

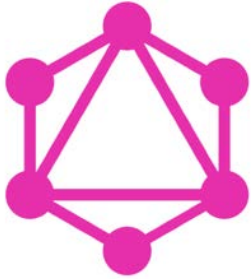
# GraphQL: Getting Started



Samer Buna

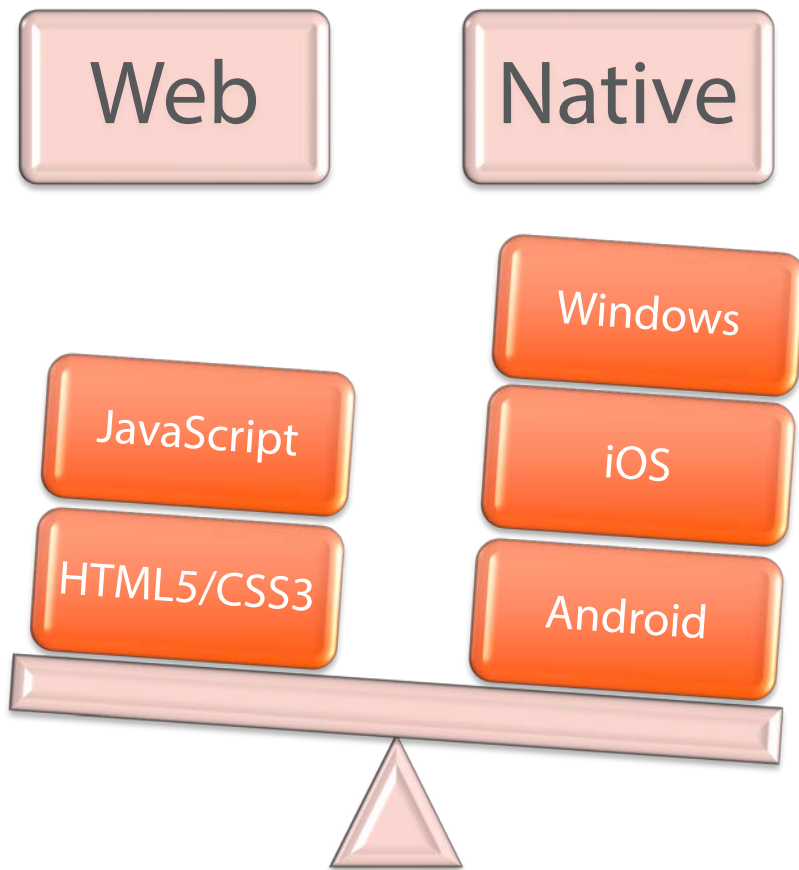
@samerbuna | [ReactjsCamp.com](https://ReactjsCamp.com)

# Why GraphQL

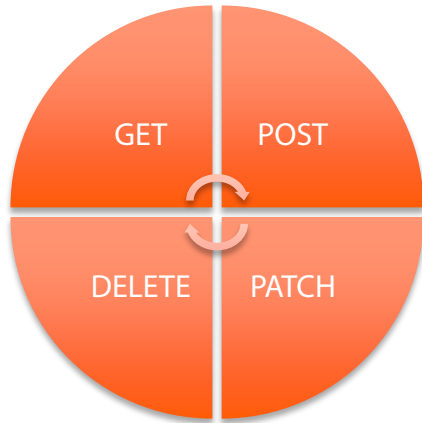


Facebook and other big players

- 1) Data communication performance
- 2) Developers Experience



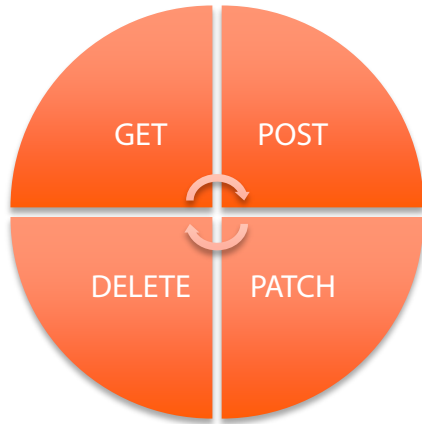
# REST APIs



“There are 2 types of people in the world, those who try to build REST APIs, and those who scream at them for not building proper REST APIs”

– Nick Schrock

# REST APIs



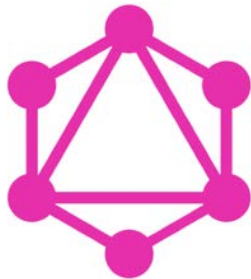
Clients depend on the server

Menu in a restaurant

Custom endpoints

Grocery store: only 1 item per visit

# DI/DX



Product-developers in mind  
How we think about the UI  
Declarative

# Declarative vs. Imperative

List of Jane's friends

Ask /users/Jane/friends

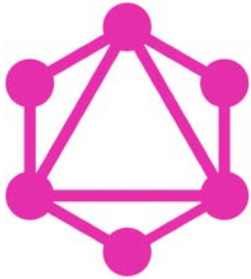
+ Their names

Ask /friends/{ID}

+ Which country they are from

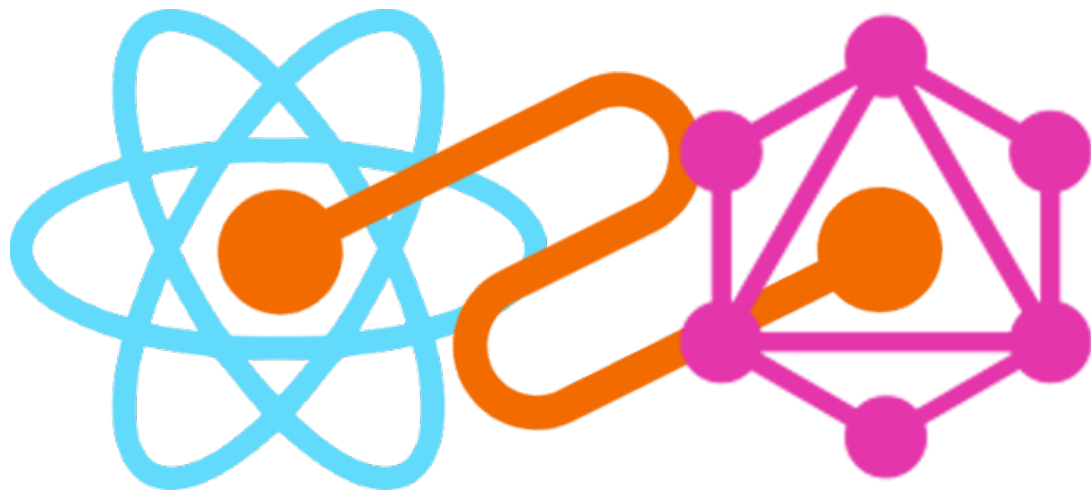
Ask /Countries/{ID}

# What Is GraphQL



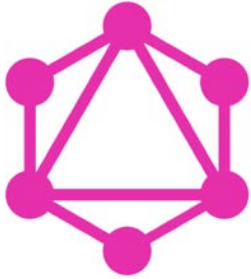
Express data needs hierarchically  
Get data with a single round trip





# Graph*i*QL

# What Is GraphQL



Execution engine on server

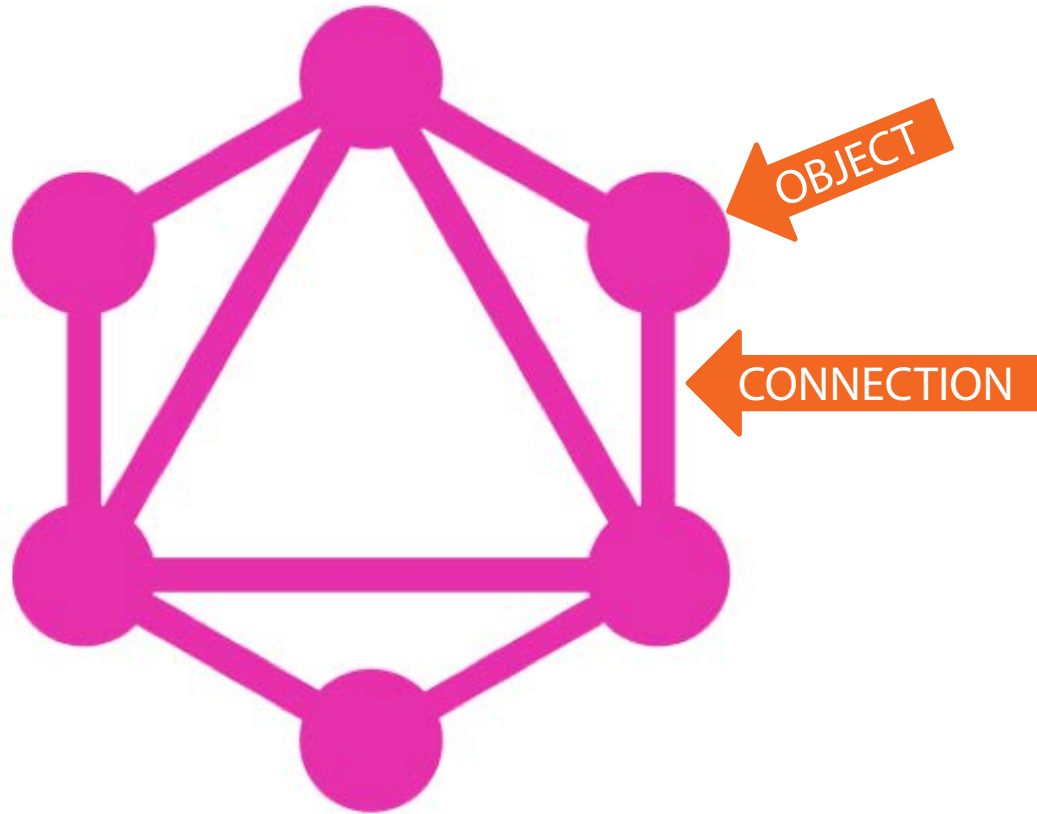
Query language on client

The lingua franca of data communication

# JSON Without the Values

```
{  
  language {  
    name  
  }  
}
```

```
{  
  "language": {  
    "name": "JavaScript"  
  }  
}
```

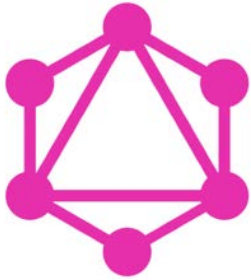


# Graph: Objects and Connections

```
{
  person(role: 'Mockingjay') {
    name,
    friends {
      name
    },
    enemies: {
      name
    }
  }
}
```

```
{
  "person": {
    "name": "Katniss",
    "friends": [
      { "name": "Peeta" },
      { "name": "Finnick" },
      { "name": "Gale" }
    ],
    "enemies": [
      { "name": "Snow" }
    ]
  }
}
```

# Reasons Why GraphQL Is a Big Deal - #1



Easy-to-write queries  
if you know the data  
that you need

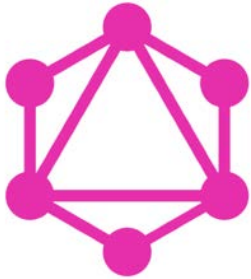
# Graph: Objects and Connections

```
{
  person(role: 'Mockingjay') {
    name,
    friends {
      name
    },
    enemies: {
      name
    }
  }
}
```

```
{
  "person": {
    "name": "Katniss",
    "friends": [
      { "name": "Peeta" },
      { "name": "Finnick" },
      { "name": "Gale" }
    ],
    "enemies": [
      { "name": "Snow" }
    ]
  }
}
```



## Reasons Why GraphQL Is a Big Deal - #2



The queries describe  
the shape of the  
response

# Properties and Selection Sets

```
{  
  language {  
    name  
  }  
}
```

```
{  
  "language": {  
    "name": "JavaScript"  
  }  
}
```

# Field Arguments

```
{  
  employee(id: 42) {  
    name,  
    age  
  }  
}
```

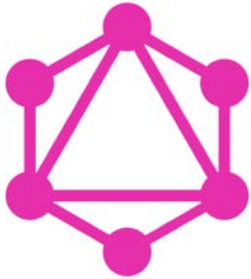
```
{  
  "employee": {  
    "name": "George",  
    "age": 19,  
  }  
}
```

# Complex Fields

```
{  
  employee(id: 42) {  
    name,  
    boss {  
      name  
    }  
  }  
}
```

```
{  
  "employee": {  
    "name": "George",  
    "boss": {  
      "name": "Michael"  
    }  
  }  
}
```

## Reasons Why GraphQL Is a Big Deal - #3



No over-fetching

No under-fetching

# Complex Fields

```
{  
  employee(id: 42) {  
    name,  
    boss(level: 2) {  
      name  
    }  
  }  
}
```

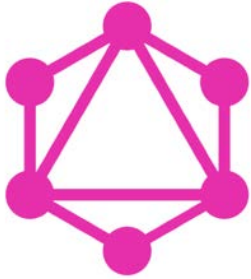
```
{  
  "employee": {  
    "name": "George",  
    "boss": {  
      "name": "Oscar"  
    }  
  }  
}
```

# Aliases

```
{
  employee(id: 42) {
    name,
    directBoss: boss(level: 1) {
      name
    },
    bigBoss: boss(level: 2) {
      name
    }
  }
}
```

```
{
  "employee": {
    "name": "George",
    "directBoss": {
      "name": "Michael"
    },
    "bigBoss": {
      "name": "Oscar"
    }
  }
}
```

## Reasons Why GraphQL Is a Big Deal - #4



You can customize the data that you get with aliases and field arguments



# Graph: Objects and Connections

```
{
  person(role: 'Mockingjay') {
    name,
    friends {
      name
    },
    enemies: {
      name
    }
  }
}
```

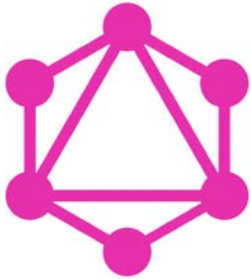
```
{
  "person": {
    "name": "Katniss",
    "friends": [
      { "name": "Peeta" },
      { "name": "Finnick" },
      { "name": "Gale" }
    ],
    "enemies": [
      { "name": "Snow" }
    ]
  }
}
```

# As Deep as Exactly Needed

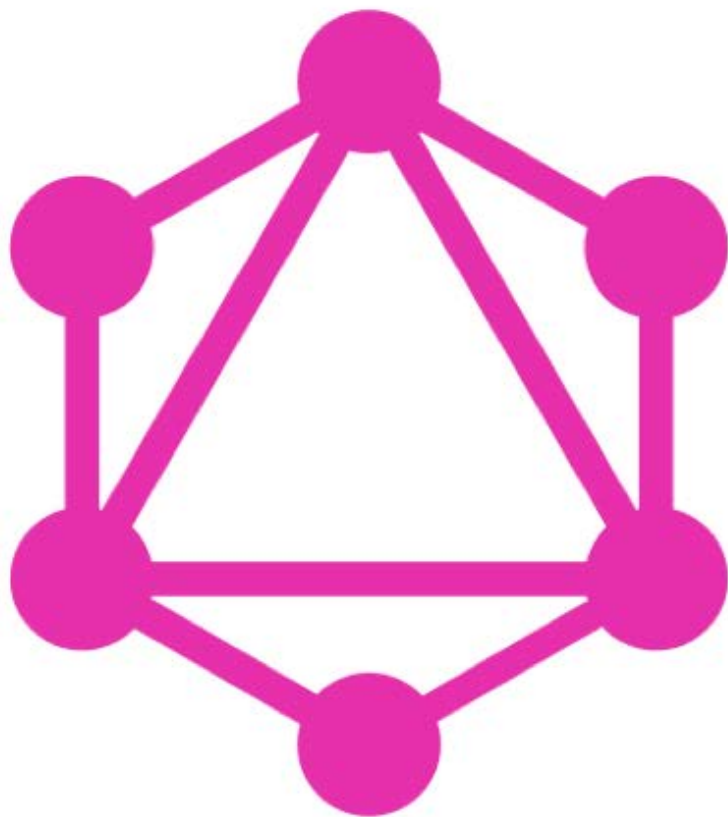
```
{
  person(role: 'Mockingjay') {
    name,
    friends {
      name,
      skills {
        name
      }
    }
  }
}
```

```
{
  "person": {
    "name": "Katniss",
    "friends": [
      { "name": "Peeta",
        "skills": [...] },
      { "name": "Finnick",
        "skills": [...] },
      { "name": "Gale",
        "skills": [...] }
    ]
  }
}
```

## Reasons Why GraphQL Is a Big Deal - #5



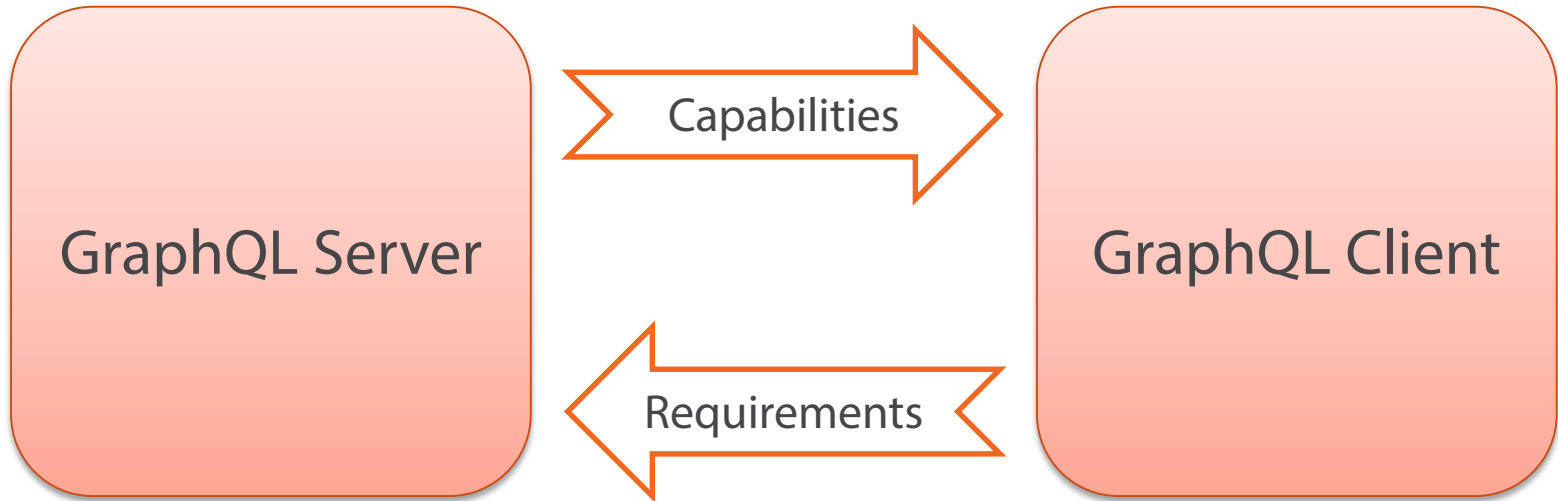
You can query objects and their connections, and nest as deep as you need, and get all that data in a single round trip



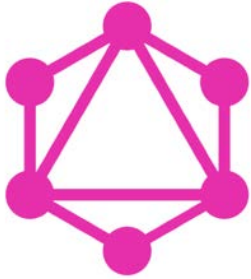
# GraphQL Core Principles

# Mental Model

# Type System



## Reasons Why GraphQL Is a Big Deal - #6

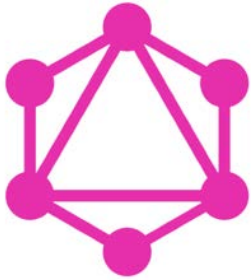


Views can change their data requirement and the server might not need to be changed



# Introspection

# Reasons Why GraphQL Is a Big Deal - #7



Built-in accurate and  
always up-to-date  
documentation

# Composition

# Fragments

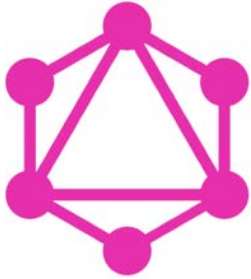
```
{  
  employee(id: 42) {  
    name,  
    age  
    directBoss: boss(level: 1) {  
      name,  
      age  
    },  
    bigBoss: boss(level: 2) {  
      name,  
      age  
    }  
  }  
}
```

# Fragments

```
fragment personInfo on Employee {  
    name  
    age  
}
```

```
{  
    employee(id: 42) {  
        ...personInfo  
        directBoss: boss(level: 1) {  
            ...personInfo  
        },  
        bigBoss: boss(level: 2) {  
            ...personInfo  
        }  
    }  
}
```

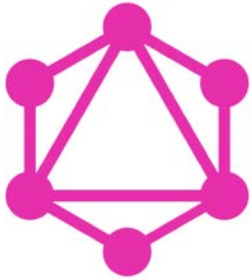
# Reasons Why GraphQL Is a Big Deal - #8



Queries are  
composable and re-  
usable

# Not a Storage Engine

## Reasons Why GraphQL Is a Big Deal - #9

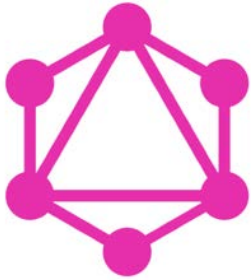


Queries can re-use  
your existing code



# Mutations

## Reasons Why GraphQL Is a Big Deal - #10



Mutations are just like queries, same structure, same benefits

# Summary

## GraphQL Is a Big Deal