

# GraphQL

- **GraphQL** — язык запросов с открытым исходным кодом, разработанный **Facebook**.
- Создавался как более эффективная альтернатива **REST** для разработки и использования программных интерфейсов приложений.
- Позволяет сформировать запрос, ответ на который не будет содержать ненужных данных.
- Нет необходимости создавать дополнительные API, можно обойтись одним endpoint'ом.
- **GraphQL** — сильно типизированный язык, что позволяет предварительно оценить корректность запроса в рамках системы типов этого синтаксиса, до исполнения.
- Описание: <https://graphql.org/>

# Ответы обычных REST сервисов

```
[
  {
    title: 'My First Post',
    likes: [
      {
        name: 'Sacha',
        avatar: 'sacha.jpg'
      },
      {
        name: 'Mike',
        avatar: 'mike.jpg'
      }
    ]
  },
  {
    title: 'Another Great Post',
    likes: [
      {
        name: 'Julia',
        avatar: 'julia.jpg'
      },
      {
        name: 'Sacha',
        avatar: 'sacha.jpg'
      }
    ]
  }
]
```

## My First Post

Liked by:



Sacha



Mike

## Another Great Post

Liked by:



Julia



Sacha

# GraphQL запрос

## ➤ Запрос:

```
➤ query {  
➤   users {  
➤     fname  
➤     age  
➤   }  
➤ }
```

## ➤ Ответ (JSON):

```
➤ data : {  
➤   users [  
➤     {  
➤       "fname": "Joe",  
➤       "age": 23  
➤     },  
➤     {  
➤       "fname": "Betty",  
➤       "age": 29  
➤     }  
➤   ]  
➤ }
```

# GraphQL запрос с параметром

```
> query {  
>   post(id: "123foo"){  
>     title  
>     body  
>     author{  
>       name  
>       avatarUrl  
>       profileUrl  
>     }  
>   }  
> }
```

# Именованный запрос с параметром

```
➤ query getMyPost($id: String) {  
➤   post(id: $id){  
➤     title  
➤     body  
➤     author{  
➤       name  
➤       avatarUrl  
➤       profileUrl  
➤     }  
➤   }  
➤ }
```

# Три вида запросов (корневых типов GraphQL)

- **Query** - получение данных в нужном формате (аналог GET в REST).
- **Mutation** - изменение данных на сервере(аналог POST и PUT в REST).
- **Subscription** — позволяет слушать изменения в базе данных в режиме реального времени (используют вебсокеты).
- Запросы — строки, которые отправляются в теле HTTP POST.

# Запрос на изменение данных

```
➤ mutation CreateReviewForEpisode($ep: Episode, $review:  
  ReviewInput) {  
➤   createReview(episode: $ep, review: $review) {  
➤     stars  
➤     commentary  
➤   }  
➤ }
```

# Реализации GraphQL на PHP

Существуют различные библиотеки PHP реализующие в той или иной степени GraphQL

<http://webonyx.github.io/graphql-php/>

<https://github.com/webonyx/graphql-php>

<https://github.com/youshido-php/GraphQL>

Пример доступа к данным через GraphQL

<https://docs.github.com/en/graphql/overview/explorer>

Установка через **composer** (<https://getcomposer.org>):

**composer require webonyx/graphql-php**

Используя функции библиотеки реализовать схему данных, ресолвер (подтягивающий данные)