

Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 21.Б08-мм

Вывод метрических функциональных зависимостей в веб-интерфейсе Desbordante

Белокопный Сергей Александрович

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
Ассистент кафедры информационно-аналитических систем Г. А. Чернышев

Санкт-Петербург
2023

Оглавление

Введение	3
1. Постановка задачи	4
2. Вводная информация	5
2.1. DESBORDANTE	5
2.2. GraphQL	5
2.3. Обзор технологий создания веб-приложений	6
2.4. Метрические функциональные зависимости	7
2.5. Веб-приложение DESBORDANTE	7
3. Обзор решения	8
4. Практическая часть	9
4.1. Реализация	9
4.2. Применение	9
Заключение	10
Список литературы	11

Введение

Удобный для чтения и понимания вывод результата программы позволяет пользователю быстро и легко сделать вывод о входных данных. Так же хорошо оформленный вывод результатов работы программы повышает её презентабельность и может привлечь новых пользователей.

Вывод результата работы программы может быть сделан с помощью диаграмм, схем или таблиц. Но без возможности отобразить результат работы перед пользователем, использование этой программы станет невозможным.

Популярными способами вывода результата являются вывод в консоль, в приложении или на веб-сайт. Для написания веб-сайта используются фреймворки и библиотеки, которые упрощают написание кода, процесс получения данных с сервера и вывода их на экран.

Разрабатывая интерфейс для вывода результата работы алгоритма в Desorbante важно помнить об удобстве пользования, об архитектуре веб-сайта и о последующем расширении функционала. Нельзя забывать, что от удобства интерфейса зависит презентабельность и виральность проекта, а выбор правильной архитектуры позволит легко его поддерживать.

1. Постановка задачи

Целью работы является создание интерфейса для работы с алгоритмом валидации метрических функциональных зависимостей в веб-приложении DESBORDANTE¹. Для её выполнения были поставлены следующие задачи:

1. реализовать запросы для получения данных о задаче с сервера с помощью языка запросов GraphQL;
2. реализовать систему хранения и получения информации о задаче и её результате;
3. реализовать интерфейс настройки алгоритма;
 - создание набора компонентов для создания формы настройки алгоритма;
 - создание запроса для получения информации об входном наборе данных;
 - добавление формы настройки алгоритма к уже существующим формам;
4. реализовать интерфейс отображения результатов работы;
 - создание страницы для отображения результатов работы;
 - создание компонента для вывода интерактивной таблицы.

¹<https://github.com/Mstrutov/Desbordante> (дата доступа: 8 января 2023 г.).

2. Вводная информация

Профилирование данных — это процесс анализа данных, целью которого является получение метаданных.

Метаданные — любая сопутствующая информация о данных. Например: размер файла, дата создания или условия, при которых этот файл был создан. Также метаданными являются закономерности в данных. Валидация этих закономерностей необходима в различных областях науки, таких как физика или медицина, для подтверждения или, наоборот, опровержения гипотез.

2.1. DESBORDANTE

DESBORDANTE — профайлер данных, который может обнаруживать различные зависимости в табличных данных с помощью примитивов. Примитив — набор алгоритмов, с помощью которых проверяются закономерности. Ядро DESBORDANTE написано на C++ и работа с ним может вестись через консоль или через веб-приложение с простым в использовании интерфейсом. Больше информации можно найти в блоге UNIDATA [3, 4].

2.2. GraphQL

GraphQL — язык запросов к API сервиса. Он позволяет разработчику получать конкретные данные с помощью настраиваемых запросов, в которых указываются конкретные поля данных, необходимые разработчику, и аргументы, по которым выполняется выборка данных. Это позволяет ускорить работу веб-приложения за счёт уменьшения количества передаваемых данных. Подробнее о GraphQL можно прочитать в статье [2] и на официальном сайте [7].

2.3. Обзор технологий создания веб-приложений

Для созданий веб-приложений используется язык программирования JAVASCRIPT. Для облегчения разработки веб-приложений вместе с JAVASCRIPT может использоваться надстройка TYPESCRIPT, разработанная компанией MICROSOFT, которая добавляет строгую типизацию в JAVASCRIPT, и специальные фреймворки. Далее рассмотрим наиболее популярные из них:

- REACTJS — JAVASCRIPT-библиотека с открытым исходным кодом, разработанная компанией FACEBOOK, используемая для разработки пользовательских интерфейсов. Цель этой библиотеки — предоставить высокую скорость разработки и масштабируемость. Часто REACTJS используют с другими библиотеками, такими как APOLLO GRAPHQL и NEXTJS. Последняя библиотека позволяет использовать SSR (Server Side Rendering — отрисовка на стороне сервера). Подробности можно узнать на главных страницах REACTJS [9], APOLLO GRAPHQL [6] и NEXTJS [8].
- ANGULAR — платформа для разработки веб-приложений на основе MVC-шаблона (Model-View-Controller — Модель-Представление-Контроллер), созданная командой из GOOGLE написанная на TYPESCRIPT. ANGULAR был разработан с использованием идей декларативного программирования для создания пользовательских интерфейсов, так как разработчики убеждены в том, что этот подход лучше всего подходит для этих целей. Подробности можно узнать на официальном сайте [5].
- VUE.JS — JAVASCRIPT-фреймворк с открытым исходным кодом, разработанный для быстрого прототипирования пользовательских интерфейсов. Подробности можно узнать на официальном сайте [10].

Так как в проекте уже использовались REACTJS с TYPESCRIPT и NEXTJS, было решено использовать именно их.

2.4. Метрические функциональные зависимости

Функциональные зависимости позволяют установить зависимости между множествами атрибутов таблицы. Но в данных могут встречаться некоторые вариации и из-за этого может быть невозможно зафиксировать точные взаимосвязи. Эту проблему можно решить используя метрики для усреднения данных, а после уже использовать функциональные зависимости [1].

2.5. Веб-приложение DESBORDANTE

Процесс создания и настройки задачи в DESBORDANTE состоит из нескольких шагов:

1. Выбор примитива — например Functional Dependencies, Conditional Functional Dependencies, Association Rules, Error Detection Pipeline или Metric Verification.
2. Выбор набора данных (датасета).
3. Настройка алгоритма примитива.
4. Получение результатов работы примитива.

В процессе создания задачи ей выдаётся уникальный идентификатор, по которому можно получить результат этой задачи.

3. Обзор решения

Для создания GraphQL-запросов была использована IDE *GRAPHiQL*².
Список созданных запросов:

- Запрос для получения типов данных в столбцах во входных данных.
- Запрос для получения результата работы алгоритма.

Для создания страницы с результатами выполнения использовался шаблон *Per-Page Layouts*³, в котором, для генерации страниц с одинаковым набором меню, но с различным наполнением, в “оболочку” вставляется наполнение страницы.

В разработке компонентов для создания формы настройки алгоритма использовалась библиотека *React-Select*⁴, которая позволяет создавать настраиваемые меню для выбора опций.

В разработке компонентов для страницы вывода результата работы алгоритма использовались уже существующие в проекте наработки.

Список созданных компонентов:

- Меню для выбора нескольких вариантов из предложенных (используется для выбора столбцов, для которых будет проверяться зависимость).
- Таблица, которая может выводить данные и позволяет взаимодействовать со своими ячейками.
- Функция, которая подготавливает данные с сервера к выводу в таблице.

²<https://github.com/graphql/graphiql> (дата доступа: 8 января 2023 г.).

³<https://nextjs.org/docs/basic-features/layouts#per-page-layouts> (дата доступа: 8 января 2023 г.).

⁴<https://react-select.com> (дата доступа: 8 января 2023 г.).

4. Практическая часть

4.1. Реализация

Во время разработки были, насколько это возможно, использованы лучшие практики. Все цели, которые были описаны в постановке задачи, были реализованы как отдельные компоненты, которые не зависят друг от друга, что позволяет использовать их и вне рамок текущих задач. Такой подход к решению задачи позволяет крайне легко масштабировать веб-приложение и даёт возможность легко добавлять новый функционал не изменяя уже существующий код.

4.2. Применение

Написанный интерфейс позволяет конечному пользователю воспользоваться алгоритмами для верификации метрических зависимостей. Это расширяет функционал веб-приложения и улучшает презентабельность проекта.

Заключение

В результате работы были достигнуты следующие цели:

- Созданы запросы для получения данных с сервера.
- Создана система хранения данных задачи.
- Созданы интерфейс настройки алгоритма и сопутствующие им компоненты.
- Созданы интерфейс отображения результатов работы и сопутствующие им компоненты.

И появились задачи на будущее:

- Написать документацию на сделанную работу.
- Переписать код интерфейса настройки алгоритма для упрощения последующего развития проекта.

Подробнее о сделанной работе можно ознакомиться по ссылке:

<https://github.com/vs9h/Desbordante/pull/59>

Список литературы

- [1] [Metric Functional Dependencies](#) / Nick Koudas, Avishek Saha, Divesh Srivastava, Suresh Venkatasubramanian // 2009 IEEE 25th International Conference on Data Engineering. — 2009. — P. 1275–1278.
- [2] Send Alik. Что же такое этот GraphQL? — URL: <https://habr.com/ru/post/326986> (дата обращения: 8 января 2023 г.).
- [3] Георгий Чернышев Михаил Полинцов Никита Бобров. Прimitives Desbordante: Функциональные зависимости и их применение в эксплорации и очистке данных. — URL: <https://www.habr.com/ru/company/unidata/blog/679390> (дата обращения: 8 января 2023 г.).
- [4] Георгий Чернышев Максим Струтовский Никита Бобров Антон Чижов Александр Смирнов Михаил Полинцов. Data profiling, и с чем его едят. — URL: <https://habr.com/ru/company/unidata/blog/667636> (дата обращения: 8 января 2023 г.).
- [5] Официальный сайт Angular. — URL: <https://angular.io> (дата обращения: 8 января 2023 г.).
- [6] Официальный сайт Apollo GraphQL. — URL: <https://www.apollographql.com> (дата обращения: 8 января 2023 г.).
- [7] Официальный сайт GraphQL. — URL: <https://www.graphql.org> (дата обращения: 8 января 2023 г.).
- [8] Официальный сайт NextJS. — URL: <https://nextjs.org> (дата обращения: 8 января 2023 г.).
- [9] Официальный сайт ReactJS. — URL: <https://www.graphql.org> (дата обращения: 8 января 2023 г.).
- [10] Официальный сайт VueJS. — URL: <https://vuejs.org> (дата обращения: 8 января 2023 г.).