nimble

Explore: Kickstarter Repository

Tobias

Growth Session #25 - XXV - June 20-21 2019

Introduction - What is Kickstarter?

KICKSTARTER

- An American company, that maintains a **crowdfunding** platform
- Their mission: “Help bring creative projects to life”
- Got a good idea but no funds? Upload your idea on Kickstarter

Introduction - Why explore Kickstarter repository?

KICKSTARTER

1. They have an open source Android app
2. It allows me to see how they built their application
3. It allows me to **discover** their implementations ideas

Comparison → hello-tablet VS kickstarter

Similarities:

- ViewModel
- RxJava
- Dagger
- ...

Differences:

- GraphQL
- LeakCanary
- Makefile
- ...

GraphQL

1. A **query language** for your API

   → Write queries using an object structure instead of a text string

2. It gives us the power to request what we need **exactly** and nothing more

   → Instead of receiving the full response, only get the fields that we asked for

3. Get many resources, in a **single** request

   → Instead of calling multiple requests, send only one that will deliver the requested resources

Apollo + GraphQL

APOLLO + GraphQL

1. A GraphQL client that generates **Java** models from your GraphQL queries

   → These models will allow us to make GraphQL requests

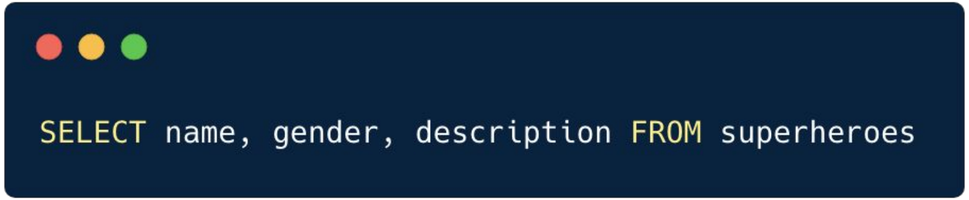
2. You can use these **generated** models to make requests to your API

   → Apollo includes a *client*, that also allows you to configure networking options

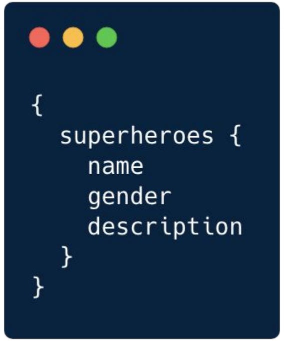
3. It will keep your query statements organized and **easy** to access from Java

Why use GraphQL? - A query language for your API

Write queries using an object structure instead of a text string



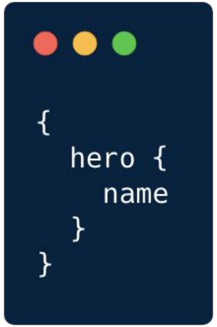
```sql
SELECT name, gender, description FROM superheroes
```



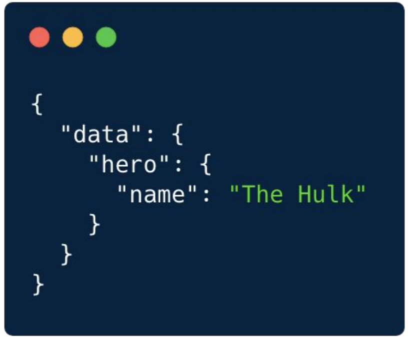
```graphql
{
  superheroes {
    name
    gender
    description
  }
}
```

Why use GraphQL? - Ask for what you need, get exactly that

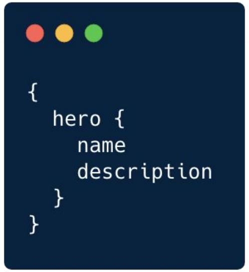
Instead of receiving the full response, only get the fields that we asked for



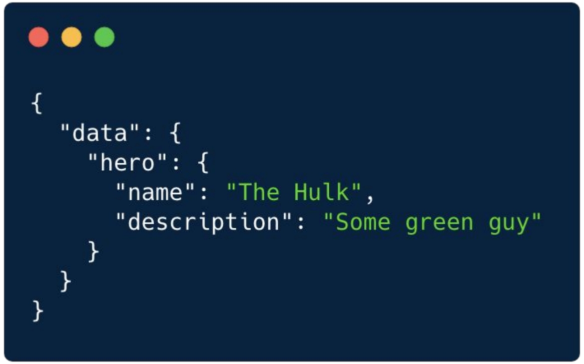
```
{
  hero {
    name
  }
}
```



```
{
  "data": {
    "hero": {
      "name": "The Hulk"
    }
  }
}
```



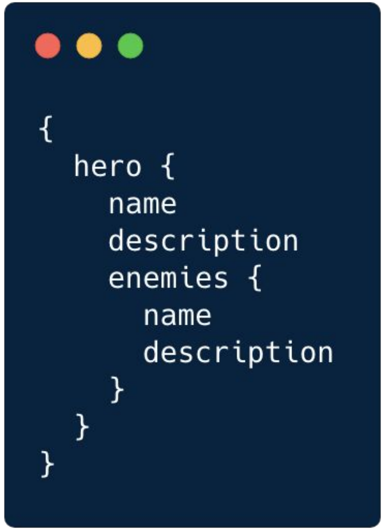
```
{
  hero {
    name
    description
  }
}
```



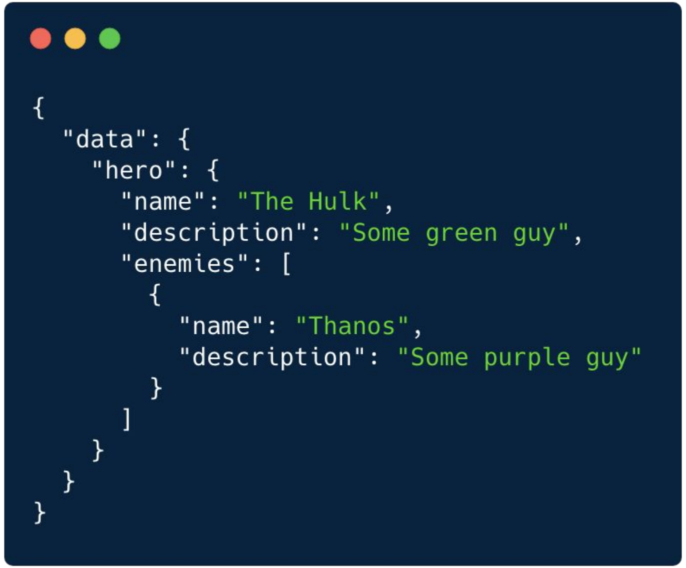
```
{
  "data": {
    "hero": {
      "name": "The Hulk",
      "description": "Some green guy"
    }
  }
}
```

Why use GraphQL? - Get many resources in a single request

Instead of calling multiple requests, send only one that will deliver the requested resources



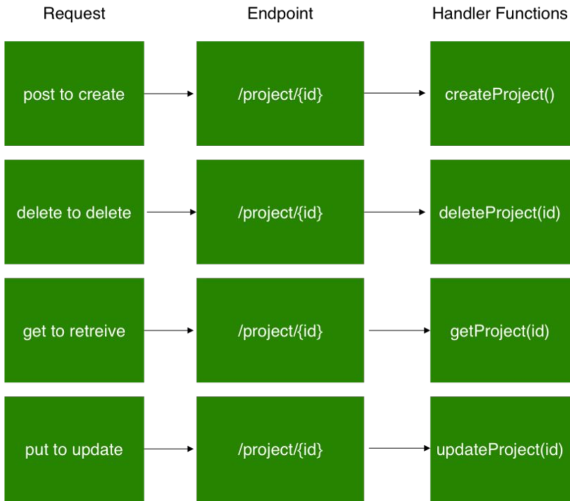
```
{
  hero {
    name
    description
    enemies {
      name
      description
    }
  }
}
```



```
{
  "data": {
    "hero": {
      "name": "The Hulk",
      "description": "Some green guy",
      "enemies": [
        {
          "name": "Thanos",
          "description": "Some purple guy"
        }
      ]
    }
  }
}
```

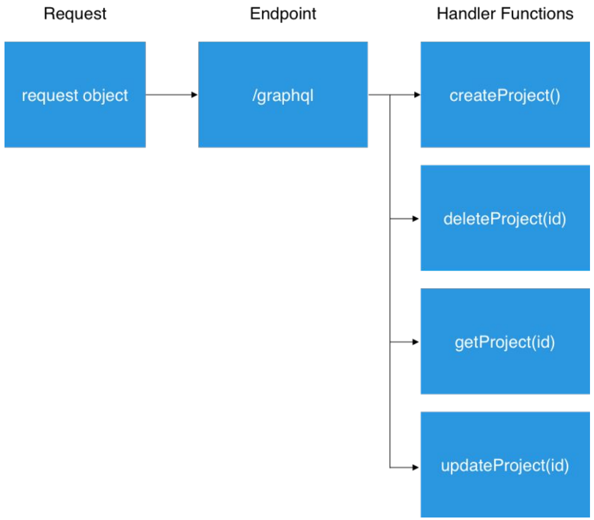
How does GraphQL work?

REST API EXAMPLE



Request	Endpoint	Handler Functions
post to create	/project/{id}	createProject()
delete to delete	/project/{id}	deleteProject(id)
get to retreive	/project/{id}	getProject(id)
put to update	/project/{id}	updateProject(id)

GRAPHQL API EXAMPLE



Request

Endpoint

Handler Functions

request object

/graphql

createProject()

deleteProject(id)

getProject(id)

updateProject(id)

- Instead of having **multiple** endpoints, have a **single** endpoint
- It stays **between** the client and the server

  → Receives client requests and fetches the necessary data

LeakCanary

- A **library** for detecting **memory leaks**

  → Memory leak = Memory allocated to an object which can’t be reclaimed

  → Consequences? Slowing down and/or crashing your application

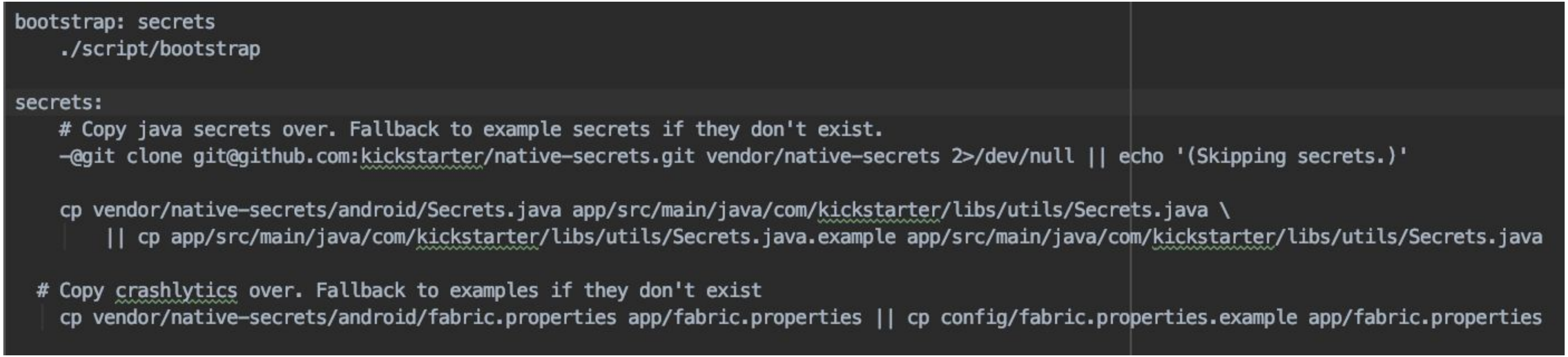
1. Watch destroyed instances using **weak references**

   → Weak reference = A reference not strong enough to keep the object in memory

2. Clear the weak references after waiting for 5 seconds & run the GC

   → If any of the watched instances are still “alive”, these could be potential leaks

MakeFile



```
bootstrap: secrets
    ./script/bootstrap

secrets:
    # Copy java secrets over. Fallback to example secrets if they don't exist.
    -@git clone git@github.com:kickstarter/native-secrets.git vendor/native-secrets 2>/dev/null || echo '(Skipping secrets.)'

    cp vendor/native-secrets/android/Secrets.java app/src/main/java/com/kickstarter/libs/utils/Secrets.java \
        || cp app/src/main/java/com/kickstarter/libs/utils/Secrets.java.example app/src/main/java/com/kickstarter/libs/utils/Secrets.java

  # Copy crashlytics over. Fallback to examples if they don't exist
    cp vendor/native-secrets/android/fabric.properties app/fabric.properties || cp config/fabric.properties.example app/fabric.properties
```

A **file** containing a set of **bash** instructions, some examples:

- Instructions to fetch application secrets & properties
- Update the new employee’s IDE with the company’s styling preferences

Thanks!

Contact Nimble

nimblehq.co
hello@nimblehq.co

Bangkok

399 Interchange 21 Sukhumvit Road, Unit
#2402-03, Klong Toei, Wattana, Bangkok
10110, Thailand

Singapore

28C Stanley St, Singapore 068737

Hong Kong

20th Floor, Central Tower
28 Queen's Road, Central, Hong Kong

nimble