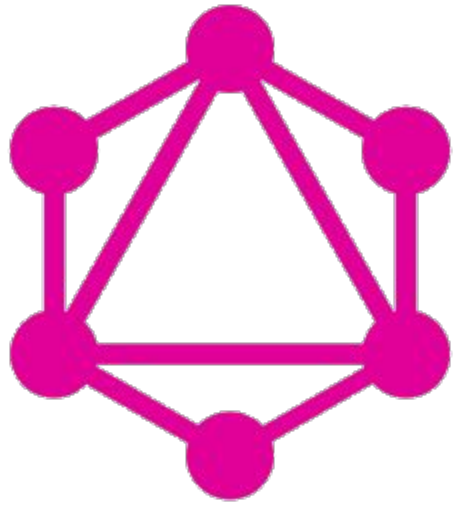


# REST in peace?

Попрактикуемся с GraphQL

Олег

 @Degtey\_o  
ДЕГТЕВ



GraphQL

## Чем GraphQL не является?

- Не новая либа
- Не новый фреймворк
- Ничего общего с графовыми базами данных

REST?

GET /api/v1/users

Demo.json • Settings



```
[  
  {  
    "id": 1,  
    "login": "Admin",  
    "password": "Qwerty1"  
  },  
  {  
    "id": 2,  
    "login": "SecuredUser",  
    "password": "123456"  
  }  
]
```



Demo.json • Settings

```
[  
  {  
    "id": 1,  
    "login": "Admin",  
    "password": "Qwerty1",  
    "groups": [1,2]  
  },  
  {  
    "id": 2,  
    "login": "SecuredUser",  
    "password": "123456",  
    "groups": [2]  
  }  
]
```

REST?

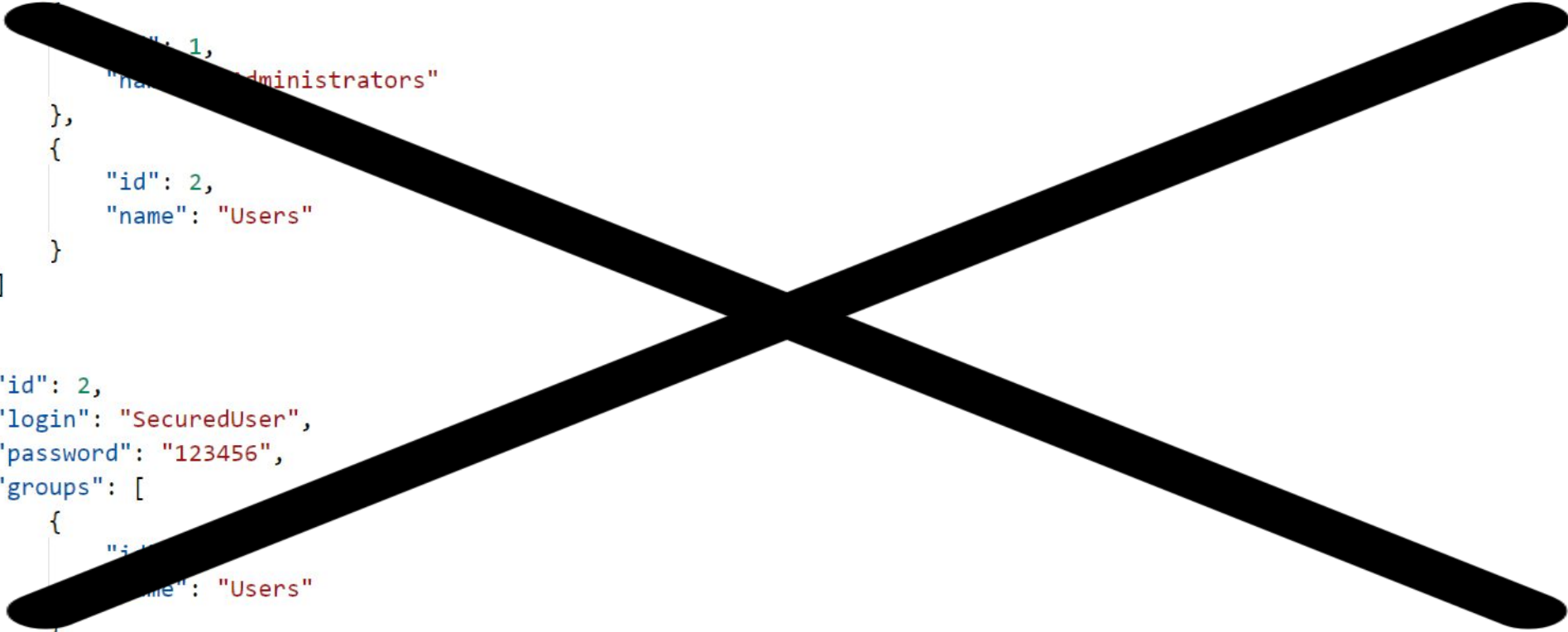
GET /api/v1/groups/1

REST?

GET `/api/v1/users/1/groups`



```
{  
  "id": 1,  
  "login": "Admin",  
  "password": "Qwerty1",  
  "groups": [  
    {  
      "id": 1,  
      "name": "Administrators"  
    },  
    {  
      "id": 2,  
      "name": "Users"  
    }  
  ]  
},  
{  
  "id": 2,  
  "login": "SecuredUser",  
  "password": "123456",  
  "groups": [  
    {  
      "id": 1,  
      "name": "Users"  
    }  
  ]  
}
```



REST?

N+1 Problem

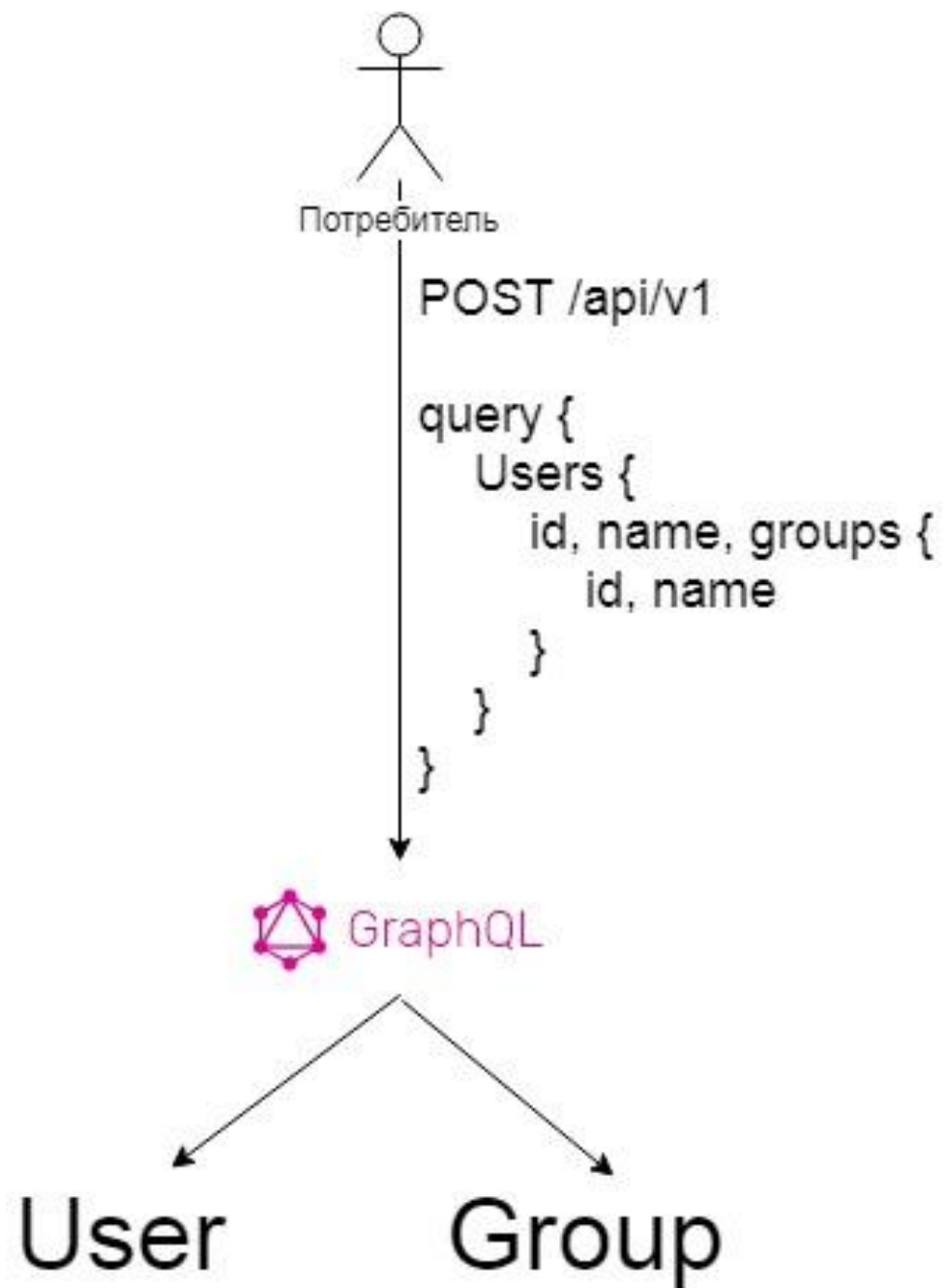
# REST?

Что использовать?

- PUT / PATCH?
- 400 Bad Request / 422 Unprocessable Entity?

## REST?

- 201 Created
- Updated?
- Deleted?





```
1 query {  
2   Users {  
3     id  
4   }  
5 }  
6
```

```
{  
  "data": {  
    "Users": [  
      {  
        "id": 1  
      },  
      {  
        "id": 2  
      },  
      {  
        "id": 3  
      },  
      {  
        "id": 4  
      },  
      {  
        "id": 5  
      }  
    ]  
  }  
}
```



```
1 query {  
2   Users {  
3     id,  
4     name  
5   }  
6 }  
7
```

```
{  
  "data": {  
    "Users": [  
      {  
        "id": 1,  
        "name": "Martin Odersky"  
      },  
      {  
        "id": 2,  
        "name": "James Gosling"  
      },  
      {  
        "id": 3,  
        "name": "Bjarne Stroustrup"  
      },  
      {  
        "id": 4,  
        "name": "Brendan Eich"  
      },  
      {  
        "id": 5,  
        "name": "Linus Benedict Torvalds"  
      }  
    ]  
  }  
}
```



```
1 query {  
2   Users {  
3     id,  
4     name,  
5     groups {  
6       id  
7     }  
8   }  
9 }  
10
```

```
{  
  "data": {  
    "Users": [  
      {  
        "id": 1,  
        "name": "Martin Odersky",  
        "groups": [  
          {  
            "id": 1  
          },  
          {  
            "id": 2  
          }  
        ]  
      },  
      {  
        "id": 2,  
        "name": "James Gosling",  
        "groups": [  
          {  
            "id": 1  
          },  
          {  
            "id": 2  
          }  
        ]  
      },  
      {  
        "id": 3,  
        "name": "Frank..."  
      }  
    ]  
  }  
}
```





```
1 query {  
2   Users {  
3     id,  
4     name,  
5     groups {  
6       id,  
7       name  
8     }  
9   }  
10 }  
11
```

```
{  
  "data": {  
    "Users": [  
      {  
        "id": 1,  
        "name": "Martin Odersky",  
        "groups": [  
          {  
            "id": 1,  
            "name": "Administrators"  
          },  
          {  
            "id": 2,  
            "name": "Users"  
          }  
        ]  
      },  
      {  
        "id": 2,  
        "name": "James Gosling",  
        "groups": [  
          {  
            "id": 1,  
            "name": "Administrators"  
          },  
          {  
            "id": 2,  
            "name": "Users"  
          }  
        ]  
      }  
    ]  
  }  
}
```

GraphiQL



Prettify

```
1 query {  
2  
3 }
```

GraphiQL



Prettify

```
1 query {  
2   Users  
3 }
```

GraphiQL



Prettify

```
1 query {  
2   Users (id: [1,2])  
3 }
```

GraphiQL



Prettify

```
1 query {  
2   Users (id: [1,2], page: 1, limit: 10)  
3 }
```

GraphiQL



Prettify

```
1 query ($page: Int, $limit: Int) {  
2   Users (id: [1,2], page: $page, limit: $limit)  
3 }
```

QUERY VARIABLES

```
1 {  
2   "page": 1,  
3   "limit": 10  
4 }
```



```
1 query ($page: Int, $limit: Int) {  
2   Users (id: [1,2], page: $page, limit: $limit) {  
3     id,  
4     name  
5   }  
6 }
```

## QUERY VARIABLES

```
1 {  
2   "page": 1,  
3   "limit": 10  
4 }
```



```
1 query ($page: Int, $limit: Int) {  
2   Users (id: [1,2], page: $page, limit: $limit) {  
3     id,  
4     name @include (if: true)  
5   }  
6 }
```

## QUERY VARIABLES

```
1 {  
2   "page": 1,  
3   "limit": 10  
4 }
```





```
1 ▾ query ($page: Int, $limit: Int) {  
2 ▾   Users (id: [1,2], page: $page, limit: $limit) {  
3     id,  
4     name @include (if: true),  
5 ▾   friends {  
6     id,  
7     name,  
8 ▾   friends {  
9     id,  
10    name,  
11 ▾   friends {  
12    id,  
13    name,  
14    friends {  
15    id,  
16    name  
17    }  
18  }  
19 }  
20 }
```

## QUERY VARIABLES

```
1 {  
2   "page": 1,  
3   "limit": 10  
4 }
```

# Недостатки graphql

- Тяжелые запросы
- Бедная экосистема
- Пользователи

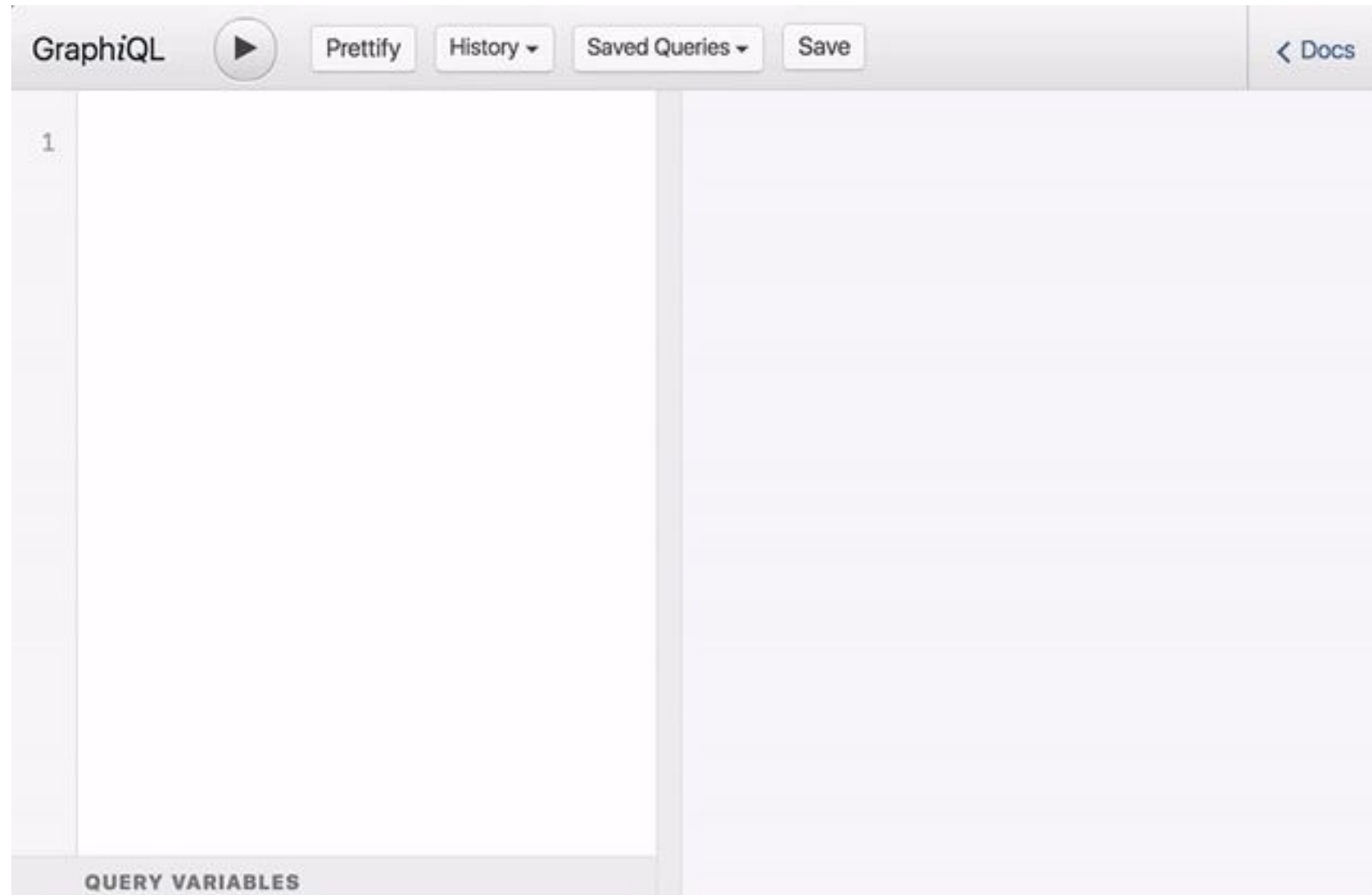
# GraphQL сервер

- Endpoint ( /api/v1 )
- Сервер (Свой или готовая имплементация)
- Схема

# Полезности

- <http://graphql.org>
- Материалы на хабре
- <https://github.com/graphql/graphiql>

# graphiql



# Полезности

- <http://graphql.org>
- Материалы на хабре
- <https://github.com/graphql/graphiql>
- <https://www.howtographql.com>

# howtographql.com

HOW TO GRAPHQL Search tutorials...

GraphQL Frontend Backend

FRONTEND BACKEND

Apollo  
SOON

Ember + Apollo

Angular + Apollo

Vue + Apollo

React + Relay

**React + Apollo**  
BEGINNERS'S CHOICE

graphql.js

graphql-elixir

graphql-ruby

graphql-java

graphql

**React + Apollo**  
Learn how to get started with React and Apollo Client on the frontend

- Introduction
- Getting Started
- Queries: Loading Links
- Mutations: Creating Links
- Routing
- Authentication
- More Mutations and Updating the Store
- Filtering: Searching the List of Links
- Realtime Updates with GraphQL Subscriptions
- Pagination
- Summary

Спасибо!

Вопросы?