

# GraphQL

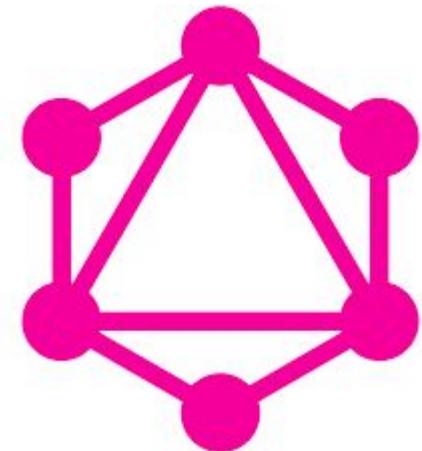
Мартынов Александр  
Сороколетов Денис

# GraphQL

---

GraphQL — это язык запросов к API и среда выполнения, которая позволяет клиенту запрашивать ровно те данные, которые ему нужны, не больше и не меньше.

- один эндпоинт
- гибкая структура ответа
- статическая типизированная схема
  
- не база данных
- не сервер
- не фреймворк



# Почему Graph?

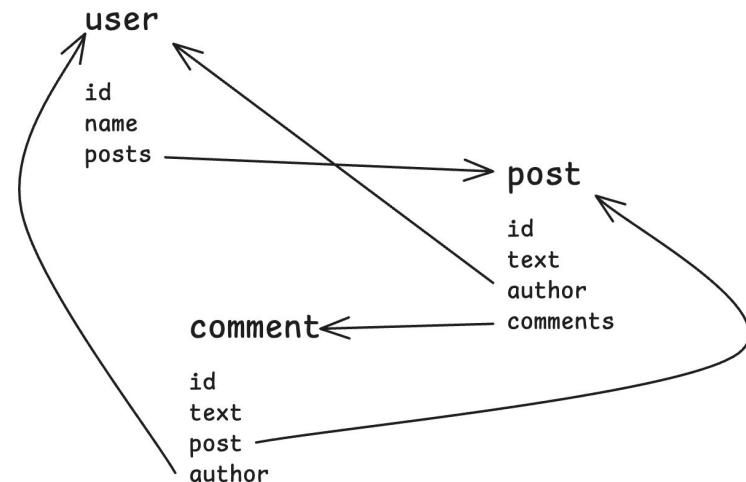
---

Данные в приложении — это не только таблицы, а сеть взаимосвязанных сущностей

GraphQL моделирует данные как **граф**: узлы (типы) и рёбра (связи)

Запрос — это навигация по этому графу: клиент выбирает путь и поля

Ответ повторяет структуру запрошенного подграфа



# GraphQL как альтернатива REST

---

REST — это архитектурный стиль, в котором, как правило, каждая сущность — это ресурс.

У каждого ресурса есть свой URL и свой формат ответа. Клиент получает данные строго в том виде, который решил сервер.

GET /users/1

GET /users/1/posts

GET /posts/10/comments

## GraphQL и REST

- API поверх HTTP
- JSON в ответах
- Передают клиенту данные
- Разделяют сервер и клиент

# Какую проблему REST решает GraphQL

---

REST-проблемы:

- Overfetching
- Underfetching
- Много запросов (N+1)
- Разрастание эндпоинтов
- Трудно обслуживать разные клиенты
- Версионирование API

```
/users/1  
/users/1/posts  
/posts/10/comments
```

GraphQL решает:

- Один запрос = вся нужная структура
- Клиент управляет полями
- Одна схема вместо множества версий

```
query { user(id:1){ posts { comments } } }
```

## Типы запросов

---

Query - запросы в GraphQL  
(аналог GET в REST), с помощь.  
них можно получать данные с  
сервера.

Запросы - строки, отправляемые  
в теле HTTP POST-запроса

Важно, что все типы запросов в  
GraphQL отправляются через  
POST (если мы говорим про  
обмен по HTTP)

Mutation - позволяют  
добавлять данные в базу  
(Аналог POST и PUT в REST)

Subscription - позволяют  
клиенту слушать изменения в  
БД в реальном времени

# Демонстрация

# Где применяется GraphQL

---

GraphQL используется:

- Web SPA (React/Vue/Svelte)
- Мобильные приложения
- Backend for Frontend (BFF)
- Микросервисы
- Часто меняющиеся API

GraphQL используют такие крупные компании, как:

- Facebook
- GitHub
- Shopify
- Netflix
- Airbnb

# Плюсы и минусы GraphQL

---

## Плюсы

- Клиент выбирает структуру данных
- Один запрос вместо нескольких
- Нет версий API
- Типизированная схема как контракт
- Богатая экосистема (Apollo, Relay)

## Минусы

- Сложность сервера
- Нужно контролировать сложность запросов
- Кеширование на уровне HTTP непросто
- Возможны проблемы с производительностью без DataLoader
- Не всегда нужен для простого CRUD