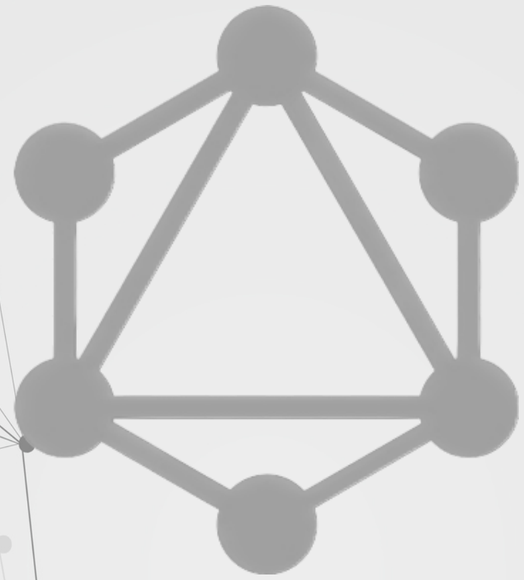
Septembre 2023



GraphQL

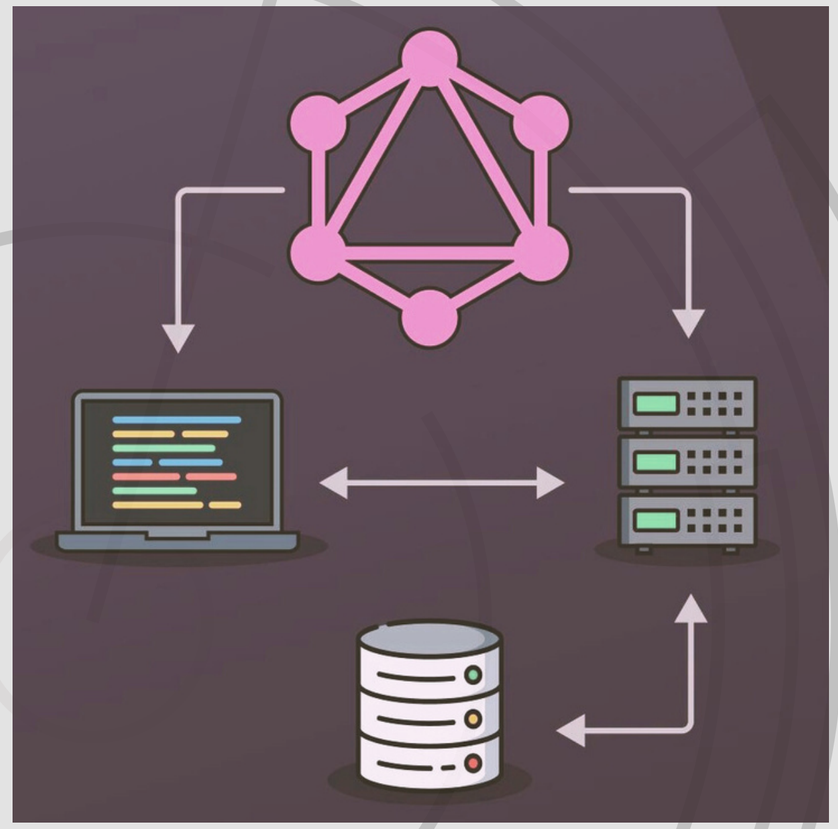
Carla Granados - Jiahao Liu - Rubén Dabrio - Paula Muñoz

ASW G13

Introducció

Que és?

- Llenguatge de consulta
- Temps d'execució del servidor per a APIs.
- Proporcionar als clients les dades que sol·liciten.
- Ràpides, flexibles i senzilles pels desenvolupadors.
- Implementar en entorn de desenvolupament integrat.
- Flexibilitat als responsables del manteniment de les APIs.
- Compatible amb tots els mètodes de diseny d’interficies



Evolució històrica

1

2010

Smartphones, Facebook comencen a utilitzar aquests navegadores utilitzant API rest per crear una app nativa per iOS.

2

2011

API RESTful comença a donar problemes

3

2012

Facebook começa a desenvolupar GraphQL

4

2015

Llençat GraphQL com a open source.

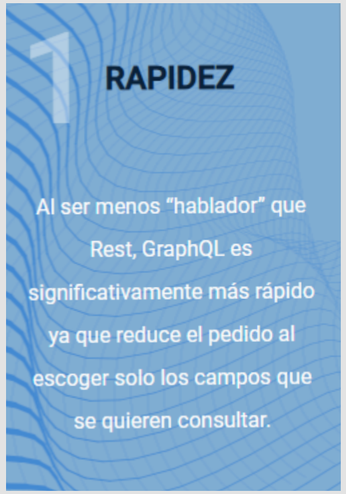
5

2018

GraphQL es va traslladar a la nova fundació GraphQL Foundation, allotjada per la non-profit Linux Foundation.

Característiques principals

- Permet al client especificar exactament quines dades necessita.
- Facilita l'agregació de dades de fonts múltiples.
- Utilitza un sistema de tipus per descriure les dades.



3 operacions primàries:

- Consulta per llegir dades.
- Mutació per escriure dades.
- Subscripció per rebre automàticament dades en temps real al llarg del temps.

Pilars Fonamentals

- De vegades només es necessita un únic camp d'un recurs, però el servidor torna tots, però, amb GraphQL és l'aplicació la que té el control del recurs, reduint l'amplada de banda i els temps.
- Amb GraphQL es pot accedir a diferents recursos en una única trucada.
- S'elimina el concepte d'endpoints, substituint-lo per tipus per indicar només el que es necessita.
- S'intenta eliminar el versionat de les API podent demanar l'ús d'atributs.
- GraphQL desacobla la dependència amb l'accés a les bases de dades.

Exemple funcional/ estructura

OBJECTES

```
type User {
  id: ID! # The "!" means required
  firstname: String
  lastname: String
  email: String
  username: String
  todos: [Todo] # Todo is another GraphQL type
}
```

CONSULTES

```
type RootQuery {
  user(id: ID): User           # Corresponds to GET /api/users/:id
  users: [User]                # Corresponds to GET /api/users
  todo(id: ID!): Todo    # Corresponds to GET /api/todos/:id
  todos: [Todo]          # Corresponds to GET /api/todos
}
```

MUTACIONS

```
type RootMutation {
  createUser(input: UserInput!): User             # Corresponds to POST /api/users
  updateUser(id: ID!, input: UserInput!): User    # Corresponds to PATCH /api/users
  removeUser(id: ID!): User                       # Corresponds to DELETE /api/users
  createTodo(input: TodoInput!): Todo
  updateTodo(id: ID!, input: TodoInput!): Todo
  removeTodo(id: ID!): Todo
}
```

Exemple funcional/ estructura

RESOLVERS

```
import sequelize from '../models';
export default function resolvers () {
  const models = sequelize.models;
  return {
    // Resolvers for Queries
    RootQuery: {
      user (root, { id }, context) {
        return models.User.findById(id, context);
      },
      users (root, args, context) {
        return models.User.findAll({}, context);
      }
    },
    User: {
      todos (user) {
        return user.getTodos();
      }
    },
  }
  // Resolvers for Mutations
  RootMutation: {
    createUser (root, { input }, context) {
      return models.User.create(input, context);
    },
    updateUser (root, { id, input }, context) {
      return models.User.update(input, { ...context, where: { id } });
    },
    removeUser (root, { id }, context) {
      return models.User.destroy(input, { ...context, where: { id } });
    },
    // ... Resolvers for Todos go here
  }
}
```

ESQUEMA

```
schema {
  query: RootQuery
  mutation: RootMutation
}
```

Punts forts

Consultes flexibles

GraphQL permet als clients sol·licitar dades específiques, cosa que redueix la sobrecàrrega i la infracàrrega. Aquesta flexibilitat pot millorar el rendiment en minimitzar la quantitat de dades transferides entre el client i el servidor.

Únic endpoint

A diferència de les API REST, que requereixen múltiples endpoints, GraphQL gestiona totes les operacions mitjançant un sol punt de sol·licitud i resposta. Això simplifica el desenvolupament del costat del servidor i permet un versionat i desplegament més manejables.

Tipat fort

El sistema de tipus integrat de GraphQL ajuda a garantir respostes coherents del servidor i facilita als desenvolupadors la comprensió de les dades amb què estan treballant.

Dades en temps real

Les subscripcions a GraphQL permeten actualitzar les dades en temps real, cosa que pot ser crucial per a les aplicacions modernes i dinàmiques que depenen d'informació actualitzada.

Punts febles

Complexitat

GraphQL té una corba d'aprenentatge més pronunciada que les API REST, fet que fa que sigui més difícil d'adoptar per als desenvolupadors, especialment aquells sense experiència prèvia amb la tecnologia.

No té emmagatzematge nadiu a la memòria

GraphQL no té suport nadiu per a l'emmagatzematge en memòria cau, per la qual cosa cal implementar estratègies d'emmagatzematge en memòria cau personalitzades per optimitzar el rendiment. Això pot augmentar la complexitat del desenvolupament i el manteniment.

Menys suport per al maneig de fitxers

El maneig de fitxers, com la càrrega o descàrrega de fitxers de gran mida, no és tan senzill a GraphQL com a les API REST, el que requereix solucions o biblioteques addicionals.

Ecosistema menys madur

Tot i que el seu ecosistema està creixent ràpidament, GraphQL segueix sent una tecnologia relativament nova en comparació amb REST, i és possible que no sempre hi hagi eines i biblioteques conformes disponibles o tan madures com les de les API REST.

Conclusions

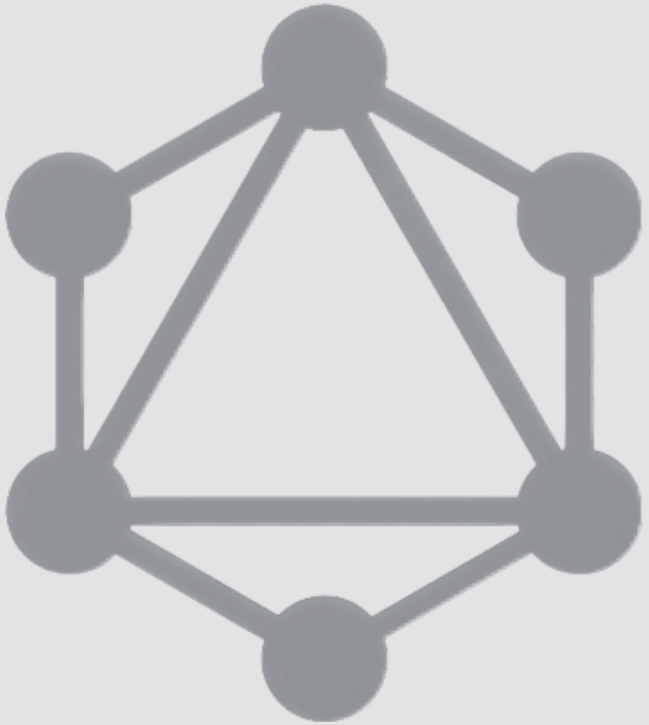
- És una tecnologia potent que ofereix moltes avantatges per a la gestió de les APIs i la comunicació client-servidor.

No obstant

- No és la millor opció per a tots els casos, i la seva adopció depèn de les necessitats específiques del projecte i de la familiaritat dels desenvolupadors amb la tecnologia.

Però

- Pot ser una elecció excel·lent per a millorar la flexibilitat i l'eficiència de les aplicacions.



Referencies

https://graphql.org/

https://www.redhat.com/es/topics/api/what-is-graphql

https://kinsta.com/es/blog/graphql-vs-rest/

https://en.wikipedia.org/wiki/GraphQL

https://medium.com/beltranc/qu%C3%A9-es-graphql-ed6262cbfa8

https://rootstack.com/es/technology/graphql-tecnologia

https://www.adictosaltrabajo.com/2017/06/01/introduccion-a-graphql/

https://appmaster.io/es/blog/graphql-vs-rest-guia-definitiva-para-elegir-el-mejor-enfoque-api#ventajas-y-desventajas-de-las-api-graphql

https://appmaster.io/es/blog/graphql-vs-rest-guia-definitiva-para-elegir-el-mejor-enfoque-api#ventajas-y-desventajas-de-las-api-graphql

Repartició de tasques

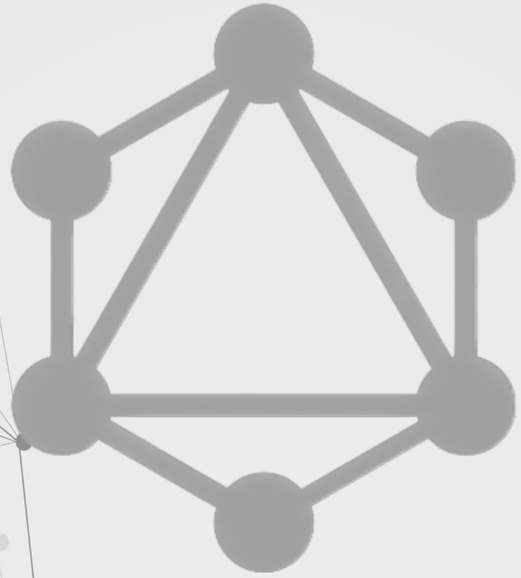
Recopilació d'Informació: Carla Granados i Rubén Dabrio

Preparar Presentació i diapositives: Paula Muñoz

Revisió de la Presentació i diapositives: Carla Granados, Jiahao Liu i Rubén Dabrio

Presentació Oral: Carla Granados i Paula Muñoz

Septembre 2023



GraphQL

Carla Granados - Jiahao Liu - Rubén Dabrio - Paula Muñoz

ASW G13