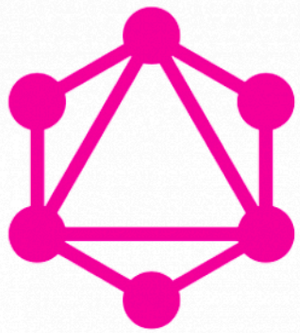


GraphQL

Understanding GraphQL

BY: CURT GRATZ

What we won't be covering

- Why GraphQL is the cats pajamas
- Why you should never make a REST API again
- Advance GraphQL topics like fragments, parameterizing fields with arguments, Dataloaders, etc
- How to prove the theory of relativity

What we will be covering

- Where GraphQL came from
- Some advantages of using GraphQL
- When its good or not good to use GraphQL
- A few key concepts to understand (Queries, Mutations, etc)
- A RB centric example of a possible GraphQL API

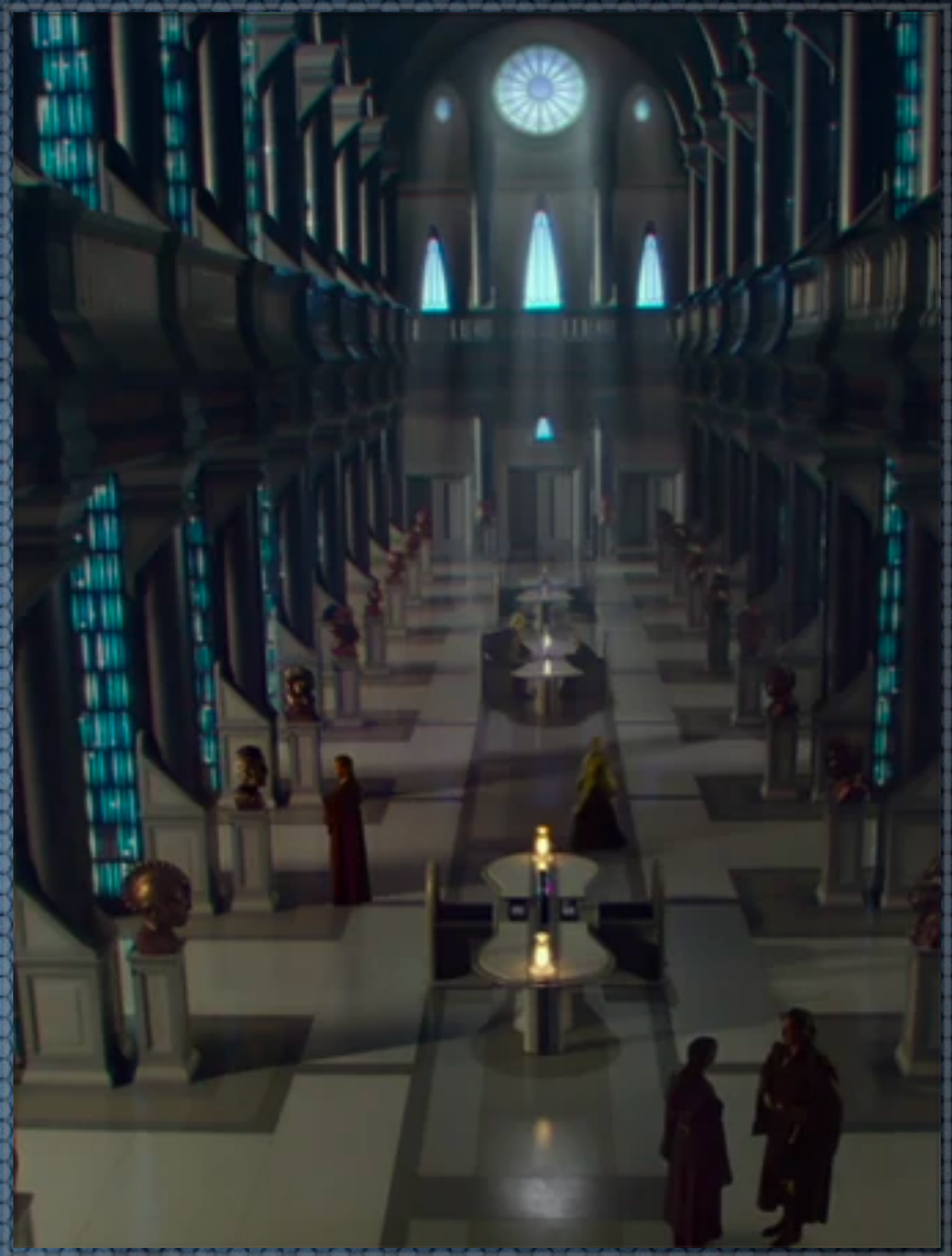
The Obligatory Who Am I?

- Husband
- Dad
- Coach (CC and Track)
- Co-Owner of Computer Know How
- Runner
- Developer lead of the CDH team

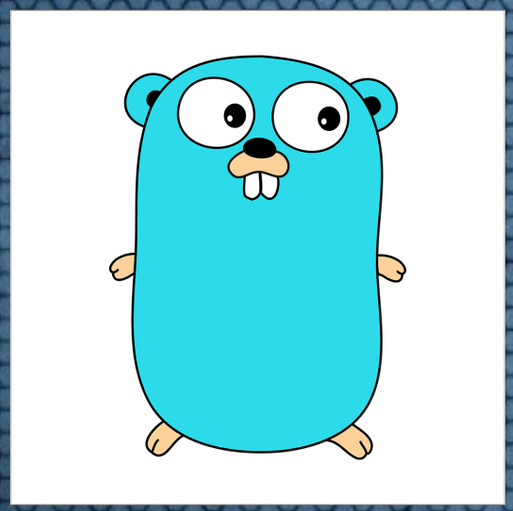
COMPUTER KNOW HOW™

Brief History

- Open source specification
- Developed by Facebook
- Initial Release in 2015
- Stable Release in 2018
- Spec has been stable since June of 2018



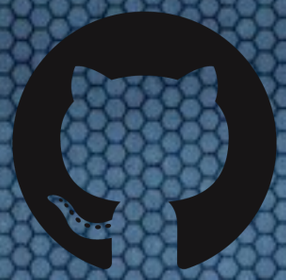
Language Support



And many more…

Who uses GraphQL













And many more…

What is GraphQL

- “A query language for your API”
- “Get many resources in a single request”
- “Describe what’s possible with a type system”

What is GraphQL

Describe your data

```
type Project {
  name: String
  tagline: String
  contributors: [User]
}
```

Ask for what you want

```
{
  project(name: "GraphQL") {
    tagline
  }
}
```

Get predictable results

```
{
  "project": {
    "tagline": "A query language for APIs"
  }
}
```

REST API

- /characters
- /characters/:id/movies

REST API /characters


```
[
    {
        "id": 1,
        "name": "Han Solo",
        "age": 35,
        "address": "124 Solo Lane"
        ...
    },
    {
        "id": 2,
        "name": "Luke Skywalker",
        "age": 21,
        "address": "124 Skywalker Way"
        ...
    },
    ...
]
```

REST API /characters/1/movies


```
[
    {
        "id": 1,
        "name": "Empire Strikes Back",
        "releaseDate": "May 1, 1980",
        "director": "George Lucas"
        ...
    },
    {
        "id": 2,
        "name": "Solo: A Star Wars Story",
        "releaseDate": "May 10, 2018",
        "director": "Ron Howard"
        ...
    },
    ...
]
```

REST API /characters/2/movies


```
[
    {
        "id": 1,
        "name": "Empire Strikes Back",
        "releaseDate": "May 1, 1980",
        "director": "George Lucas"
        ...
    },
    {
        "id": 3,
        "name": "A New Hope",
        "releaseDate": "May 25, 1977",
        "director": "George Lucas"
        ...
    },
    ...
]
```

REST API

- So because we know that is an issue we make a new endpoint
- /charactersNamesWithMovies

REST API / charactersNamesWithMovies


```
[
    {
        "id": 1,
        "name": "Han Solo",
        "movies": [
            {
                "id": 1,
                "name": "Empire Strikes Back"
            },
            {
                "id": 3,
                "name": "A New Hope"
            },
            ...
        ]
    },
    {
        "id": 2,
        "name": "Luke Skywalker",
        "movies": [
            {
                "id": 1,
                "name": "Empire Strikes Back"
            },
            {
                "id": 2,
                "name": "Solo: A Star Wars Story"
            },
            ...
        ]
    },
    ...
]
```

REST API

- What if the requirements change and now we want Age and Director
- /v2/charactersNamesWithMovies
- or
- /charactersNamesAgeWithMoviesDirectors

REST API / charactersNamesAgeWithMoviesDirectors


```
[
    {
        "id": 1,
        "name": "Han Solo",
        "age": 35,
        "movies": [
            {
                "id": 1,
                "name": "Empire Strikes Back",
                "director": "George Lucas"
            },
            {
                "id": 2,
                "name": "Solo: A Star Wars Story",
                "director": "Ron Howard"
            },
            ...
        ]
    },
    {
        "id": 2,
        "name": "Luke Skywalker",
        "age": 21,
        "movies": [
            {
                "id": 1,
                "name": "Empire Strikes Back",
                "director": "George Lucas"
            },
            {
                "id": 3,
                "name": "A New Hope",
                "director": "George Lucas"
            },
            ...
        ]
    },
    ...
]
```

GraphQL

- This can be done with one call, by sending in GraphQL what is called a query
- This is the type of problems GraphQL looks to solve

GraphQL Query

```
query {
    characters {
        id
        name
        movies {
            id
            name
        }
    }
}
```

GraphQL Query


```
[
    {
        "id": 1,
        "name": "Han Solo",
        "movies": [
            {
                "id": 1,
                "name": "Empire Strikes Back"
            },
            {
                "id": 3,
                "name": "A New Hope"
            },
            ...
        ]
    },
    {
        "id": 2,
        "name": "Luke Skywalker",
        "movies": [
            {
                "id": 1,
                "name": "Empire Strikes Back"
            },
            {
                "id": 2,
                "name": "Solo: A Star Wars Story"
            },
            ...
        ]
    },
    ...
]
```

GraphQL Query

```
query {
    characters {
        id
        name
        age
        movies {
            id
            name
            director
        }
    }
}
```

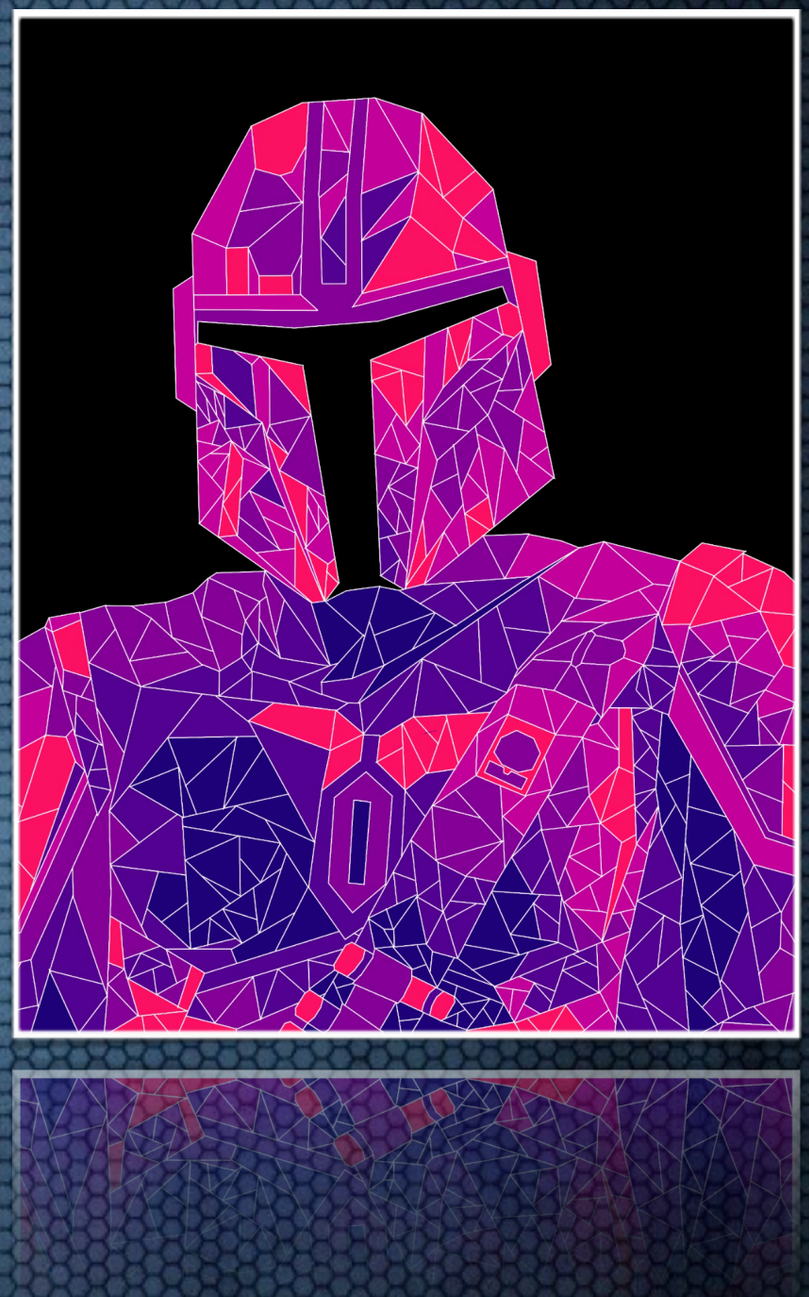
GraphQL Query


```
[
    {
        "id": 1,
        "name": "Han Solo",
        "age": 35,
        "movies": [
            {
                "id": 1,
                "name": "Empire Strikes Back",
                "director": "George Lucas"
            },
            {
                "id": 2,
                "name": "Solo: A Star Wars Story",
                "director": "Ron Howard"
            },
            ...
        ]
    },
    {
        "id": 2,
        "name": "Luke Skywalker",
        "age": 21,
        "movies": [
            {
                "id": 1,
                "name": "Empire Strikes Back",
                "director": "George Lucas"
            },
            {
                "id": 3,
                "name": "A New Hope",
                "director": "George Lucas"
            },
            ...
        ]
    },
    ...
]
```

GraphQL Vocabulary

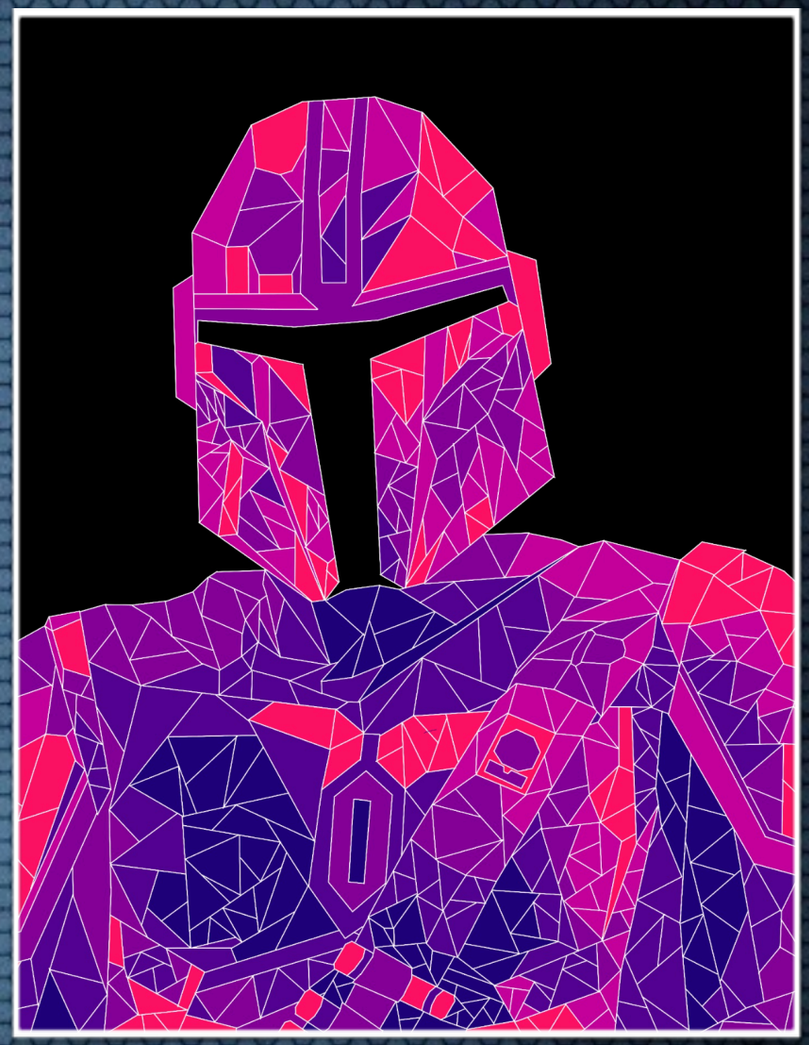
GraphQL Vocabulary

- Schema
  - Describes the functionality available to the clients which connect to it.
- Field
  - A unit of data you are asking for/ defining in a Schema
- Object Type
  - A type in a GraphQL schema that has fields.

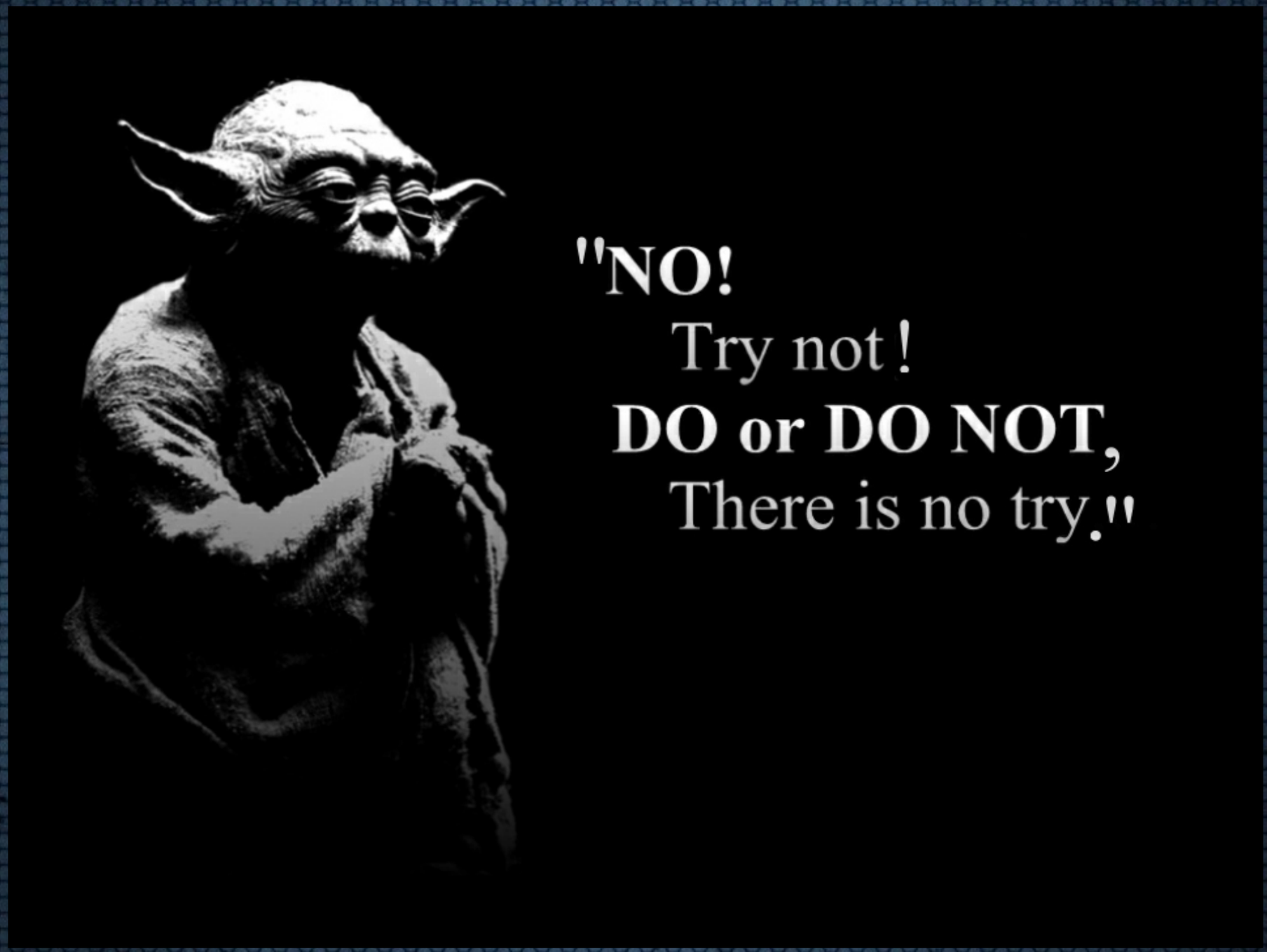


GraphQL Vocabulary

- Query
  - A read-only fetch operation to request data from a GraphQL service.
- Mutation
  - An operation for creating, modifying and destroying data.
- Resolver
  - A function that connects schema to data. (Call to a DB, etc)



Simple Example



Specs

- Imagine a site that wants to sell heavy equipment via auctions
- Design a GraphQL API for the auction listings
  - Listings belong to a certain category
  - Listings have 1 or more pictures

Demo Time

GraphQL is not a silver bullet

- Very simple API or schema
  - Introduces overhead
- Very complex API or schema
  - Can be hard to get the schema right
  - Can do a partial GraphQL/partial REST system
- Authentication concerns, etc
  - REST handles this better IMHO
- Watch out for performance
  - Can be more or less performant depending on design

GraphQL Performance Concerns

- Care needs to be taken in your Resolvers
- Take advantage of caching
- Watch for N+1 problem
  - DataLoader packages can help
  - Customizing resolvers
- Don’t write your resolvers like you would for REST endpoints
  - You know the data the client wants, get just that

More Information

- GraphQL Website
  - https://graphql.org/
- Apollo Website
  - https://www.apollographql.com/
- What Is GraphQL? by LevelUpTuts
  - https://www.youtube.com/watch?v=VjXb3PRL9WI
- GraphQL Full Course - Novice to Expert by FreeCodeCamp
  - https://www.youtube.com/watch?v=ed8SzALpx1Q

“Thank you.”

–hold up applause sign here

Contact Info

- Email - gratzc@compknowhow.com
- Blog - https://ckhconsulting.com/blog/
- Twitter - gratzc
- Instagram - gratzc
- League of Legends - gratzc