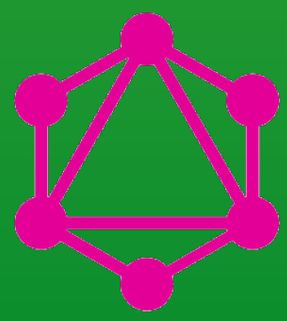
JESUÏTES
educació

DAWM07UF4
SERVEIS WEB

UF4.7. GRAPHQL

CFGS Desenvolupament d'Aplicacions Web
M06. Desenvolupament web en entorn servidor
Fundació Jesuïtes Educació - Escola del Clot
Sergi Grau sergi.grau@fje.edu



GraphQL

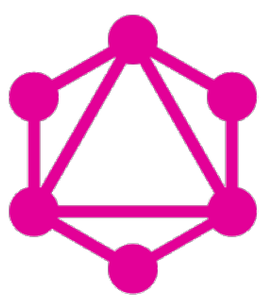
OBJECTIUS

- Coneixer que és el protocol GraphQL
- Crear esquemes de GraphQL
- Desenvolupar clients i servidors en JS que en facin us.

GRAPHQL

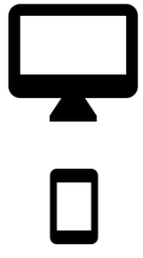
- GraphQL és una de les alternatives que han sorgit per solucionar la major part dels problemes dels serveis web REST.

- El primer esborrany del RFC (encara en desenvolupament) que especifica GraphQL va ser creat per Facebook . Tal com ells mateix expliquen porten usant-ho i perfeccionant des de 2012 en les seves apps mòbils. I no són els únics, també Github,Pinterest o Shopify s'han unit al carro. Tots els interessats en implementar GraphQL poden col·laborar en la seva definició, és Open Source. https://graphql.org/



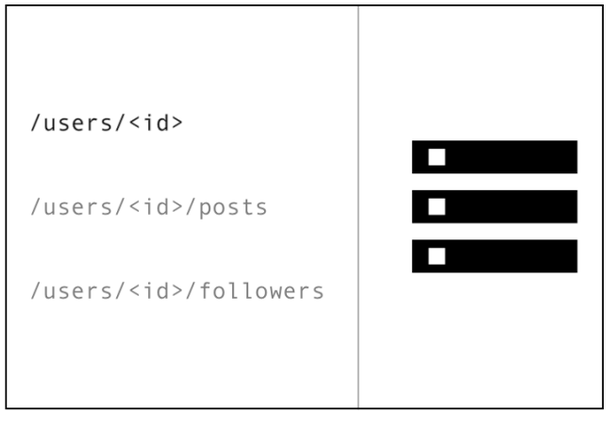
GraphQL

1

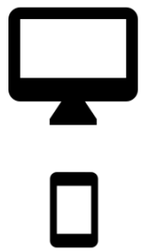


HTTP GET

```
{
  “user”: {
    “id”: “er3tg439frjw”
    “name”: “Mary”,
    “address”: { … },
    “birthday”: “July 26, 1982”
  }
}
```

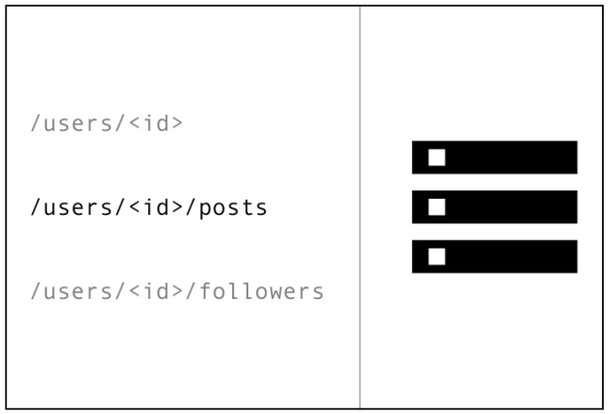


2

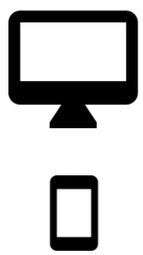


HTTP GET

```
{
  “posts”: [{
    “id”: “ncwon3ce89hs”
    “title”: “Learn GraphQL today”,
    “content”: “Lorem ipsum … ”,
    “comments”: [ … ],
  }]
}
```

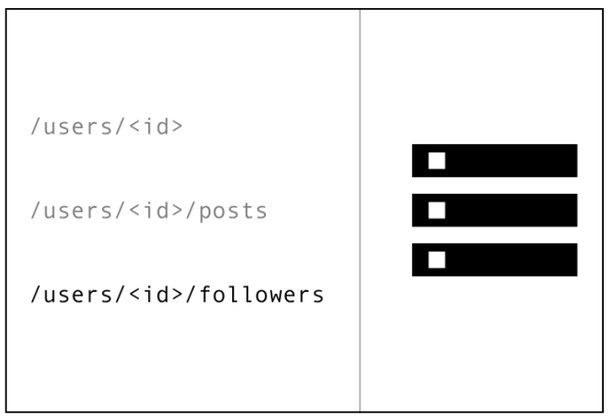


3



```
{
  “followers”: [{
    “id”: “leo83h2dojsu”
    “name”: “John”,
    “address”: { … },
    “birthday”: “July 26, 1982”
  },
  …]
}
```

HTTP GET



PRINCIPALS DEBILITATS DE RESTFUL

- En RESTful utilitzem una URI per llegir o escriure un únic recurs, ja sigui, per exemple, un producte, una persona, un post o un comentari. Si necessitem treballar amb múltiples recursos com una llista de posts amb un llistat de comentaris, necessitem utilitzar diversos endpoint i encadenar diferents trucades, de vegades de forma seqüencial. **És en les apps on recau la responsabilitat d'unir i mergear cada recurs**.

- **Quan fem una petició, no rebem simplement la informació que necessitem, sinó tot el conjunt de dades relatives al recurs** allotjat en aquesta URI.Una cosa bastant complicada de gestionar i costós en recursos quan el 90% de les dades procedents de cada resource / endpoint són innecessaris.

PRINCIPALS DEBILITATS DE RESTFUL

- **El versionat d'una API REST no és trivial**. En moltes ocasions necessitem afegir nous camps o modificar el tipus d'algun d'ells però per poder donar suport de retrocompatibilitat els incloem en el payload que els clients vells, la qual cosa fa créixer el payload de resposta innecessàriament.

- Tots coneixem els verbs HTTP involucrats en les peticions RESTful però això **no assegura la seva correcta utilització**. Ni molt menys de les seves codis d'error basats en HTTP.

- Per descomptat, les respostes d'error amb el payload que reben els clients no es queden enrere en la incoherència de moltes APIs REST que consumim. No hi ha una especificació que el fixi.

COM FUNCIONA?

- GraphQL és un protocol agnòstic i **no depèn en res d'HTTP. No fa servir mètodes o respostes HTTP**. Per descomptat, segueix sent el canal més popular per comunicar-se entre consumidors de GraphQL.

- Unes de les principals característiques de GraphQL és que el llenguatge i sintaxi usat en la request és el mateix que el de la resposta. Analitzant el JSON podem comprendre clarament el diccionari de key-value. Simplificant el seu ús podríem dir que és un llenguatge pregunta / resposta expressada en relacions entre els objectes exposats. A més GraphQL disposa d'un sistema de tipat fort pel que qualsevol mal ús pot ser ràpidament detectat en temps de desenvolupament (fins i tot des del IDE) en contraposició a un mapatge clàssic de JSON.

COM FUNCIONA?

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

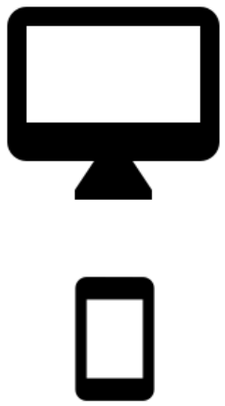
```
type Query {
  hero: Character
}

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

type Planet {
  name: String
  climate: String
}

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

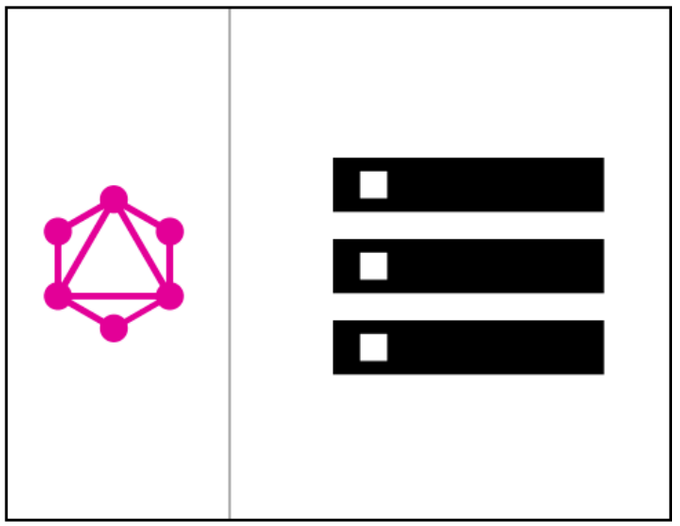
COM FUNCIONA?



```
query {
  User(id: “er3tg439frjw”) {
    name
    posts {
      title
    }
    followers(last: 3) {
      name
    }
  }
}
```

HTTP POST

```
{
  “data”: {
    “User”: {
      “name”: “Mary”,
      “posts”: [
        { title: “Learn GraphQL today” }
      ],
      “followers”: [
        { name: “John” },
        { name: “Alice” },
        { name: “Sarah” },
      ]
    }
  }
}
```



COM FUNCIONA?

- Un altre dels seus principals trets és que l'API pot ser **single endpoint**. És a dir, un únic endpoint pot gestionar sense problemes les peticions dels clients. Totes elles pot preguntar sobre múltiples recursos i camps, tan sols indicant què necessiten. La resposta s'ha d'ajustar a aquesta demanda d'informació, ni més ni menys.

- El disseny GraphQL permet gestionar jerarquies d'objectes per compondre les vistes que necessitem d'ells, segons els requisits demanats pel client.

INTRODUCCIÓ

- GraphQL és un llenguatge de consulta per a l'API i un runtime del servidor per executar consultes mitjançant un sistema de tipus que defineix les dades.

- GraphQL no està vinculat a cap base de dades específica ni a un motor d'emmagatzematge.

- Juntament amb funcions per a cada camp de cada tipus

```
type Query {
  me: User
}

type User {
  id: ID
  name: String
}

function Query_me(request) {
  return request.auth.user;
}

function User_name(user) {
  return user.getName();
}
```

INTRODUCCIÓ

- Una vegada que el servei GraphQL s'estigui executant (normalment en un URL d'un servei web), es poden enviar consultes de GraphQL per validar i executar.

- Es verifica primer una **consulta** rebuda per garantir que només es refereix als tipus i camps definits, i executa les funcions proporcionades per produir un **resultat**.

```
{
  me {
    name
  }
}
```

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

ELEMENTS CORE I ARQUITECTURA

https://www.howtographql.com/basics/2-core-concepts/

https://www.howtographql.com/basics/3-big-picture/

BIBLIOGRAFIA

+ https://graphql.org/

+ https://www.howtographql.com/