

GraphQL



Demystified

Martynas Bardauskas

Front-end developer @ Wix

building Wix's mobile presence with React Native

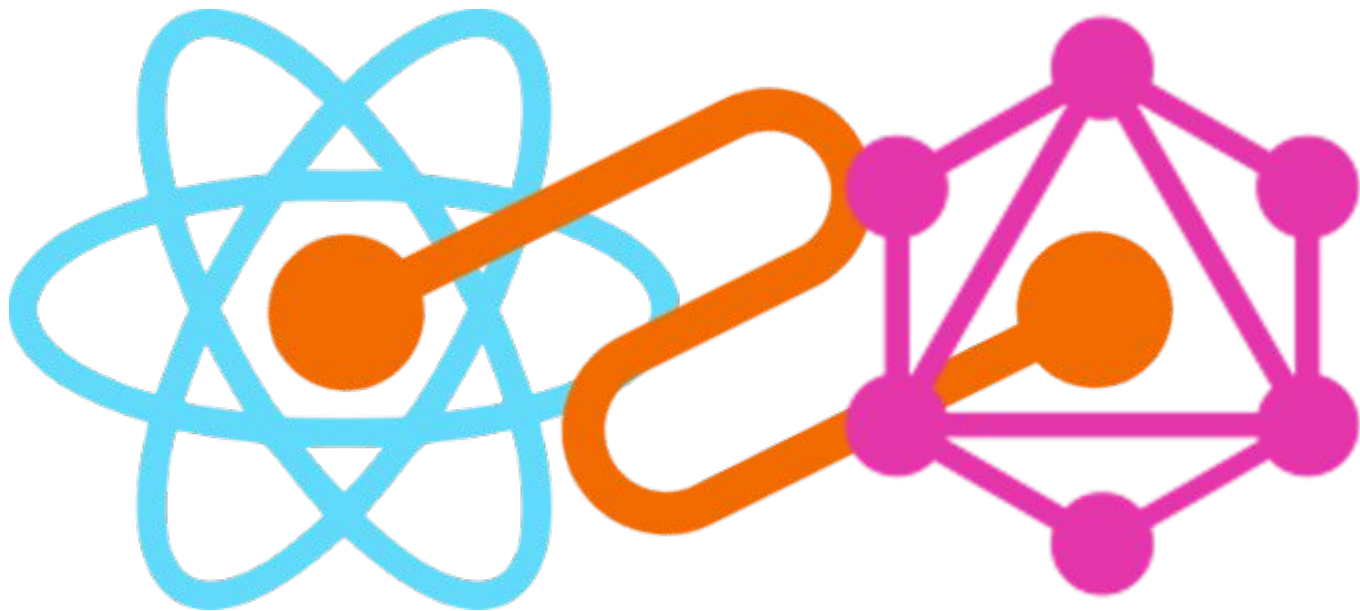


github.com/mbardauskas



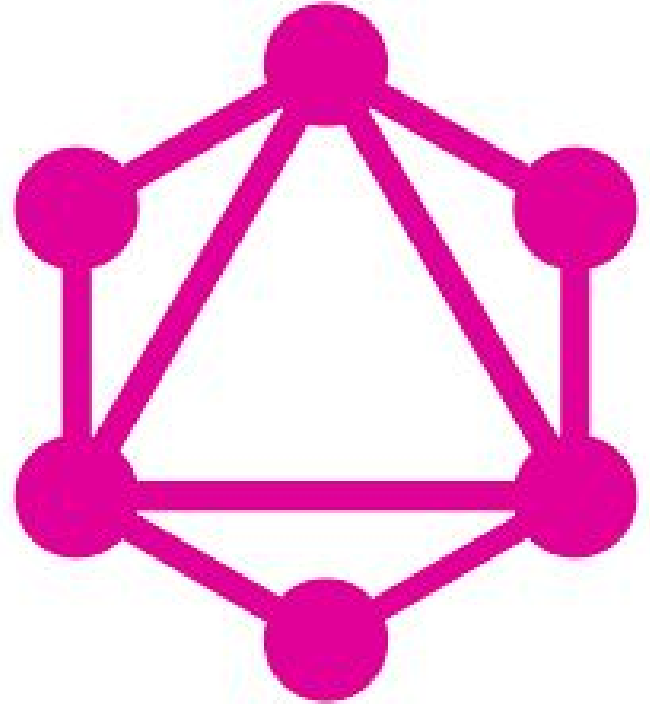
[@mbardauskas](https://twitter.com/mbardauskas)

Facebook development stack



Today's talk: GraphQL

Back-end



**Why are we
using GraphQL?**

We're building a mobile App

Mobile networks tend to have bad latency.

Goal: fast data fetches and less network requests

We connect a lot of in-house services

15 and counting

Goal: combine resources, avoid maintenance hell

We need a fast evolving API

We've released our initial app in 86 days with impressive feature set

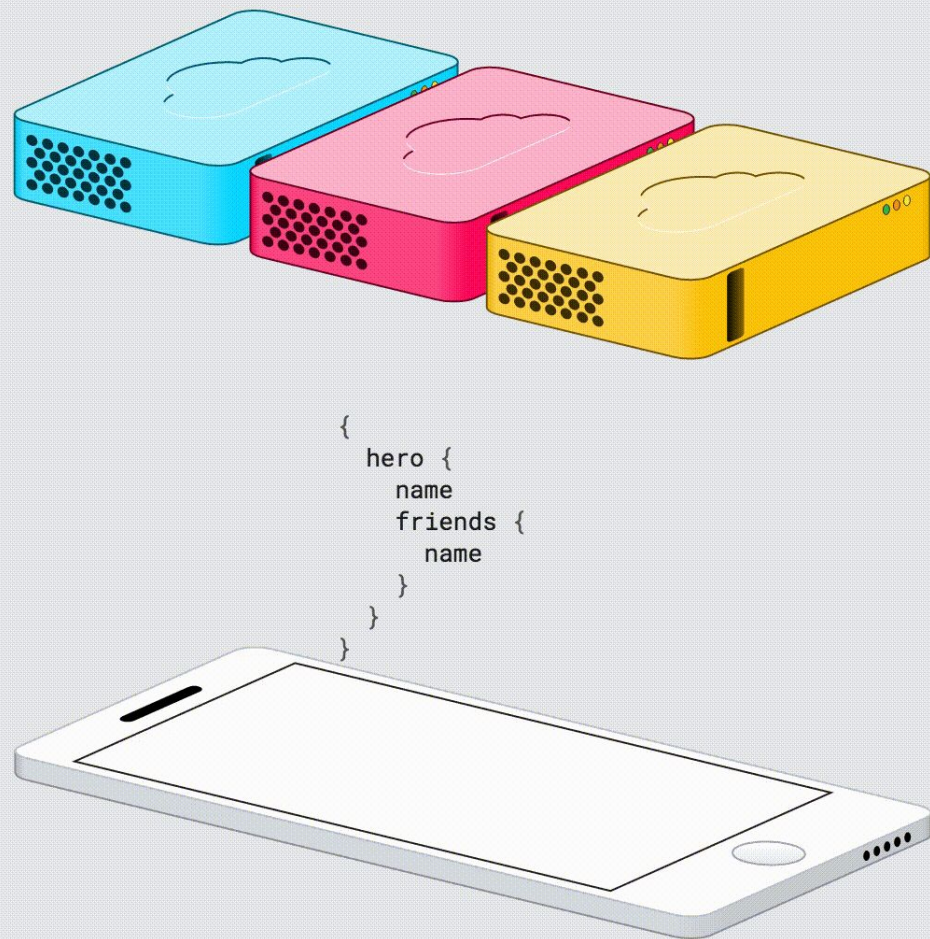
Goal: backwards compatibility, minimised tech debt

**How does
GraphQL help?**

Get many resources in a single request

It helps us have less network requests.

GraphQL queries access not just the properties of one resource but also smoothly follow references between them. While typical REST APIs require loading from multiple URLs, GraphQL APIs get all the data your app needs in a single request. Apps using GraphQL can be quick even on slow mobile network connections.



A query language for your API

It helps us have predictable results.

Send a GraphQL query to your API and get exactly what you need, nothing more and nothing less.

```
{  
  hero {  
    name  
  }  
}
```

```
{  
  "hero": {  
    "name": "Luke Skywalker"  
  }  
}
```

Describe what's possible with a type system

It helps us to ensure Apps only ask for what's possible and provides clear and helpful errors.

Provides data shape.

GraphQL APIs are organized in terms of types and fields, not endpoints. Access the full capabilities of your data from a single endpoint.

```
{
  hero {
    name
    friends {
      name
      homeWorld {
        name
        climate
      }
      species {
        name
        lifespan
        origin {
          name
        }
      }
    }
  }
}

type Query {
  hero: Character
}

type Character {
  name: String
  friends: [Character]
  homeWorld: Planet
  species: Species
}

type Planet {
  name: String
  climate: String
}

type Species {
  name: String
  lifespan: Int
  origin: Planet
}
```

Evolve your API without versions

It helps us with deprecation mechanism and freedom to extend existing types when treating backwards compatibility.

Add new fields and types to your GraphQL API without impacting existing queries. Aging fields can be deprecated and hidden from tools. GraphQL APIs give apps continuous access to new features and encourage cleaner, more maintainable server code.

```
type Film {  
  title: String  
  episode: Int  
  releaseDate: String  
  
}
```

More features



- **GraphiQL**
GUI tool to write queries. Provides docs from your types.
- **Your own data and code**
Mix and match your data sources by combining APIs / data storage / any other function.
- **Mutations**
Mutates data instead of getting it.
- **Typed inputs**
Helps by providing types, built-in validation and helpful errors

Pain points

Denial of Service attacks

Not specific to GraphQL, but when working with GraphQL, we have to be extra careful about them, since they can have ripple effects throughout all used services

N+1 SQL queries

GraphQL query fields are designed to be stand-alone functions. Resolving fields might result in a new database request per resolved field

DataLoader uses a combination of batching and caching to avoid such issues

Cache

RESTful API can be cached by location, GraphQL is limited

With GraphQL, cache handling spans to both Client (Relay.js) and Server (GraphQL).

Ownership

Specific to middleware servers

Monitoring

REST API can be monitored by location

You need your own mechanism to monitor

Live example

SWAPI

API endpoints

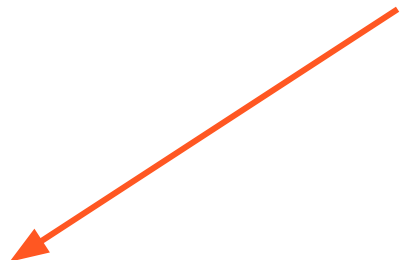
```
// http://swapi.co/api/films/1
{
  "title": "A New Hope",
  "episode_id": 4,
  "release_date": "1977-05-25",
  "characters": [
    "http://swapi.co/api/people/1/",
    "http://swapi.co/api/people/2/",
    "http://swapi.co/api/people/3/",
    "http://swapi.co/api/people/4/",
    "http://swapi.co/api/people/5/",
    "http://swapi.co/api/people/6/",
    // ... other characters
  ],
  // ... other data
}
```

```
// http://swapi.co/api/people/1
{
  "name": "Luke Skywalker",
  "mass": "77",
  "hair_color": "blond",
  "skin_color": "fair",
  "eye_color": "blue",
  "birth_year": "19BBY",
  "gender": "male",
  "homeworld": "http://swapi.co/api/planets/1/",
  "films": [
    "http://swapi.co/api/films/6/",
    "http://swapi.co/api/films/3/",
    // ... other films
  ],
}
```

What we'll do

```
// http://swapi.co/api/films/1
```

```
{  
  "title": "A New Hope",  
  "episode_id": 4,  
  "release_date": "1977-05-25",  
  "characters": [  
    {  
      "name": "Luke Skywalker",  
      "mass": "77",  
      "hair_color": "blond",  
      // ... other fields  
    },  
    // ... other characters  
  ],  
  // ... other data  
}
```



```
// http://swapi.co/api/people/1
```

```
{  
  "name": "Luke Skywalker",  
  "mass": "77",  
  "hair_color": "blond",  
  "skin_color": "fair",  
  "eye_color": "blue",  
  "birth_year": "19BBY",  
  "gender": "male",  
  "homeworld": "http://swapi.co/api/planets/1/",  
  "films": [  
    "http://swapi.co/api/films/6/",  
    "http://swapi.co/api/films/3/",  
    // ... other films  
  ],  
}
```


Demo

<https://github.com/mbardauskas/graphql-demo>

Thank you

Martynas Bardauskas

Front-end developer @ Wix

building Wix's mobile presence with React Native



github.com/mbardauskas



[@mbardauskas](https://twitter.com/mbardauskas)

BSD 3-Clause License

LICENSE AGREEMENT For graphql.org software

Facebook, Inc. ("Facebook") owns all right, title and interest, including all intellectual property and other proprietary rights, in and to the graphql.org software. Subject to your compliance with these terms, you are hereby granted a non-exclusive, worldwide, royalty-free copyright license to (1) use and copy the graphql.org software; and (2) reproduce and distribute the graphql.org software as part of your own software ("Your Software"). Facebook reserves all rights not expressly granted to you in this license agreement.

THE SOFTWARE AND DOCUMENTATION, IF ANY, ARE PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES (INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE) ARE DISCLAIMED. IN NO EVENT SHALL FACEBOOK OR ITS AFFILIATES, OFFICES, DIRECTORS OR EMPLOYEES BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You will include in Your Software (e.g., in the file(s), documentation or other materials accompanying your software): (1) the disclaimer set forth above; (2) this sentence; and (3) the following copyright notice:

Copyright (c) 2015, Facebook, Inc. All rights reserved.