам, управления нагрузкой заведующим кафедрой, распределение аудиторного фонда.

Список использованных источников

1. Казаков, В. Е. Разработка back-end приложения «Учебно-методический отдел (учебные планы)» / М. С. Карнилов, В. Е. Казаков // Материалы докладов 55-й Международной научно-технической конференции преподавателей и студентов: в 2 т. / УО «ВГТУ». – Витебск, 2022. – Т. 2. – с. 5–7
2. Сайт «react.dev» [Электронный ресурс] <https://react.dev/learn> – Дата доступа: 12.05.2023.
3. Сайт «ag-grid.com» [Электронный ресурс] / Spring Framework – Режим доступа: <https://www.ag-grid.com/> – Дата доступа: 12.05.2023.

УДК 004.4

ЭКСПОРТ ДАННЫХ ИЗ FRONT-END ПРИЛОЖЕНИЯ В XLSX-ФОРМАТ

*Поздняков К.В., студ., Жизневский В.А., к.ф.-м.н., доц., Казаков В.Е., к.т.н., доц.
Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. В статье представлен обзор программных средств, предназначенных для извлечения данных из front-end приложения в формат *xlsx*.

Ключевые слова: импорт, XLSX, JavaScript, библиотека.

Существует несколько способов экспортировать данные из интерфейсного приложения в формат файла Excel (XLSX).

Вот несколько методов:

1. Использование библиотеки JavaScript. Существует множество доступных библиотек JavaScript, таких как *xlsx* и др, которые могут помочь вам экспортировать данные из вашего интерфейсного приложения в файл Excel. Эти библиотеки предоставляют API для преобразования данных JSON в формат XLSX и могут быть интегрированы в ваше приложение.

2. Использование серверного API. Если ваше интерфейсное приложение взаимодействует с серверным API, вы можете отправить запрос в API для экспорта данных в файл Excel. Затем серверный API может сгенерировать файл Excel и отправить его обратно во внешнее приложение.

3. Использование стороннего сервиса. Доступно несколько сторонних сервисов, которые позволяют экспортировать данные в формат Excel. Эти службы обычно предоставляют API, которые могут быть интегрированы в ваше приложение.

Рассмотрим библиотеку *xlsx* – это популярная библиотека JavaScript, используемая для работы с файлами Excel. Библиотека предоставляет ряд функций для работы с файлами Excel, включая:

1. Чтение и запись файлов Excel в различных форматах, таких как XLSX, XLS и CSV.
2. Работа с отдельными листами, строками и ячейками в файлах Excel.
3. Преобразование между различными типами данных и типами данных Excel.

4. Синтаксический анализ и генерация формул в файлах Excel.
5. Форматирование и оформление файлов Excel.
6. Извлечение метаданных и свойств из файлов Excel.

Библиотека `xlsx` широко используется в веб-разработке и может быть интегрирована с популярными фреймворками, такими как `React`, `Vue` и `Angular`. Это библиотека с открытым исходным кодом, которая активно поддерживается и обновляется сообществом.

Одним из преимуществ использования библиотеки `xlsx` является ее простота в использовании и гибкость. Он предоставляет простой API, который позволяет разработчикам быстро считывать и записывать данные в файлы Excel без необходимости использования сложных библиотек Excel или API-интерфейсов.

API библиотеки реализует известный подход к работе с `xlsx`-документом на основе его DOM. Кроме того имеются возможности: быстрого формирования листа на основе `json`-объекта; стилизацию ячеек с помощью `js`-объектов; добавлять объекты DOM в документ с помощью удобной функции `WriteFile`.

Для сохранения файла используется библиотека `file-saver`.

Есть несколько преимуществ использования библиотеки `xlsx` в приложении `React`:

1. Знакомый интерфейс. Библиотека `xlsx` предоставляет знакомый интерфейс для работы с данными электронных таблиц, облегчая работу разработчикам, которые уже знакомы с Excel или Google Sheets.
2. Кроссплатформенная совместимость. Библиотека `xlsx` работает на нескольких платформах, включая Windows, Mac и Linux, что делает ее надежным выбором для разработки кроссплатформенных приложений `React`.
3. Настраиваемый. Библиотека `xlsx` предоставляет ряд опций для настройки функциональности экспорта, позволяя разработчикам адаптировать процесс экспорта к своим конкретным потребностям.
4. Производительность. Библиотека `xlsx` обладает высокой производительностью, позволяя разработчикам быстро и эффективно экспортировать большие наборы данных. Это важно для приложений, которые имеют дело с большими объемами данных.

Существуют некоторые потенциальные недостатки использования библиотеки `xlsx` в приложении `React`:

1. Крутая кривая обучения: библиотека `xlsx` предоставляет знакомый интерфейс для работы с данными электронных таблиц, разработчикам, которые не знакомы с Excel или Google Sheets, все равно может потребоваться много времени для её освоения.
2. Ограниченная функциональность: библиотека `xlsx` предоставляет базовую функциональность для экспорта данных в формат XLSX, она может не подходить для приложений, которым требуется более продвинутая функциональность электронных таблиц.
3. Совместимость с браузером: библиотека `xlsx` опирается на браузерные API, такие как `Blob` и `FileReader`, которые могут не поддерживаться старыми браузерами. Это может ограничить совместимость приложения со старыми браузерами.
4. Проблемы безопасности: библиотека `xlsx` может быть подвержена уязвимостям системы безопасности, таким как атаки XSS.

Существует несколько аналогов или альтернатив библиотеке `xlsx` для front-end приложений. Вот несколько вариантов:

- 1) `react-excel-workbook`: легкий компонент `React`, который позволяет экспортировать данные в формат Excel с использованием простой разметки `JSX`. Он прост в использовании и легко настраивается;
- 2) `react-data-export`: компонент `React`, который позволяет экспортировать данные в несколько форматов, включая Excel, CSV и JSON. Он предоставляет широкий спектр вариантов настройки для экспорта данных;
- 3) `react-csv`: облегченный компонент `React`, который позволяет экспортировать данные в формат CSV. Он предоставляет опции для настройки формата файла CSV и разделителя;
- 4) `react-xlsx`: оболочка `React` для библиотеки `xlsx`, которая предоставляет простой интерфейс для экспорта данных в формат XLSX. Он прост в использовании и легко настраивается;
- 5) `exceljs`: мощная библиотека JavaScript для работы с файлами Excel, которую можно использовать в приложении `React`. Он предоставляет ряд расширенных функций для работы с файлами Excel, включая форматирование, формулы и диаграммы.

Библиотека `react-excel-workbook` представляет собой оболочку библиотеки `xlsx`, которая упрощает создание файлов Excel в приложениях `React`.

Основная идея, реализованная в библиотеке, состоит в возможности формирования листов и книг Excel в виде react-компонента, при его отрисовке react сгенерирует xlsx документ и разместит в месте отрисовки кнопку для его скачивания.

Рассмотренная связка библиотек будет хорошей альтернативой применяемому в данный момент разработчиками информационной системы ВГТУ способу формирования отчётов на стороне серверной части с последующей его передачей через локальную сеть университета.

Список использованных источников

1. SheetJS [Электронный ресурс]. – Режим доступа: <https://www.npmjs.com/package/xlsx>. – Дата доступа: 14.04.2023
2. react-excel-workbook [Электронный ресурс]. – Режим доступа: <https://softwareconnect.com/advanced-planning-scheduling/>. – Дата доступа: 14.04.2023

УДК 004.415.25

РАЗРАБОТКА ПРИЛОЖЕНИЙ С REST API НА PYTHON

Антонова Т.А., студ., Бизюк А.Н., ст. преп.

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. В статье рассмотрены вопросы создания веб-приложений на языке Python. Язык Python очень активно развивается и технологии создания веб-приложений на этом языке имеют хорошие перспективы. Фреймворк Django значительно упрощает создание программных интерфейсов для веб-приложений.

Ключевые слова: веб-сервис, REST, фреймворк, Python, Django.

Веб-сервис (англ. web service) – идентифицируемая веб-адресом программная система со стандартизированными интерфейсами. Веб-службы могут взаимодействовать друг с другом и со сторонними приложениями посредством сообщений, основанных на определённых протоколах (XML, JSON и т. д.). Веб-служба является единицей модульности при использовании сервис-ориентированной архитектуры приложения.

Одним из подходов создания веб сервиса является REST.

REST (сокр. англ. Representational State Transfer, «передача состояния представления») – стиль построения архитектуры распределенного приложения. Данные в REST должны передаваться в виде небольшого количества стандартных форматов (например HTML, XML, JSON). Сетевой протокол (как и HTTP) должен поддерживать кэширование, не должен зависеть от сетевого слоя, не должен сохранять информацию о состоянии между парами «запрос-ответ». Утверждается, что такой подход обеспечивает масштабируемость системы и позволяет ей эволюционировать с новыми требованиями [1].

На данный момент можно найти фреймворк для создания приложений в стиле REST практически для каждого языка программирования, используемого в веб-разработке.

На сегодняшний день наиболее функциональным фреймворком для создания веб-приложений на языке Python является фреймворк Django [2]. Django можно назвать MVC-фреймворком, так он реализует взаимодействие пользователя и системы:

- Model (хранит данные пользователя);
- View (отображает данные пользователя);
- Controller (принимает изменения данных от пользователя).

Внутри Django эта терминология звучит немного иначе (Model, View, Template), но суть остается той же.

Фреймворк не просто так пользуется популярностью среди бизнеса и разработчиков.

Преимущества использования Django:

- безопасность;
- защита от SQL-инъекций, межсайтового скриптинга (XSS) и подделки межсайтовых запросов (CSRF);