

# GraphQL Framework

Nicholas Yang

June 21, 2018

## Contents

|          |                         |          |
|----------|-------------------------|----------|
| <b>1</b> | <b>Idea</b>             | <b>1</b> |
| <b>2</b> | <b>Key Features</b>     | <b>1</b> |
| <b>3</b> | <b>Design Questions</b> | <b>2</b> |

## 1 Idea

A framework based around GraphQL that takes the ease of use of Rails and strips out all the crud. Allows for fast, easy to build APIs

## 2 Key Features

- Easy GraphQL Object Type definition (integrated with classes ideally)
- Automated queries:

```
class Article < GraphQL::Object
  query_on :title # 9Makes the query article_by_title
end
```

- Well integrated ORM for automatic field resolving:

```
class Article < GraphQL::Object

  # Fields
  field :title, types.String
  field :title, types[Comment]
```

```

# Queries
query_on :title

# Associations
belongs_to :section
has_many :comments # Could be optional, since fields already specifies this
end

class Comment < GraphQL::Object
  # Fields
  field :content, types.String
end

```

Should allow for:

```

article_by_title {
  title
  comments {
    content
  }
}

```

- GraphQL by default, no REST (not that REST is bad, just that GraphQL always feels tacked on)
- Super nice code generation. Able to introspect and see what generators have been applied (migrations for generators?)
- Decent JWT support
- Authorization levels
- Instead of Model-View-Controller, how about Model-Resolver-Query? Or Model-Resolver-Query-Mutation?

### 3 Design Questions

- What language? Ruby is probably easiest, but least popular.
  - Could start in one language then port over to others like JS

- Try Crystal? Elixir?
- What name? Lattice? Ruby on Lattice?
- How should middleware work?
- How can we get authorization and authentication to work smoothly?
- How can we improve error handling from GraphQL-Ruby?