

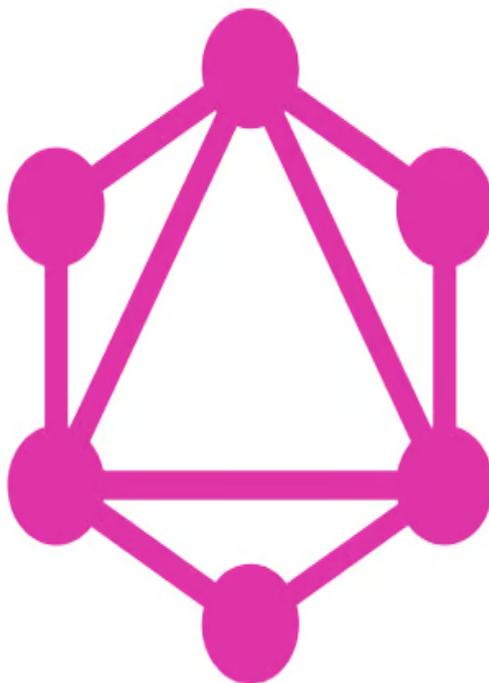


## 22 Лучших программы для работы с GraphQL, которые необходимо знать разработчику

### **Описание**

Можете ли вы представить себе мир, в котором разработчики имеют доступ к инструменту, соединяющему приложения с огромными типами данных, как масло на тост? Мир, в котором инструмент помогает разработчикам с абсолютной точностью извлекать то, что им нужно? Это не волшебный мир, это возможно уже сегодня благодаря GraphQL. Разработанный компанией Facebook в 2015 году, GraphQL с самого начала призван сделать процесс получения данных и манипулирования ими эффективным.

GraphQL достигает этого за счет декларативного и гибкого подхода к запросам к данным. В отличие от традиционных методов, таких как RESTful API, которые в итоге передают гораздо больше данных, чем нужно, этот метод позволяет определить структуру необходимых данных с помощью языка запросов. Используя этот язык запросов, можно запрашивать конкретные поля, вложенные отношения и многое другое.



Такой уровень специфичности и эффективности ранее был невозможен. Помимо получения данных, GraphQL также очень полезен для их модификации.

Использование традиционных RESTful API для модификации данных иногда приводит к проблемам совместимости. Но GraphQL предоставляет надежную систему типов, возможности интроспекции, а также самодокументирования, что позволяет эффективно создавать безверсионные API и обеспечивать плавную модернизацию.

Проще говоря, GraphQL – это язык выполнения и запросов, который позволяет пользователям эффективно получать данные из API и манипулировать ими, снижая при этом проблему недостаточной или избыточной выборки. Здесь клиент называет данные, которые ему действительно нужны, а API отвечает соответствующим образом. В этой статье мы расскажем о лучших программах для работы с GraphQL, которые изменят ваш рабочий процесс разработки в лучшую сторону. Они позволят вам создавать более эффективные приложения, чем когда-либо, и использовать ваши ресурсы в полной мере. Итак, мы начинаем наше путешествие по лучшим программам для работы с GraphQL.

## 22 Лучших программы для работы с GraphQL

### Apollo GraphQL

Apollo GraphQL – это набор программного обеспечения, включающий такие инструменты, как Apollo Server и Apollo Client. Все эти инструменты представляют собой единый инструментарий, упрощающий разработку на GraphQL. Эта богатая экосистема позволяет разработчикам создавать продукты быстрее и эффективнее.

### Apollo Server

Apollo Server – это сервер GraphQL, который можно легко настроить и запустить. Низкая конфигурация Apollo Server делает его очень простым в использовании как для профессионалов, так и для новичков. Apollo Server соответствует спецификации, что делает его совместимым с любым клиентским GraphQL-программным обеспечением.

Apollo Server поддерживает различные бэкенды, API и базы данных. Он поддерживает REST API и такие фреймворки, как Express, MongoDB, Firestore и многие другие. Кроме того, Apollo Server обладает такими возможностями, как сшивание схем, что является очень мощной функцией, поскольку позволяет создать только один API, предоставляющий доступ к нескольким back end'ам различных сервисов.

The image shows a live coding session on a green background. On the left, a vertical banner reads 'MISSION BRIEFING' with a large letter 'A' in a circle. Below it, a chat window shows messages from users like 'awp001', 'jumb011', and 'awp001'. The main area is a code editor displaying a JavaScript file named 'index.js'. The code defines an ApolloServer with a schema and a resolver. The schema includes a 'Query' type with a 'hello' field. The resolver for 'hello' returns 'Hello from Apollo!'. The server is configured with the schema, resolvers, and context. The terminal at the bottom shows the command 'serverless-gql-amplify on' and its output, indicating that the local backend resources are provisioned in the cloud. The status bar at the bottom of the code editor shows 'Ln 1, Col 62, Spaces: 4, UTF-8, LF, JavaScript, Deck | Connected | Active, Premier'.

## Serverless GraphQL

With AWS Amplify and Apollo Server

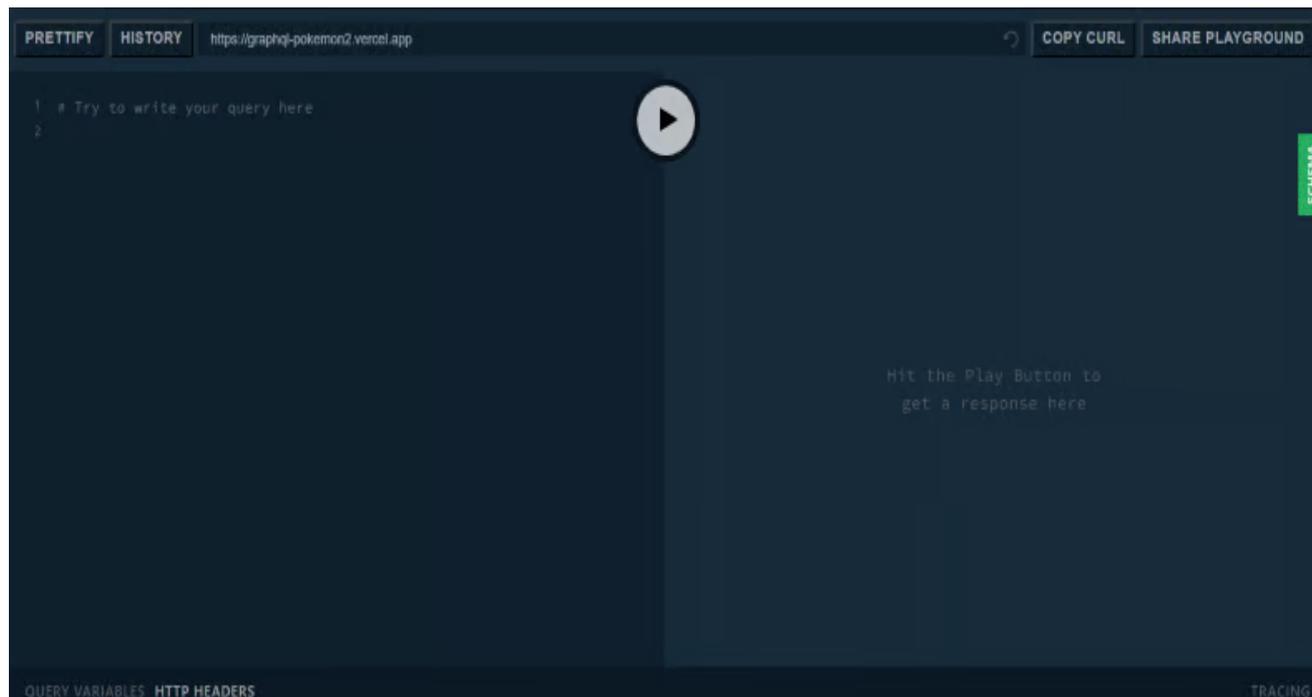
## Apollo Client

Apollo обеспечивает работу как на стороне клиента, так и на стороне сервера. Apollo Client предоставляет упрощенный способ управления состоянием и получения данных. Он имеет интуитивно понятный пользовательский интерфейс и простую в использовании компоновку. Apollo Client предоставляет такие возможности, как обновление пользовательского интерфейса, подписка на данные в реальном времени и многое другое. Подобные функции значительно повышают удобство разработки. Кроме того, они значительно упрощают процесс управления состояниями и получения данных. Все эти возможности делают Apollo Client полезным для многих разработчиков по всему миру.

# BUILD reactive variables

## GraphQL Playground

GraphQL Playground – это веб-инструмент, позволяющий составлять запросы и выполнять их. Этот инструмент очень функционален и имеет дружелюбный и интуитивно понятный интерфейс. С GraphQL Playground вы получаете доступ к мощной среде разработки, а также мобильность и удобство браузера. Несмотря на то, что GraphQL Playground работает в браузере, это не идет ни на какие компромиссы. Вы можете рассчитывать на те возможности, которые вы ожидаете от полноценной настольной IDE. Здесь есть подсветка синтаксиса, валидация кода и многие другие возможности. Помимо этих функций, вы также получаете доступ к предложениям в реальном времени и функциям автозаполнения.



GraphQL Playground также поддерживает несколько вкладок. Эта функция завоевала сердца многих разработчиков благодаря своему удобству. Это связано с тем, что вы получаете возможность выполнять и составлять несколько запросов одновременно. Такая возможность делает процесс работы очень эффективным и не требующим больших затрат времени. Кроме того, вы получаете доступ к функциям настройки, которые позволяют разработчикам персонализировать его по своему усмотрению. Проще говоря, GraphQL Playground – это GraphQL IDE в браузере без потери функциональности. Вы получаете все необходимые функции, такие как автозаполнение, подсветка синтаксиса и даже предложения в реальном времени.

## Prisma

Prisma – это инструмент для объектно-реляционного отображения (ORM). ORM – это фактически мост между объектно-ориентированными языками и реляционными базами данных. Prisma обеспечивает слой абстракции между базой данных и языком программирования и делает получение и обработку данных очень простой и удобной. При использовании Prisma вы получаете автоматически генерируемый API. Этот API безопасен для типов и сохраняет эту безопасность даже в JavaScript. Вам нужно только указать схему ваших данных, а все остальное Prisma сделает за вас.

Она использует собственный язык определения схем GraphQL, что делает работу очень простой. Эта простота является одной из ключевых особенностей данного

программного обеспечения. Этот инструмент можно использовать во многих языках и базах данных, поскольку он поддерживает все основные реляционные базы данных и различные популярные библиотеки и фреймворки. Кроме того, программа поддерживает миграцию данных, генерацию моделей и многое другое.

## Relay

Relay – это клиентская библиотека, созданная для React. С помощью Relay каждый компонент может объявить, какие данные ему нужны, а Relay позаботится обо всем остальном. Вы также можете повторно использовать компонент, при этом запросы также будут обновляться.

**Can Relay Work For Me?**

**Adopt Incrementally**

If you already can render React components, you're most of the way there. Relay requires a Babel plugin, and to also run the Relay Compiler.

You can use Relay out of the box with Create React App and Next.js.

**Used at Facebook Scale**

Relay is critical infrastructure in Facebook, there are tens of thousands of components using it. Relay was built in tandem with GraphQL and has full-time staff working to improve it.

**Make Complexity Explicit**

Relay requires a bit more up-front setup and tools, in favour of supporting an architecture of isolated components which can scale with your team and app complexity.

Learn these principles once, then spend more time working on business logic instead of pipelining data.

**Not Just for Big Apps**

If you're the sort of team that believes in using Flow or TypeScript to move error detection to dev-time, then Relay is likely a good fit for you.

It's probable you'd otherwise re-create a lot of Relay's caching, and UI best practices independently.

В Relay также предусмотрена возможность автоматического пакетирования. Она анализирует и объединяет похожие запросы, а также следит за тем, чтобы не были получены дубликаты данных. Это позволяет экономить время и пропускную способность сети и делает процесс гораздо быстрее и эффективнее. Иногда разработчикам требуется детальный контроль, и Relay предоставляет такую возможность. Relay оснащен очень полезным и мощным инструментарием.

В нем есть расширение dev tools, которое предоставляет такие возможности, как мониторинг сетевых запросов, проверка запросов, отладка и многое другое. Этот инструмент очень удобен, особенно когда речь идет об устранении неполадок. И

самое главное – Relay обеспечивает бесшовную экосистему. Он может интегрироваться с другими системами GraphQL и имеет поддержку различных популярных библиотек GraphQL, что делает его очень хорошим выбором.

## PostGraphile

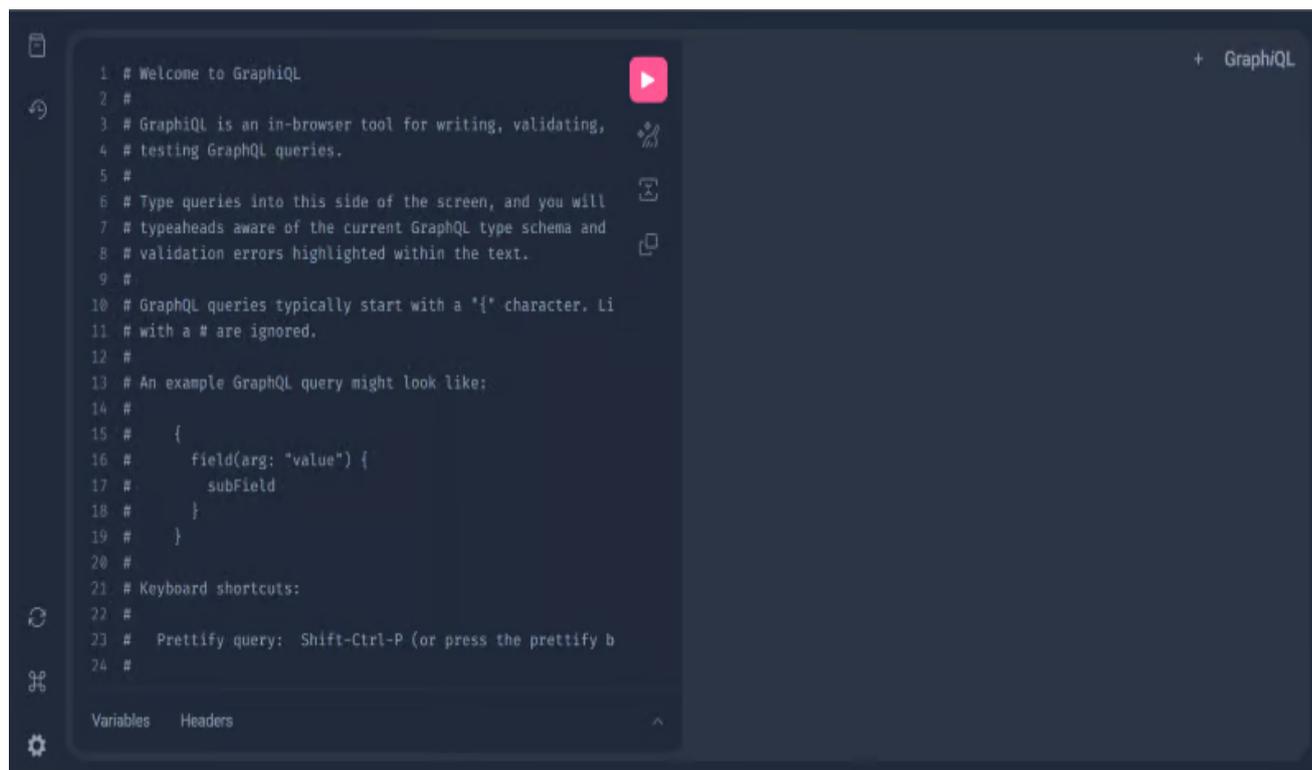
PostGraphile предоставляет возможность использовать GraphQL с базами данных PostgreSQL. Для этого необходимо открыть базу данных как конечную точку GraphQL. Таким образом, вы можете получать доступ к базе данных и взаимодействовать с ней с помощью GraphQL-запросов или мутаций.



Она обладает такими возможностями, как автоматическая генерация схемы. Это означает, что он самостоятельно анализирует схему базы данных и генерирует нужную схему. Она способна просматривать реляционные данные и анализировать сложные отношения, представления и процедуры. Затем он создает для вас API, который вы можете использовать для доступа к данным. Безопасность и данные – это синонимы. Поэтому мы должны помнить об аутентификации и авторизации. Но, к счастью, этот инструмент вас в этом вопросе поддержит. Он поддерживает популярные стандарты аутентификации. Он поддерживает OAuth, а также JWT (JSON Web Tokens), так что вам не придется беспокоиться о безопасности.

## GraphiQL

GraphiQL – это интегрированная среда разработки (IDE) в браузере. Вы получаете мощную IDE с удобствами, присущими браузеру. Она известна своей интерактивностью и интуитивностью и не разочаровывает, предоставляя чистый и простой пользовательский интерфейс.



GraphiQL обладает всеми возможностями, которые только можно пожелать от GraphQL IDE. Она поддерживает построение и выполнение запросов, подсветку синтаксиса, а также подсветку ошибок. Кроме того, в запросах можно использовать

переменные. Это отличная возможность, поскольку она позволяет легко передавать динамические значения. Данный инструмент оснащен встроенным проводником документации, позволяющим получить четкое представление о типах, операциях и полях схемы. Проводник документации очень полезен для понимания того, как работает API и чего он может достичь. Подобные возможности делают GraphQL востребованным. Поддержка документации, подсветка ошибок и помощь в построении запросов делают работу с GraphQL целостной и плавной и обеспечивают отличную ценность.

## Nexus

Nexus – это такой же инструмент для создания GraphQL API, как и другие. Но его отличием является подход, основанный на коде. В отличие от других инструментов, он не использует языки определения схем. Вы можете определить схему непосредственно в самом коде. Такой подход является более гибким и удобным для интеграции с существующими кодовыми базами.

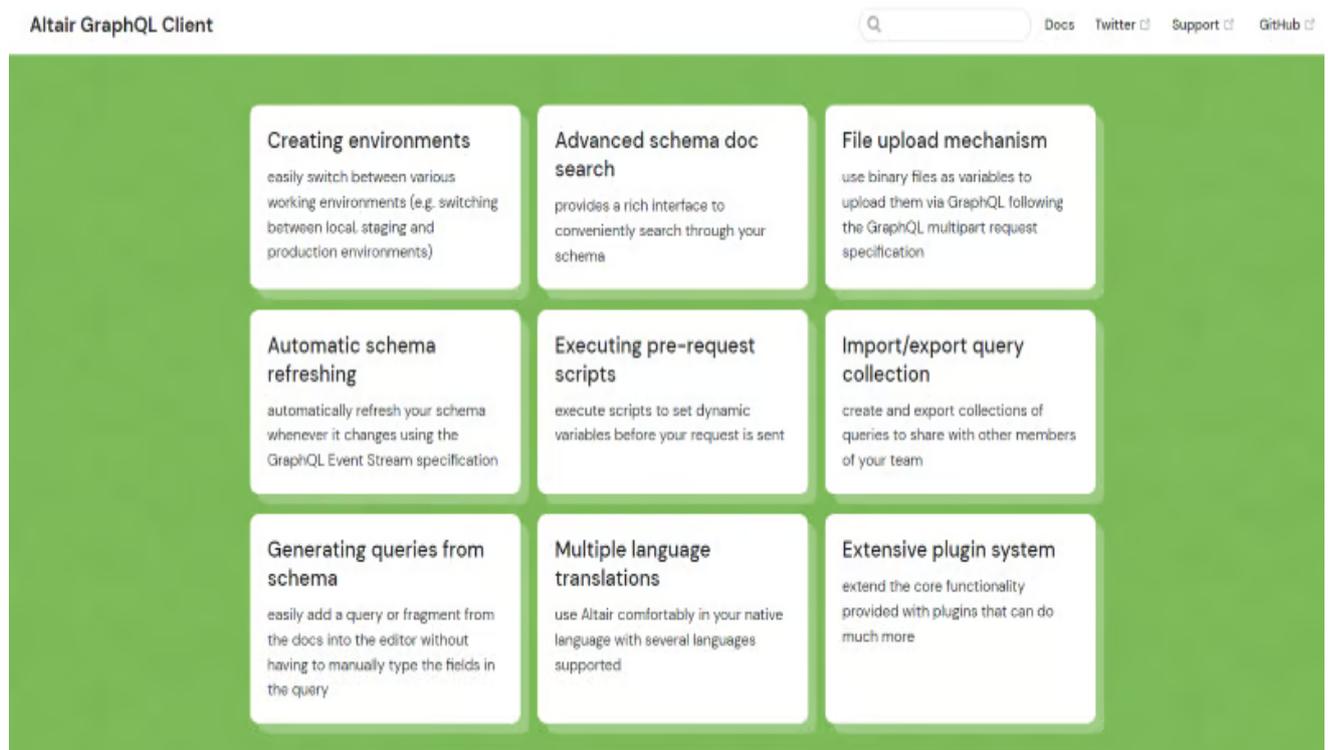


Nexus также безопасен для типов. Nexus использует Typescript, что делает его по своей сути безопасным для типов. Кроме того, вы получаете доступ к функциям автозавершения Typescript. Для определения схемы используется язык, специфичный для конкретной области. Поэтому синтаксис для определения схемы лаконичен, прост для понимания и читаем. Совместимость очень важна, когда речь идет о GraphQL. Но не стоит беспокоиться. Nexus поддерживает основные GraphQL-

серверы, такие как Apollo и Express. Вы можете легко интегрировать их с помощью поставляемых плагинов. Кроме того, Nexus очень модулен по своей конструкции. Nexus позволяет разработчикам разбивать схему и резольвер на различные небольшие компоненты, которые могут быть использованы повторно. Такой подход помогает поддерживать код и аккуратно его организовывать.

## Altair

Altair – это многоплатформенная клиентская IDE для GraphQL. Она доступна для всех основных операционных систем и браузеров и является отличным инструментом для работы с GraphQL. Altair поддерживает различные функции, такие как запросы, подписки и мутации.



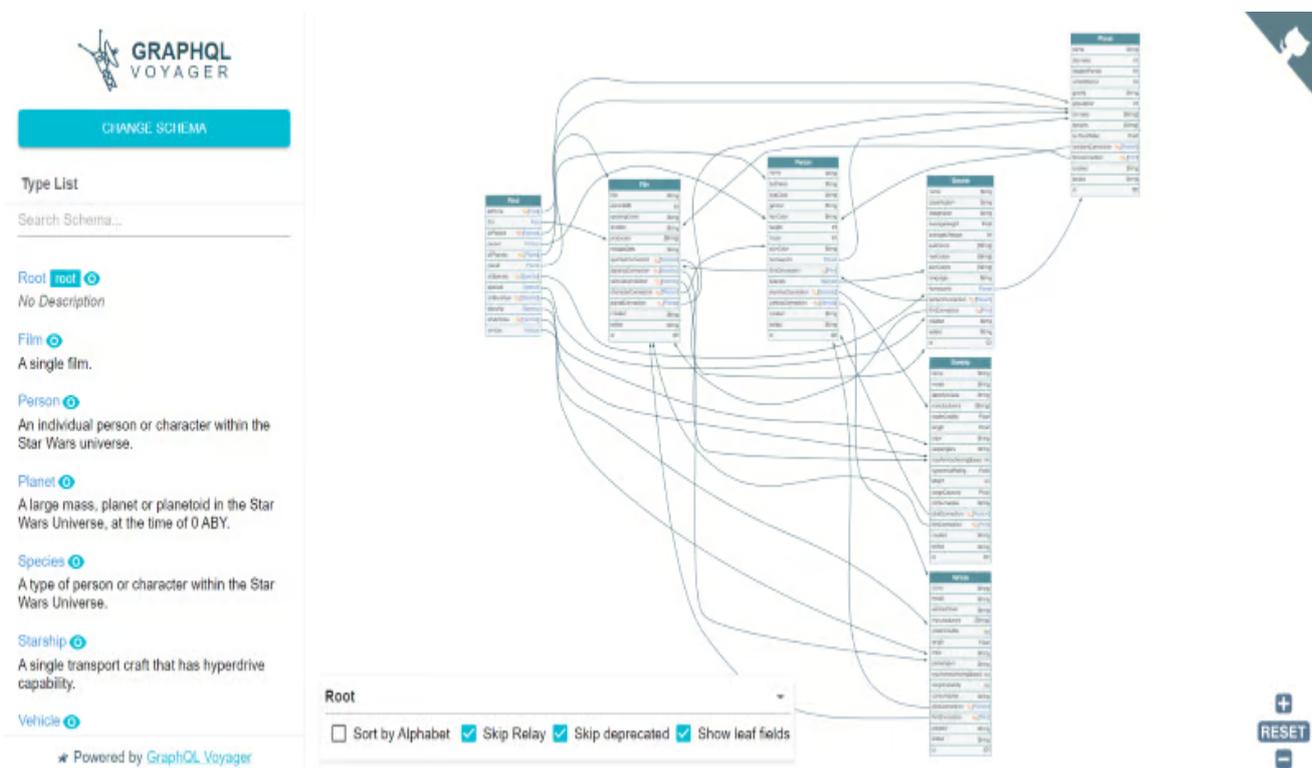
Altair интегрируется с рабочим процессом GraphQL и предоставляет удобный и визуально интерактивный пользовательский интерфейс для удовлетворения всех ваших потребностей в GraphQL. Кроме того, в Altair встроена поддержка нескольких конечных точек. Эта функция очень удобна и позволяет работать с несколькими конечными точками GraphQL одновременно. Кроме того, вы можете легко переключаться между несколькими конечными точками, что повышает производительность.

Сниппеты – это отличный способ легко повторно использовать определенный

запрос. В Altair имеется встроенная история запросов, которая позволяет легко сохранять и повторно использовать старые запросы. Кроме того, Altair обладает такими функциями, как поддержка переменных, мониторинг производительности, проверка ответов и многое другое, что делает Altair отличным дополнением к вашему рабочему процессу.

## GraphQL Voyager

GraphQL Voyager – это инструмент, необходимый для легкого понимания схем. Он анализирует схему GraphQL и помогает понять ее с помощью визуального представления типов и полей схемы. Это очень мощный и удобный инструмент как для опытных пользователей, так и для новичков. Начиная с интерфейса и заканчивая самими представлениями, они сделаны очень интуитивно понятными и легко воспринимаемыми.

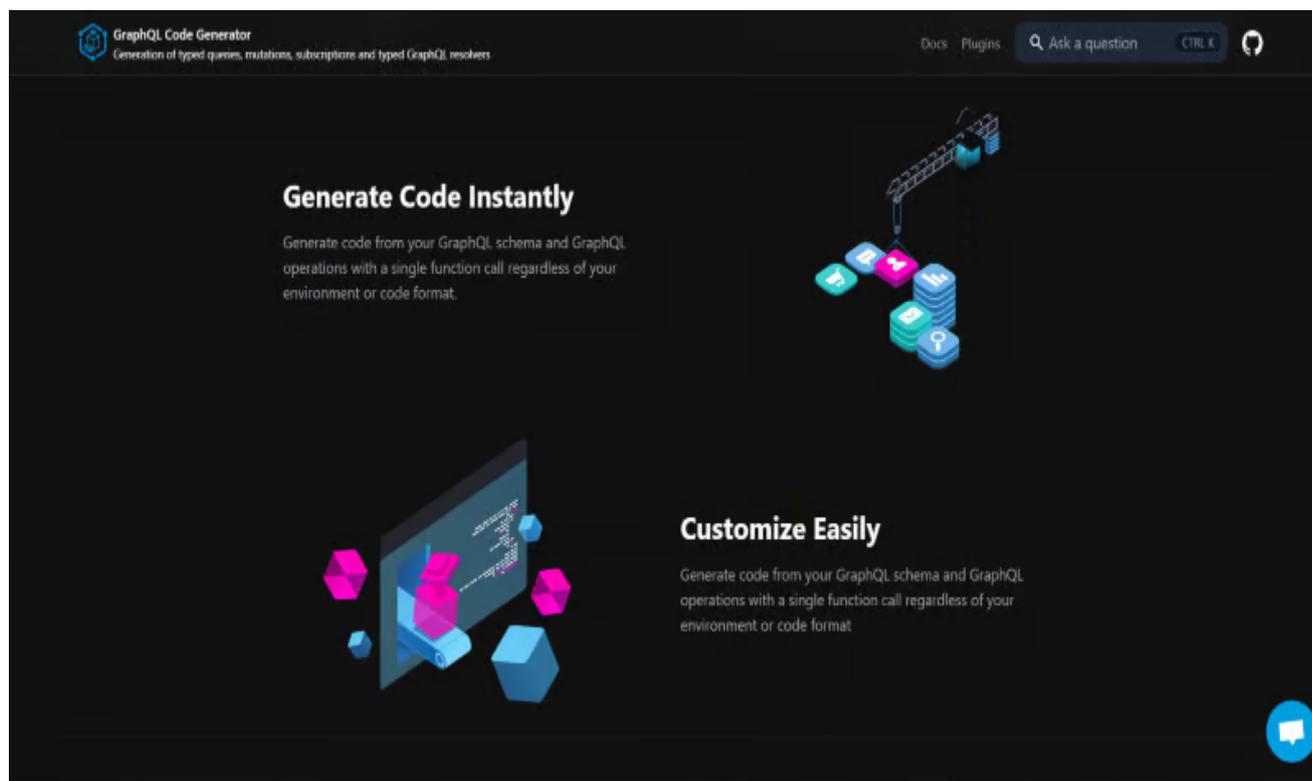


Вы также получаете возможность выбрать способ отображения данных. Данные могут быть представлены как в виде изображения, так и в формате JSON. Кроме того, GraphQL Voyager оснащен функцией поиска, которая позволяет легко найти конкретный фрагмент данных. Поиск может осуществляться по полям, ключевым словам и т.д. Voyager создан для того, чтобы быть простым. Благодаря функции авторазметки можно быть уверенным в том, что график будет читаемым и не будет

загроможденным. Кроме того, данные организованы в кластеры, что делает огромные данные очень удобными для понимания и чтения.

## GraphQL Code Generator

GraphQL Code Generator – это популярный инструмент для генерации кода на языке GraphQL. С помощью Code Generator можно сэкономить много времени и усилий, поскольку он предназначен для генерации кода как для фронт-энда, так и для бэк-энда. Сгенерированный код очень лаконичен и эффективен, а также обладает безопасностью типов.



Code Generator подходит для генерации кода для различных языков, таких как Ruby, TypeScript, JavaScript и Java. Генерируемый код также является настраиваемым, поскольку его можно легко изменить с помощью различных шаблонов. На этом список не заканчивается. Code Generator имеет богатую экосистему плагинов для различных фреймворков, что расширяет его функциональность и позволяет генерировать код для конкретного фреймворка. Кроме того, он не ограничивается схемами GraphQL. Поддерживаются также JSON, документы GraphQL и конечные точки. Этот инструмент экономит время разработчиков, избавляя их от написания повторяющегося кода, что позволяет им сосредоточиться на том, что они умеют делать лучше всего.

## GraphQL Editor

GraphQL Editor – очень мощный инструмент для разработчиков GraphQL. Это многоцелевой инструмент, поддерживающий такие функции, как редактирование, проектирование и даже совместная работа. Его визуальная привлекательность и простота использования делают его отличным выбором для разработчиков.

**Visual Editor**  
Work with a transparent GraphQL tree and visualize queries & mutations as forms. Try out our GraphQL Playground and take advantage of easy to use no-code solutions like click-out queries and intelligent frontend widgets. The full power of GraphQL is now available to everyone!  
[Learn more →](#)

**REST to GraphQL**  
Post your REST backends to GraphQL, create a proxy GraphQL service, deploy proxies and share them with others. Empower frontend developers with a well defined GraphQL backend.  
[Learn more →](#)

**Built-in E2E & Snapshot Testing**  
Utilize our fully functional GraphQL Playground. Collaborate on GraphQL queries live with your team, store them in the cloud and save queries as tests.  
[Learn more →](#)

**Self-deployed Cloud Backend**  
Create an instant fake backend from your GraphQL schema. Configure fake fields, mock responses instantly and allow both frontend and backend teams to work simultaneously. Share your fake backend with your team or deploy it publicly. With GraphQL Editor business people can finally define app logic.  
[Learn more →](#)

Благодаря функциям совместной работы большие и малые команды могут интегрировать его в свой рабочий процесс для обогащения опыта разработки и

повышения эффективности, поскольку разработчики могут легко делиться своими идеями и получать доступ к данным в одном и том же месте. Документация – лучший друг разработчика. GraphQL Editor понимает это и имеет специальный раздел для описания конкретного элемента. В дальнейшем это описание используется для создания подробной документации. Эта документация помогает другим разработчикам легко понять код. Валидация и проверка ошибок также встроены в GraphQL Editor. Она выделяет ошибки в схеме по мере ее написания и помогает писать безошибочный код.

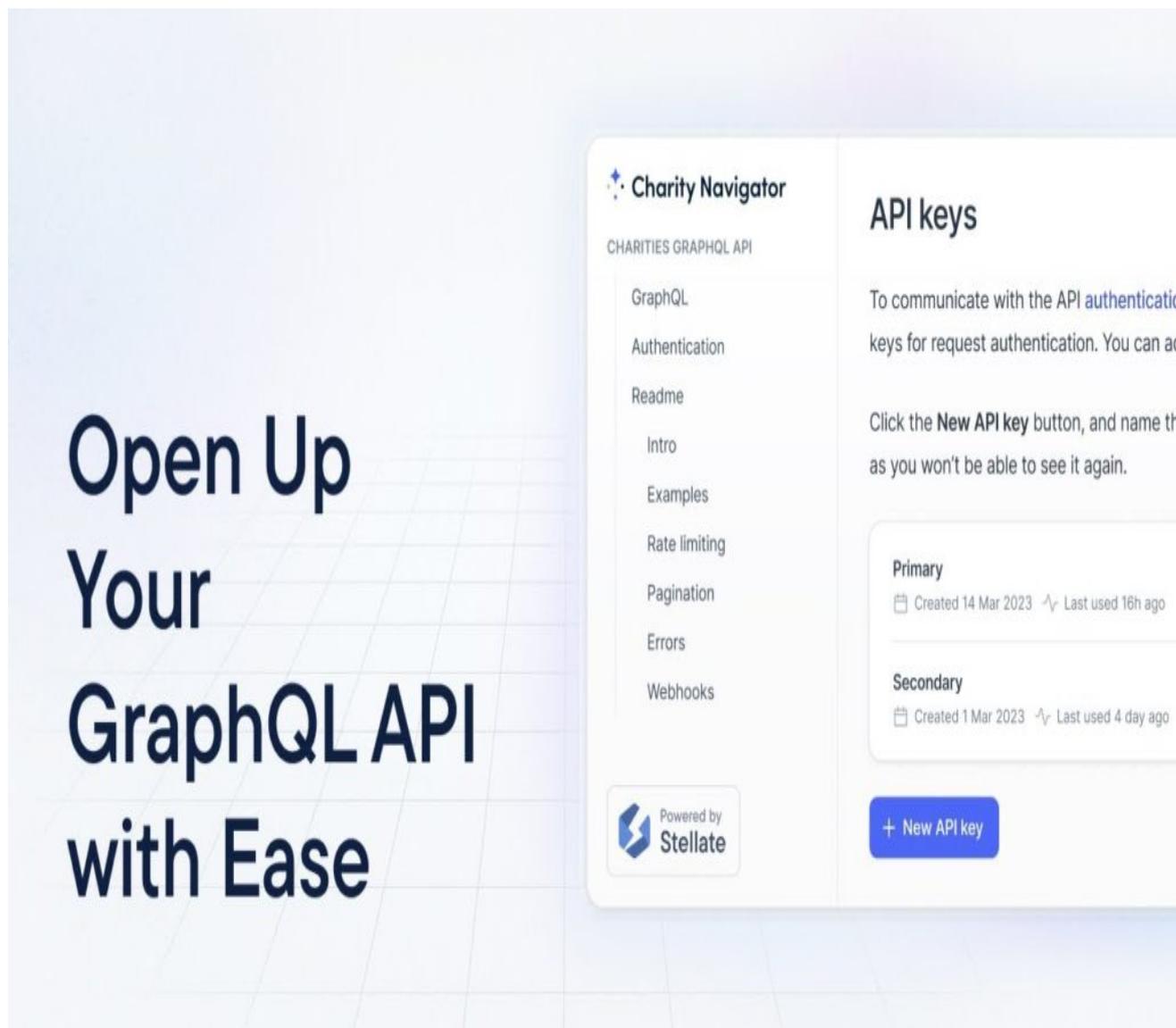
## Swagger2GraphQL

Swagger-to-GraphQL – это фантастический инструмент, который автоматически генерирует схемы GraphQL с помощью уже существующих спецификаций Swagger. Этот инструмент очень полезен, поскольку дает возможность использовать RESTful API, получая при этом преимущества GraphQL. Swagger-to-GraphQL принимает на вход спецификацию Swagger и предоставляет в машиночитаемом виде конечные точки API, полезную нагрузку и другие сведения. Затем он генерирует схему для GraphQL. Этот инструмент очень хорошо настраивается. Вы можете настроить его по своему вкусу, чтобы получить схему GraphQL и резольверы.

Кроме того, у вас есть возможность вручную редактировать схему. Инструмент очень гибкий. Наряду с этими возможностями, Swagger-to-GraphQL также совместим с различными инструментами в экосистеме GraphQL. Он поддерживает Apollo Server, GraphQL.js и многое другое. Подводя итог, можно сказать, что Swagger-to-GraphQL помогает соединить RESTful API и GraphQL и делает этот процесс достаточно простым. Если вы хотите интегрировать GraphQL в свой REST-рабочий процесс, это может стать отличным выбором.

## Stellate

Stellate позволяет ускорить вызовы API с помощью кэширования. Он представляет собой пограничный GraphQL-шлюз, окружающий ваш API. Stellate также полезен для ограничения скорости.



Кроме того, с помощью данных метрик можно увидеть и понять закономерности нагрузки и спланировать изменения схемы. Такие функции, как мгновенная очистка кэша, возможность экранирования API и непрерывное выполнение операций, делают этот инструмент обязательным для использования.

## AWS AppSync

AppSync от AWS – это сервис, позволяющий с легкостью разворачивать масштабируемые приложения, работающие в режиме реального времени. Такие тривиальные вещи, как синхронизация, совместная работа в реальном времени и поддержка автономной работы, AppSync делает очень простыми, поскольку они выполняются AppSync, в то время как вы сосредоточены на создании более

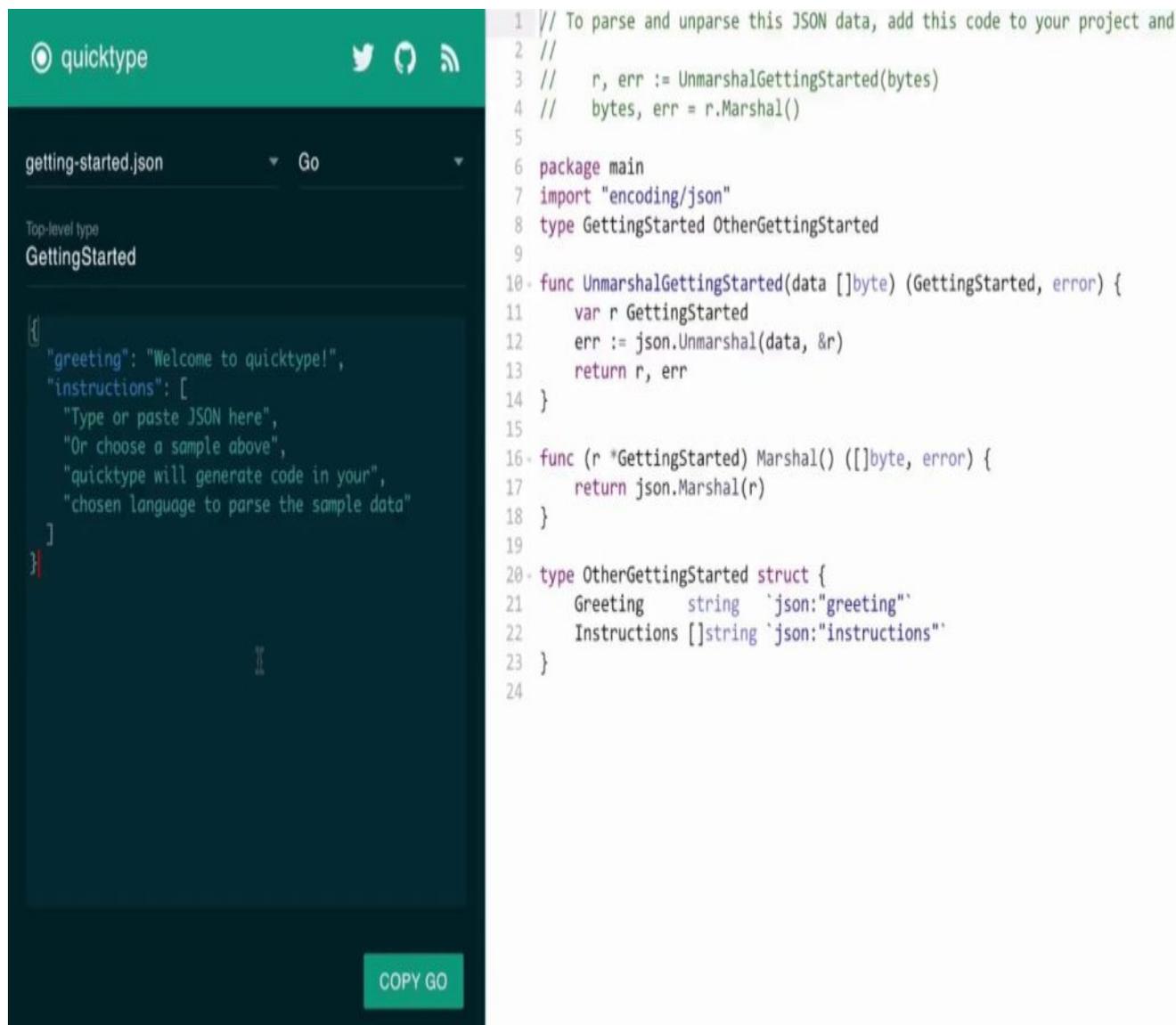
сложных приложений.

The screenshot shows the AWS AppSync console interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and user information. The main content area is titled 'MOBILE SERVICES' and features the 'AWS AppSync' logo and the tagline 'Build data-driven apps with real-time and offline capabilities'. A prominent orange button labeled 'Create API' is visible, with 'Estimated 3-5 mins' next to it. Below this, there's a 'Getting started' section with links to 'What is AWS AppSync?' (2 min read), 'Getting started with AWS AppSync' (6 min read), 'Designing GraphQL schemas' (14 min read), and 'GraphQL Overview' (10 min read). A 'More resources' section includes links to 'Documentation', 'FAQ', and 'AWS AppSync forum'. At the bottom, a 'How it works' diagram illustrates the workflow: 'Create and Upload Schema' (Developers use the console editor to define and deploy a GraphQL API), 'Connect Data Sources' (AWS AppSync automatically provisions data sources and connects them to the GraphQL API), and 'AppSync updates data in real time and manages data when offline' (Client applications make GraphQL API calls to fetch data, make changes, or subscribe to changes in real time from all users and devices).

Кроме того, вы получаете такие возможности, как объединенные API и кэширование, что обеспечивает пользователям большую гибкость. Кроме того, в систему включены такие функции, как наблюдаемость и корпоративная безопасность.

## Quicktype

Quicktype – это инструмент, позволяющий генерировать модели и сериализаторы из GraphQL и других форматов. Он может автоматически генерировать код для легкого чтения, записи и валидации. Это очень экономит время и повышает эффективность.



Quicktype может работать с различными языками, такими как Kotlin, Java, Dart, Typescript и др. Quicktype также может автоматически менять клиентские библиотеки. Кроме того, встроены такие функции, как автозаполнение и рефакторинг.

## Saleor

Saleor – это коммерческий API с открытым исходным кодом. Он помогает устранить презентационный слой из основной деятельности. Вы можете легко получить

---

доступ к API вашего экземпляра Saleor из браузера с помощью предоставляемой площадки GraphQL Playground.



# Introduction to Saleor

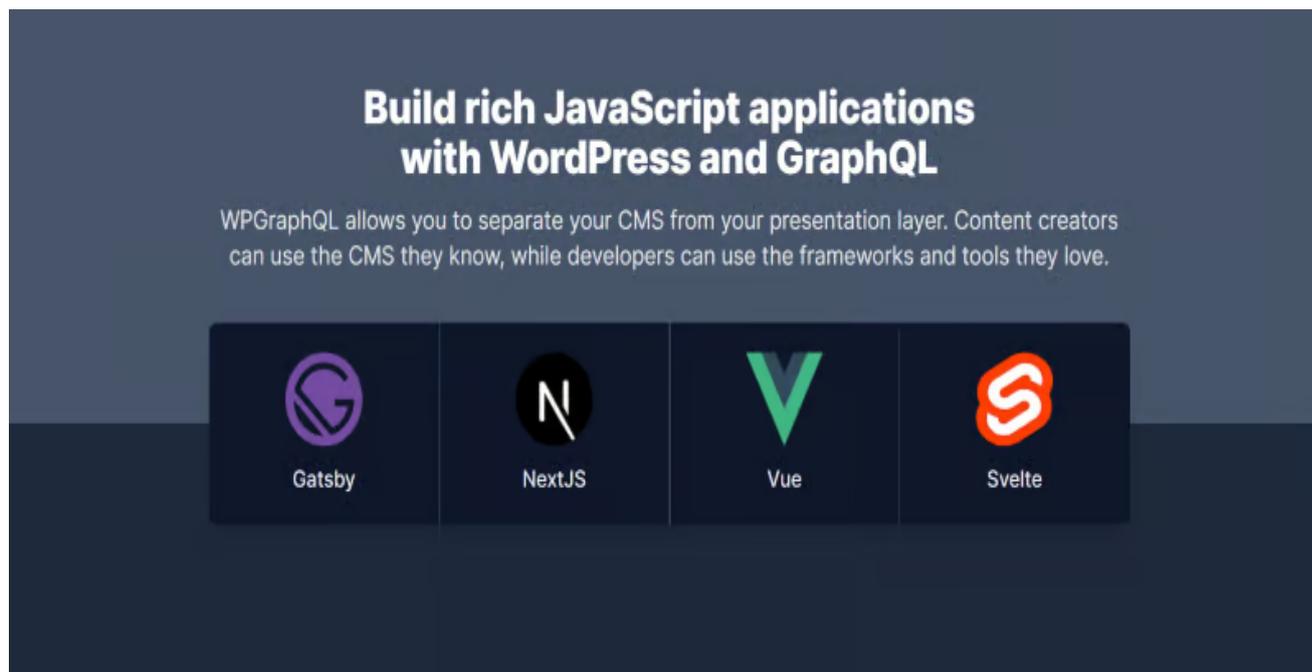
В комплект поставки Saleor также входит функция “Генерация кода GraphQL”. Это означает, что генерация запросов, типизация и другие аспекты могут быть легко автоматизированы.

## WPGraphQL

WPGraphQL, как следует из названия, представляет собой GraphQL API, созданный специально для WordPress. Он привносит все преимущества GraphQL в WordPress и обеспечивает эффективную выборку данных и возможность одновременной выборки нескольких ресурсов. Установка очень проста. Вы можете установить его

---

как любой другой плагин WordPress и легко активировать.



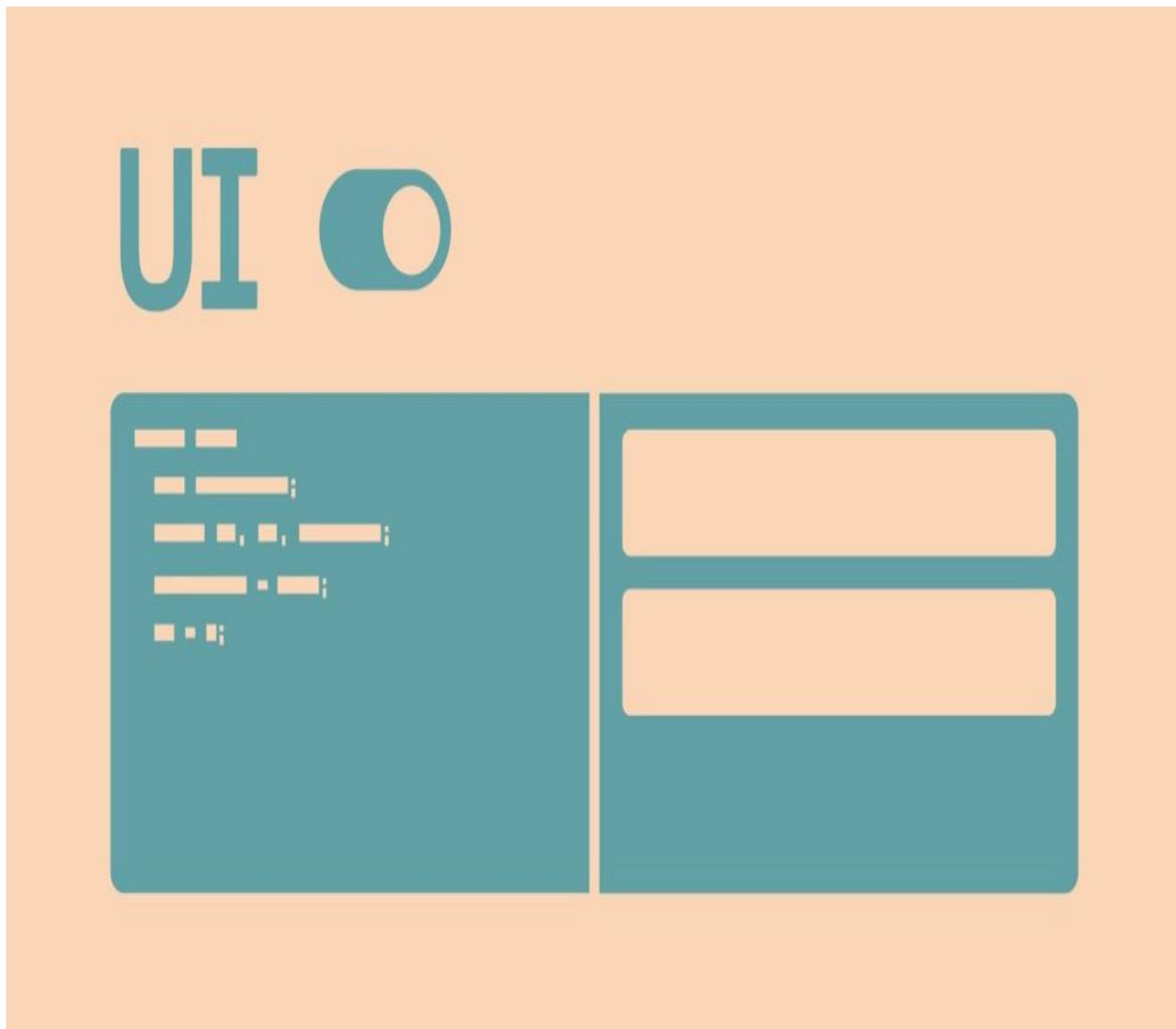
**Build rich JavaScript applications  
with WordPress and GraphQL**

WPGraphQL allows you to separate your CMS from your presentation layer. Content creators can use the CMS they know, while developers can use the frameworks and tools they love.

Gatsby NextJS Vue Svelte

## EdgeDB

EdgeDB – это база данных с поддержкой языков запросов, языков моделирования данных и др. EdgeDB поддерживает GraphQL с его расширениями. Она предоставляет “графоподобную схему с реляционным ядром”.



Кроме того, в нем реализована поддержка всех основных языков, таких как Python, Java, JavaScript, Rust и др.

## **Dgraph**

Dgraph позволяет создавать API и бэкенды графов, используя только схемы GraphQL. Dgraph имеет встроенную поддержку GraphQL, что означает отсутствие необходимости в трансляции GraphQL/SQL, что значительно упрощает процесс. Кроме того, Dgraph отличается высокой скоростью работы.



Она поддерживает одновременные запросы и такие функции, как оптимизация запросов на основе индексов, что придает ей быстродействие. Кроме того, существует единая схема приложения и базы данных, что означает отсутствие трансформации схемы и несоответствия импедансов.

## Hasura

Hasura – это замечательный инструмент для автоматической генерации API. С ее помощью достаточно указать на базу данных, и она сделает все остальное за вас, предоставив унифицированный, безопасный и работающий в реальном времени API.



Этот инструмент оснащен механизмом декларативной авторизации, который позволяет легко контролировать данные с помощью ролевых разрешений, а также таких вещей, как действия и удаленные схемы. Кроме того, он обеспечивает масштабируемую производительность и получение информации в реальном времени.

## URQL

URQL - это клиент GraphQL для различных JavaScript-фреймворков. К ним относятся Vue, Svelte, React и даже обычный JavaScript. URQL известен своей универсальностью и настраиваемостью. Кроме того, URQL отличается высокой производительностью, несмотря на свою легкость. URQL дает возможность

изменять способ доступа к данным, настраивая слой данных. Кроме того, он достаточно прост, поэтому не требует сложных конфигураций.

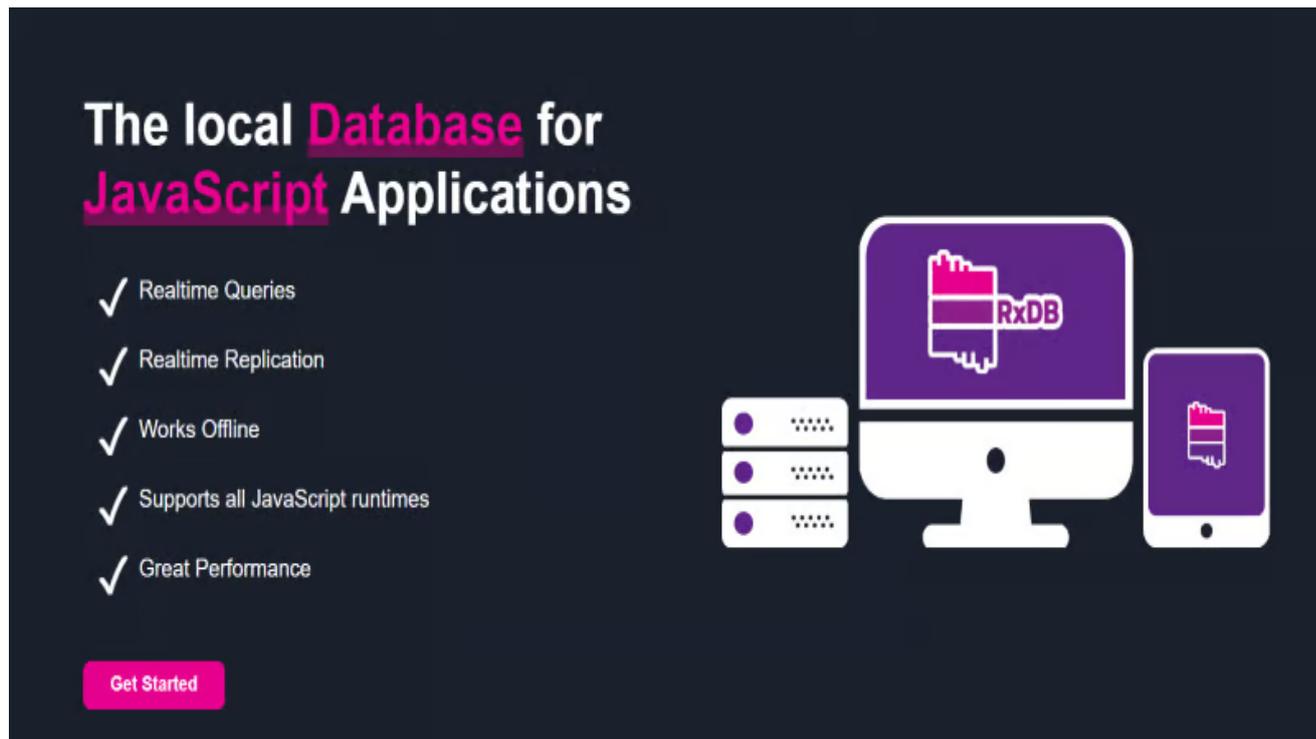
## Features



- Performant and functional**  
Lightweight, powerful, and easy to use; urql is a great alternative to bulky GraphQL clients.
- Extensible library that grows with you**  
Want to change how you fetch, cache, or subscribe to data? The urql exchanges allow you to customize your data layer to suit your needs.
- Logical default behavior and caching**  
Adding urql enables you to rapidly use GraphQL in your apps without complex configuration or large API overhead.

## RxDB

RxDB – это NoSQL-база данных, созданная специально для Javascript-приложений, причем в первую очередь для работы в автономном режиме. Это делает реализацию намного проще, быстрее и надежнее.



Он также оснащен обработчиками для GraphQL. Это означает, что вы можете выполнять процесс репликации с использованием GraphQL в качестве транспортного уровня. Это делает процесс простым и удобным.

## Заключение

GraphQL – это мощная технология. Она упрощает процесс разработки приложений. API, построенные с использованием GraphQL, более эффективны и очень гибкие. Такая эффективность и гибкость необходимы для того, чтобы соответствовать характеру современных приложений. Мы обсудили различные программные средства на языке GraphQL, которые обладают различными отличительными особенностями. Мы обсудили различные типы программного обеспечения. Среди них были интегрированные среды разработки, средства тестирования API, внутренние сервисы, клиентские сервисы, а также программные комплексы, обеспечивающие целостную работу.

Все это создано для различных целей и имеет свои сценарии использования. Как разработчику GraphQL, так и обычным разработчикам, знать о них крайне важно, поскольку этот инструмент помогает повысить эффективность и производительность. Теперь вам предстоит выбрать подходящий инструмент для конкретного рабочего процесса. В зависимости от степени вовлеченности GraphQL в

ваш рабочий процесс вы можете выбрать наиболее подходящий для вас инструмент. Вы также можете попробовать их в работе и определить, что вам больше подходит.

**Дата Создания**

14.07.2023