The introduction about GraphQL in iOS

Presenter: Duy Doan

Danang, 01/06/2018

Agenda

- The introduction about GraphQL
- How to use GraphQL in iOS
- Showing an example about GraphQL in iOS
- The common problems and how to deal when working with GraphQL
- Q&A

The introduction about GraphQL

- What is GraphQL?
  - GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.
- What difference between GraphQL and REST?

The introduction about GraphQL

/countries


```
[
    {
        "id":"1",
        "name":"Vietnam",
        "population":95414640
    },
    {
        "id":"2",
        "name":"Thailand",
        "population":68863514
    }
]
```


/countries/<id>/cities


```
[
    {
        "id":"1",
        "name":"AnGiang",
        "area":35367
    },
    {
        "id":"2",
        "name":"CaMau",
        "area":52949
    }
]
```

The introduction about GraphQL

- Problem: How do clients get first two cities in counties ?

```
[  {
      "id":"1",
      "name":"Vietnam",
      "cities":[
         {
            "id":"1",
            "name":"DaNang"
         },
         {
            "id":"2",
            "name":"DienBien"
         }
      ]
   }]
```

The introduction about GraphQL

- REST
  - Solution 1:
    - Create multiple requests and then calculate to find the right result afterward
      /countries/1/cities
      /countries/2/cities
      /countries/3/cities
      /countries/n/cities

The introduction about GraphQL

- REST
  - Solution 2:
    - Talk to back-end developers to adjust the API in the proper way

/countries

```
[   {
        "id":"1",
        "name":"Vietnam",
        "cities":[
            {
                "id":"1",
                "name":"DaNang"
            },
            …
    }]
```

The introduction about GraphQL

- GraphQL
  - Solution:
    - Just change a little in queries to get the right result

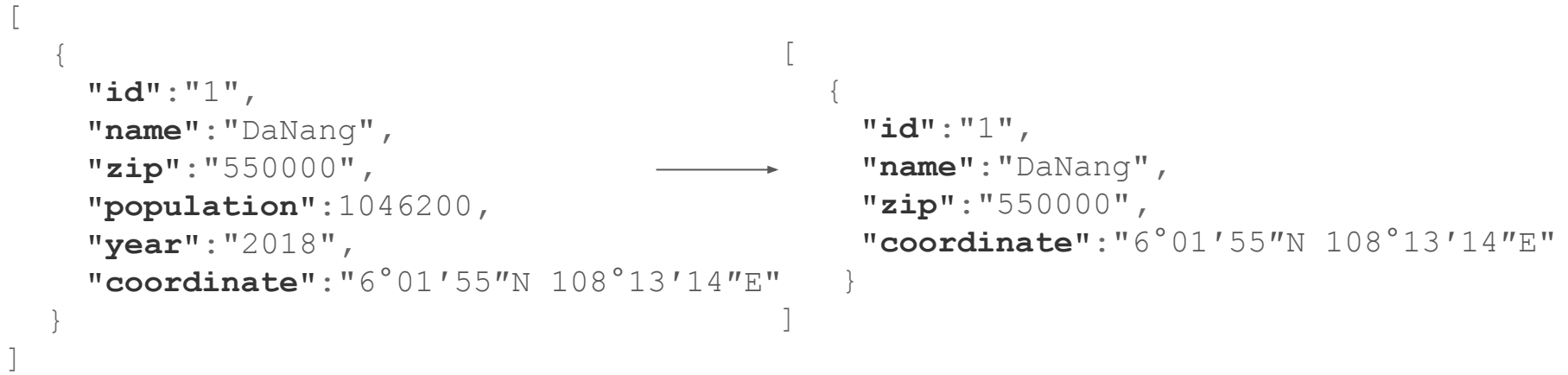
```
allCountries {
    id
    name
    cities(first: 2) {
      id
      name
    }
  }
```



```
[  {
      "id":"1",
      "name":"Vietnam",
      "cities":[
         {
            "id":"1",
            "name":"DaNang"
         },
         …
}]
```

The introduction about GraphQL

- Problem: How do clients reduce some fields in the response data ?



The introduction about GraphQL

- REST
  - Solution:
    - Talk to back-end developers to adjust the API
- GraphQL
  - Solution
    - Just change a queries to get the right result

```
allCities {
   id
   name
   zip
   coordinate }
```



```
{
 "id":"1",
 "name":"DaNang",
 "zip":"550000",
 "coordinate":"6°01′55″N 108°13′14″E"
}
```

The introduction about GraphQL

- We can ***READ*** data by ***queries***

```
query AllCities {
  allCities {
    id
    name
    zip
  }
}
```



```
[
 {
  "id":"1",
  "name":"DaNang",
  "zip":"550000",
  }
]
```

The introduction about GraphQL

- We can ***WRITE & READ*** data by ***mutations***

```
mutation CreateCity {
    createCity( name: "DaNang", zip: "550000") {
    id
    name
    zip
  }
}
```



```
{
"data":{
   "createCity":{
      "id":"1",
      "name":"DaNang",
      "zip":"550000"
   }
}
}
```

How to use GraphQL in iOS

- Two important dependencies is used to make sure that we can integrate to call GraphQL endpoints in iOS
  - Apollo iOS
    - Apollo iOS is a strongly-typed, caching GraphQL client for iOS, written in Swift.
    - It allows you to execute queries and mutations against a GraphQL server, and returns results as query-specific Swift types
    - References Link: https://github.com/apollographql/apollo-ios

How to use GraphQL in iOS

- Two important dependencies is used to make sure that we can integrate to call GraphQL endpoints in iOS
  - Apollo-codegen
    - apollo-codegen will search for GraphQL code in the Xcode project and generate the Swift types
    - Install apollo-codegen: *npm install -g apollo-codegen*

How to use GraphQL in iOS

- Why do we need apollo-codegen dependency?
  - Using apollo-codegen to download Schema file
    - *apollo-codegen download-schema _ENDPOINT_ --output schema.json*
  - Creating a Run Script Phase in Build Phases
    - Showing apollo-codegen configuration in XCode

Showing an example about GraphQL in iOS

The common problems and how to deal when working with GraphQL

- How to read schema file easily ?
  - Using **graphqlviz**
    - References Link: https://github.com/sheerun/graphqlviz
    - Showing a result when use **graphqlviz** to schema file.
- How to generate API.swift files from corresponding .graphql files?

The common problems and how to deal when working with GraphQL

- How to deal with Long type in iOS?
  - Apple is going to support Long type, so that we could not cast this one in Codable. What we need to do is custom own Long type to catch data.

The common problems and how to deal when working with GraphQL

- How to get ***statusCode*** from the response in iOS?
  - Cast the error up to ***GraphQLHTTPResponseError***
- How to add header for a request in iOS?
  - Create an ***URLSessionConfiguration*** instance
  - Make an ***ApolloClient*** instance with ***HTTPNetworkTransport***

The common problems and how to deal when working with GraphQL

- Take note about Catch type when working in Codable (Sub: Sometime the error happens with Int, sometime with String, etc)
- Using “if let” to check the error content before using this one to check result.
- How to remove all unnecessary keys?

Q&A

Thanks for your attention.

If you have any questions or concerns, feel free to ask me