

# GRAPHQL – ЯЗЫК ЗАПРОСОВ И МАНИПУЛИРОВАНИЯ ДАННЫМИ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ

Татьяна АГАКИ, Руслан ПАНЕВСКИЙ

Технический Университет Молдовы, Департамент Программной Инженерии и Автоматики

**Аннотация:** Данная работа посвящена GraphQL. Она освещает историю этого популярного языка запросов, рассказывает о его преимуществах, недостатках и принципе работы. В статье будет рассмотрен пример, показывающий принцип работы GraphQL, а так же будет сделан вывод на основе рассмотренного материала.

**Ключевые слова:** GraphQL, язык запросов, данные, запрос, таблицы.

## Введение

GraphQL — это стандарт декларирования структуры данных и способов получения данных, который выступает дополнительным слоем между клиентом и сервером. GraphQL был разработан в большом старом Facebook, но даже гораздо более простые приложения могут столкнуться с ограничениями традиционных REST API интерфейсов.

Представьте, что необходимо отобразить список записей (posts), и под каждым опубликовать список лайков (likes), включая имена пользователей и аватары. На самом деле, это не сложно, вы просто измените API posts так, чтобы оно содержало массив likes, в котором будут объекты-пользователи. Но затем, при разработке мобильного приложения, оказалось, что из-за загрузки дополнительных данных приложение работает медленнее. Так что вам теперь нужно два endpoint, один возвращающий записи с лайками, а другой без них. Добавим ещё один фактор: оказывается, записи хранятся в базе данных MySQL, а лайки в Redis! Что же теперь делать?!

Facebook придумал концептуально простое решение: вместо того, чтобы иметь множество "глупых" endpoint, лучше иметь один "умный" endpoint, который будет способен работать со сложными запросами и придавать данным такую форму, какую запрашивает клиент.

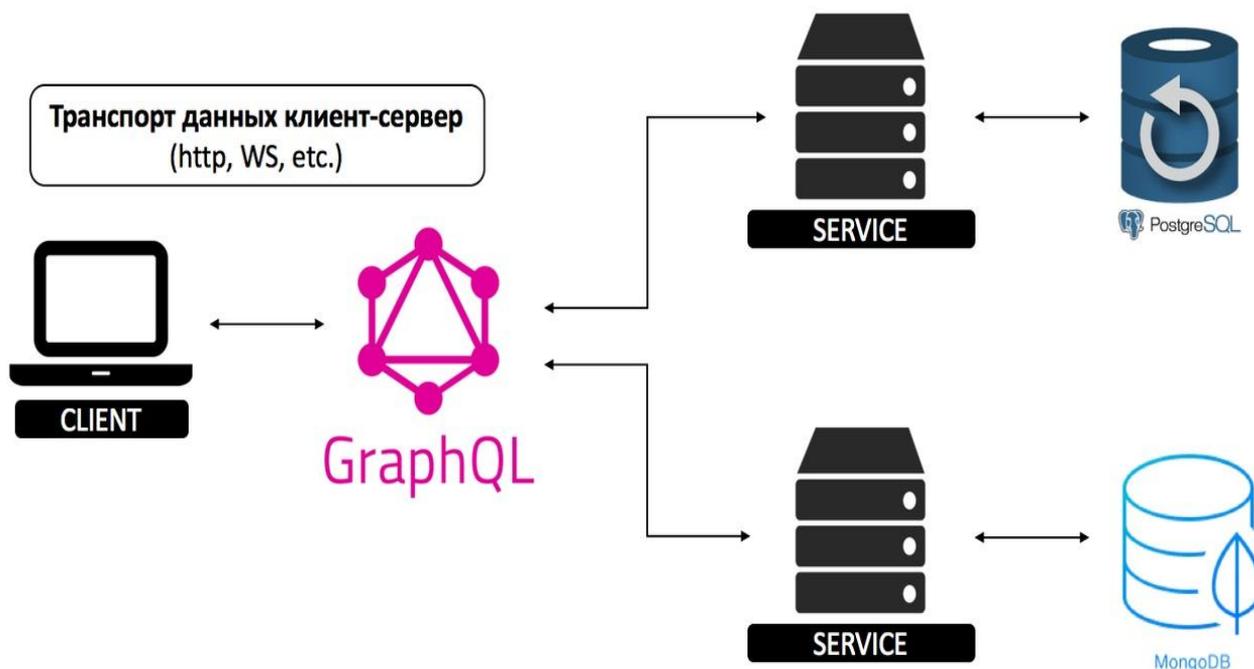


Рис. 1 Транспорт данных клиент-сервер.

Фактически, слой GraphQL находится между клиентом и одним или несколькими источниками данных; он принимает запросы клиентов и возвращает необходимые данные в соответствии с переданными инструкциями. Данный вид обработки данных показан на рисунке 1. GraphQL похож на личного помощника: вы можете передать ему адреса нескольких мест, а затем просто запрашивать то, что вам нужно и ждать их получения.

Одной из основных возможностей GraphQL является то, что структура и объем данных определяется клиентским приложением. GraphQL — это стандарт декларирования структуры данных и способов получения данных, который выступает дополнительным слоем между клиентом и сервером. GraphQL является действительно хорошей альтернативой REST API, делает front-end программирование очень легким, так как позволяет использовать единственный endpoint.

## 1. Наиболее важные проблемы, которые решает GraphQL

*Необходимость несколько раз обращаться за данными* для рендеринга компонента. GraphQL позволяет получить все необходимые данные за один запрос к серверу.

*Зависимость клиента от сервера.* С помощью GraphQL клиент общается на универсальном языке запросов, который отменяет необходимость для сервера жестко задавать структуру или состав возвращаемых данных и не привязывает клиента к конкретному серверу.

*Неэффективные способы разработки.* На GraphQL разработчики описывают необходимые для интерфейса данные с помощью декларативного языка. Разработчики сосредоточены на том, что хотят получить, а не как это сделать. Данные, необходимые для UI, тесно связаны с тем, как эти же данные описываются в GraphQL.

Facebook использовал GraphQL с 2012 г., еще до перехода этого языка в open source. Именно Facebook – та движущая сила, что отвечает за разработку спецификации GraphQL и ее справочной реализации на языке JavaScript. Итак, работая с GraphQL, вы уже стоите на плечах гигантов. Однако, и другие хорошо известные компании применяют этот язык в своих приложениях. Они инвестируют в экосистему GraphQL, поскольку современные приложения колоссально нуждаются именно в таком языке.

## Заключение

В заключении перечислим некоторые возможности GraphQL, которые значительно облегчают работу с данными и их обработку:

- Несколько наборов данных в одном запросе;
- Фильтрация;
- Ограничение доступа;
- Авто генерируемая схема;
- Запросы (базовые, вложенные объекты, список из простых типов);
- Полнофункциональный парсер.

Если база данных представляла бы собой публичную библиотеку, то GraphQL стал бы библиотекарем. Он слушает Ваши запросы и определяет, что Вам нужно. Затем он идет вдоль книжных полок и находит данные, которые Вы просили. Книжные полки в этой аналогии являются таблицами базы данных.

## Библиография

1. A query language for your API. - [Электронный ресурс]. – Режим доступа: <https://graphql.org>
2. GraphQL API v4. - [Электронный ресурс]. – Режим доступа: <https://developer.github.com/v4/>
3. Что же такое этот GraphQL? - [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/326986/> - 2017
4. Алекс Банкс и Ив. Порселло. - Learning GraphQL: Declarative Data Fetching for Modern Web Apps – 2018.

# AMAZON WEB SERVICES COGNITO

Анна ЗИНГАН, Максим КУКЛЕВ

Технический Университет Молдовы, Департамент Программной Инженерии и Автоматики

**Аннотация:** В данной работе проводится описание возможностей и принципа работы базы данных Amazon Cognito.

**Ключевые слова:** база данных, веб-сервисы Amazon, User Pools, Identity Pools.

## 1. Назначение, компоненты и возможности Amazon Cognito

Amazon Cognito – это веб-сервис, предоставляемый компанией Amazon. Это простая в использовании, и вместе с тем, надёжная база данных, предназначенная для авторизации, регистрации и контроля доступа пользователей к веб-ресурсу или мобильному приложению.

Двумя основными компонентами Amazon Cognito являются сервисы User Pools и Identity Pools.

User Pools позволяют создавать профили и токены аутентификации для всех типов пользователей: как для тех, кто регистрируется в приложении напрямую, так и для тех, кто авторизуется через социальные или корпоративные сети. Данный сервис обладает способностью к масштабированию до десятков и даже сотен миллионов пользователей, а для его настройки нет необходимости прибегать к созданию специальной серверной инфраструктуры.

Сервис Identity Pools позволяет выдавать пользователям доступ к другим веб-сервисам Amazon. Он сохраняет информацию о том, какие платформы, девайсы и операционные системы используются для входа тем или иным пользователем. Данная информация синхронизируется с веб-сервисами Amazon каждый раз, когда пользователь находится в сети. Информация, предоставляемая сервисом Identity Pools необходима для определения и назначения ролей для пользователей.

Сервисы User Pools и Identity Pools можно использовать и вместе, и по отдельности. В качестве примера совместного использования можно рассмотреть типичный рабочий сценарий, целью является аутентификация юзера и выдача доступа к другому сервису AWS.

На первом шаге пользователь авторизуется через user pool и получает соответствующие токены после успешной аутентификации. Затем ваше приложение обменивает токены на credenциалы AWS с использованием identity pool. Теперь пользователь может воспользоваться этими credenциалами для получения доступа к другим сервисам AWS, к примеру, Amazon S3 и DynamoDB (рисунок 1).

Amazon Cognito позволяет своим пользователям пользоваться для авторизации такими распространёнными поставщиками социальных удостоверений, как Google, Facebook и Amazon, и поставщиками корпоративных удостоверений на основе SAML.

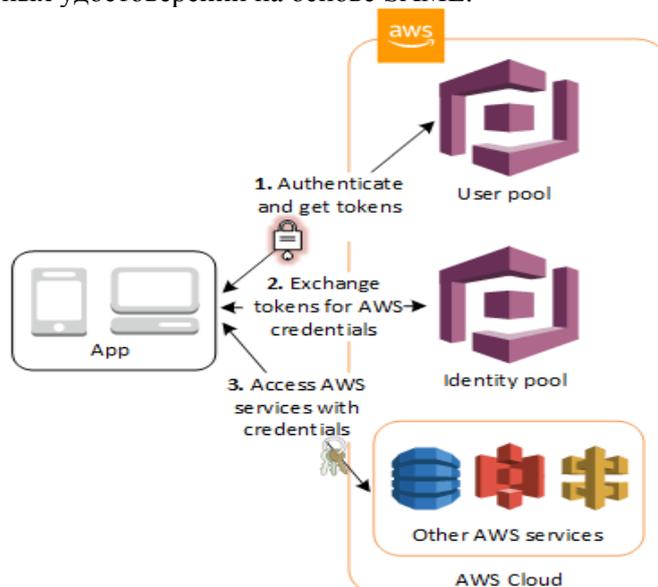


Рисунок 1 – Рабочий сценарий Amazon Cognito

Среди возможностей Amazon Cognito есть многофакторная аутентификация и шифрование данных во время их передачи и хранения. Обеспечиваемый ею уровень конфиденциальности подходит даже для организаций с самыми высокими требованиями к безопасности. Amazon Cognito соответствует требованиям HIPAA, PCI DSS, SOC, а также стандартам ISO/IEC 27001, ISO/EIC 27017, ISO/EIC 27018 и ISO 9001.

Дополняют список возможностей Amazon Cognito адаптивная аутентификация и защита от использования скомпрометированных учетных данных. Адаптивная аутентификация – это механизм, основанный на анализе риска и обеспечивающий безопасность как пользователей приложения, так и самого приложения в целом. Такие подозрительные действия, как попытка входа с не использовавшегося ранее устройства или из неизвестной локации, могут как принудить пользователя пройти дополнительную проверку, так и вовсе отказать ему в доступе.

### **Заключение**

Сохранение и синхронизация пользовательских данных, а также встроенный настраиваемый интерфейс для регистрации и авторизации пользователей делают Amazon Cognito достойным подспорьем любому разработчику мобильных и веб-приложений, позволяя не беспокоиться об организации и поддержке back-end архитектуры приложения.

### **Библиография**

1. What is Amazon Cognito? [Электронный ресурс]. – Режим доступа: <https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html>
2. Amazon Cognito – простая и безопасная регистрация и авторизация пользователей. - [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/cognito/>
3. Amazon Cognito – вопросы и ответы. - [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/cognito/faqs/>
4. AWS на понятном русском – шпаргалка по веб-сервисам Amazon. - [Электронный ресурс]. – Режим доступа: <https://tproger.ru/translations/aws-in-plain-russian/>
5. AWS представляет мобильные средства нового поколения [Электронный ресурс]. – Режим доступа: [http://cloudzone.ru/community/blogs/amazon\\_com/358.html](http://cloudzone.ru/community/blogs/amazon_com/358.html)