

환영합니다!

- WiFi: mz_cloud_edu#3 / mzclouD!@#
- 강의실 뒤편에 김밥이 준비되어있습니다. 자유롭게 가져가시면 됩니다.
- 물은 강의실 바깥의 정수기를 이용해주세요.
- 세션 자료 Repository: <https://github.com/cadenzah/ausg-2020-handson-appsync>

Amazon AppSync로 만드는 서비스 GraphQL 서비스

React.js, GraphQL, AWS Amplify, AWS AppSync

AUSG 3기 김정모

발표자를 소개합니다.



김정모 AUSG 3기

- 한양대학교 정보시스템학과 재학
- 주로 Web 개발 → 다방면에 관심
- React, Node.js
- <https://github.com/cadenzah>

오늘 함께 할 내용은?

1부: GraphQL

- GraphQL 소개
- 왜 이걸 만들었을까?
- REST vs. GraphQL
- GraphQL의 특징
- GraphQL 주요 개념

2부: AWS 기초

- AWS란
- On-Premise와 Cloud
- 관리형 서비스
- 사용할 AWS 서비스 알아보기

3부: Hands-On

- 환경 설정
- Code
- Deploy
- Wrap-up

GraphQL

GraphQL



GraphQL 소개

GraphQL 소개

GraphQL?

GraphQL is a **query language for APIs** and a runtime for fulfilling those queries with your existing data.

(출처: [GraphQL 공식 홈페이지](#))

GraphQL 소개

2012년, Facebook에서 개발

→ Facebook에서 내부적으로 사용

2015년, 오픈소스로 대중에 공개 (@JSconf)

→ [React.js Conf 2015 – Data fetching for React applications at Facebook](#)

GraphQL

←
왜 이걸 만들었을까?

왜 이걸 만들었을까?

TL;DR – 기존 웹 개발의 문제점

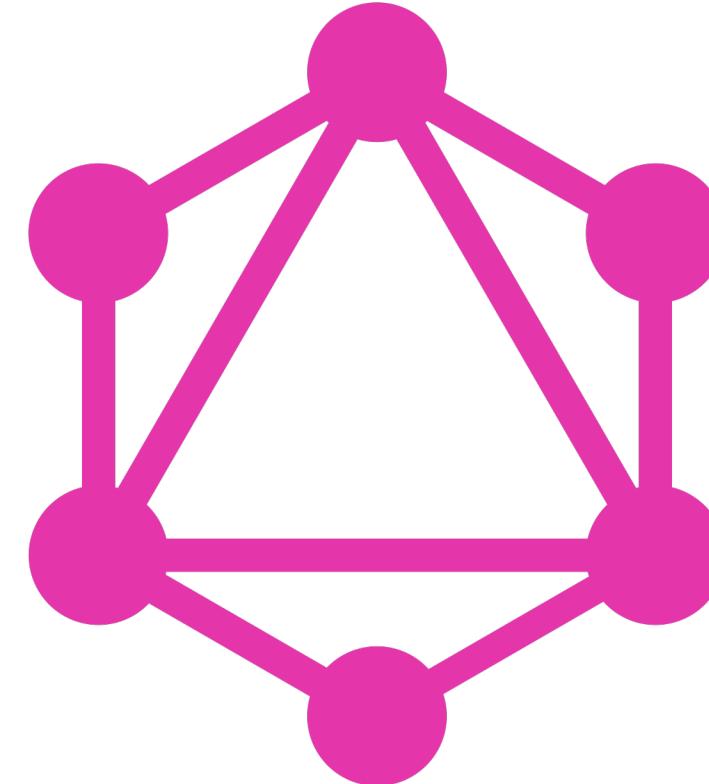
- UI에 맞추어 데이터를 보내주어야 함
- Frontend 변경시, Server도 변경 필요
 - Frontend와 Server 간의 종속성 심화
 - 앱의 규모가 거대해지면, **유지보수 복잡도** 상승

왜 이걸 만들었을까?

TL;DR – 해결을 위한 접근 방식

1. 클라이언트에서 필요한 데이터 형식을 서버가 아닌 **클라이언트 측**에서 정의 할 수 있고, 이에 대하여 서버는 유연하게 대응할 수 있다.
2. 클라이언트에서 필요한 데이터 요청을 모아서 서버에 한번 요청하여 한번에 가져온다.

왜 이걸 만들었을까?



(짠)

GraphQL



REST vs. GraphQL

REST vs. GraphQL

상황: 블로그 플랫폼

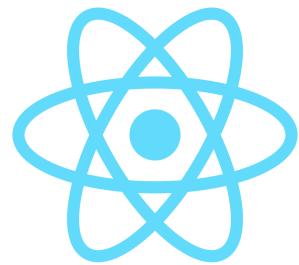
블로그 주인 정보

최근 블로그 게시글 리스트

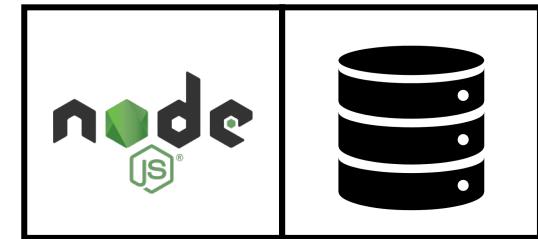
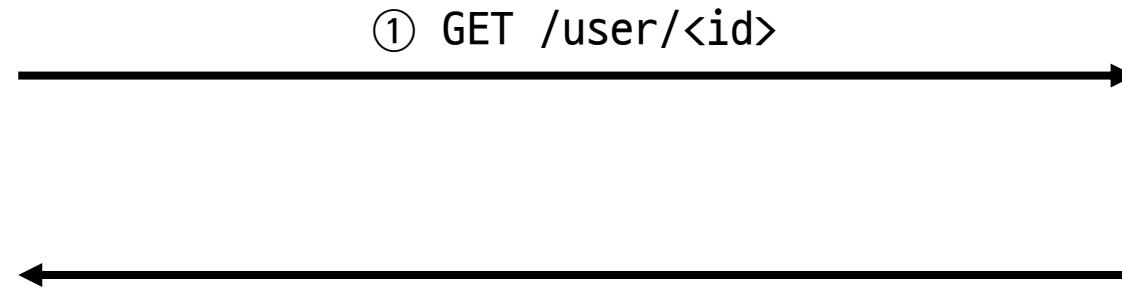
블로그 팔로워

REST vs. GraphQL

상황: 블로그 플랫폼 – REST API



Client

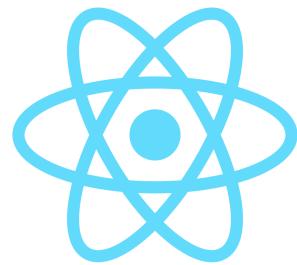


Server

```
{  
  "user": {  
    "id": "cadenzah",  
    "name": "김정모",  
    "email": "cadenzah93@gmail.com",  
  }  
}
```

REST vs. GraphQL

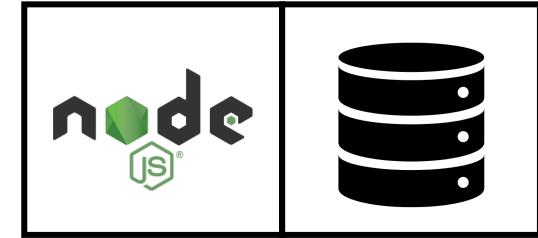
상황: 블로그 플랫폼 – REST API



Client

② GET /user/<id>/posts

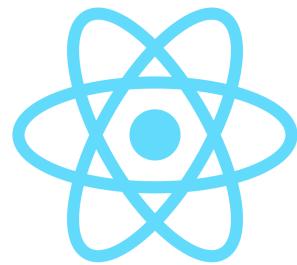
```
{  
  "posts": [  
    {  
      "id": 123,  
      "title": "한반도 최고의 평냉을 찾아서",  
      "date": "2019-12-23",  
      "content": "Lorem ipsum...",  
    },  
    ...  
  ]  
}
```



Server

REST vs. GraphQL

상황: 블로그 플랫폼 – REST API

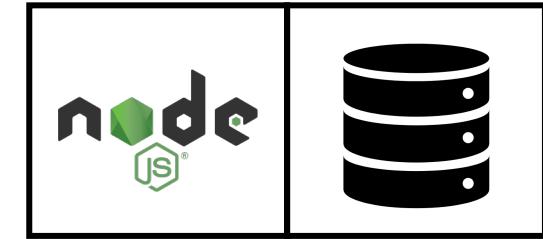


Client

③ GET /user/<id>/followers



```
{  
  "followers": [  
    {  
      "id": "john_doe",  
      "name": "John Doe",  
      "email": "john_doe@gmail.com",  
    },  
    ...  
  ]  
}
```



Server

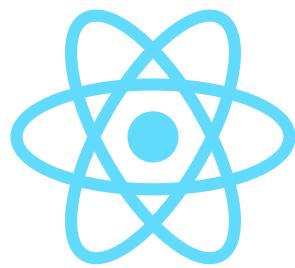
REST vs. GraphQL

상황: 블로그 플랫폼 – REST API

1. 하나의 화면을 위하여 요청이 3번 필요
2. 필요 이상의 데이터를 불러오기 (Overfetch)
3. 유연하지 않은 API 엔드포인트

REST vs. GraphQL

상황: 블로그 플랫폼 – GraphQL

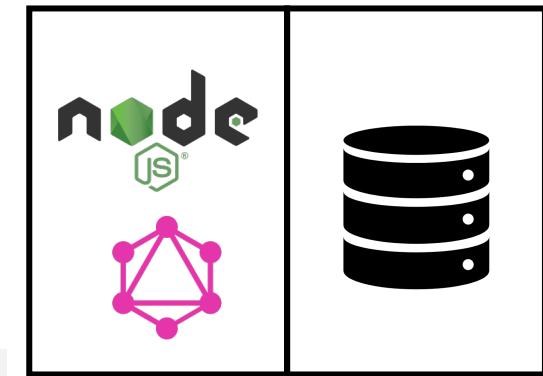


Client

```
query {
  User("cadenzah") {
    name
    posts {
      title
    }
    followers {
      name
    }
  }
}
```

POST /graphql

```
{
  "User": {
    "name": "김정모",
    "posts": [{ "title": "한반도 최고의 평냉을 찾아서" }],
    "followers": [
      { "name": "John Doe" },
      ...
    ]
  }
}
```



Server

GraphQL



GraphQL의 특징

GraphQL의 특징

1. 필요한 데이터를 정확히 명시하고, 필요한 데이터만 요청
2. 한번의 요청으로 모든 필요 데이터를 가져오기
3. 타입 시스템 – 안정성과 견고함
4. 기본 제공되는 API 명세 및 문서화 도구 (GraphiQL / GraphQL Playground)
5. API의 점진적인 향상

GraphQL



GraphQL 주요 개념

GraphQL 주요 개념

Schema

GraphQL을 통하여 **Client**와 **Server** 간에 데이터를 주고 받기 위한 약속

GraphQL 주요 개념

Type

GraphQL에서 주고 받을 데이터의 형식

Field로 구성된다

```
type Person {  
    name: String!  
    age: Int!  
}
```

!는 값이 null일 수 없음을 의미

GraphQL 주요 개념

Schema 내의 기본 타입

- Int : 부호가 있는 정수
- Float : 부호가 있는 부동소수점수
- String : UTF-8 문자열
- Boolean : true / false
- ID : 고유 식별값을 뜻하는 UTF-8 문자열

GraphQL 주요 개념

그 외의 Schema 내의 타입

- Object : 다른 타입으로 구성되는 타입 ({ })
- List : 배열 형태 ([])
- Enum : 특정 값들로 제한

기타 생략

GraphQL 주요 개념

Query

데이터를 읽어오기

```
query {  
  Root Field getPersons {  
    name  
    age } ] Payload  
  }  
}
```



```
{  
  "getPersons": [  
    {  
      "name": "John Doe",  
      "age": 24,  
    },  
    {  
      "name": "Mary Sue",  
      "age": 27,  
    },  
    ...  
  ]  
}
```

GraphQL 주요 개념

Mutation

데이터를 생성하거나 변경하기

```
mutation {
  createPerson(name: "cadenzah", age: 24) {
    name
    age
  }
}
```



```
{
  "createPerson": {
    "name": "cadenzah",
    "age": 24,
  }
}
```

GraphQL 주요 개념

Schema 정의

Query, Mutation, Type의 정의

```
type Query {  
    getPersons: [Person!]!  
    getCount: Int!  
}  
  
type Person {  
    name: String!  
    age: Int!  
}  
  
type Mutation {  
    createPerson(name: String!, age: Int!): Person!  
    updatePerson(name: String, age: Int): Person!  
    deletePerson(name: String!): Person!  
}
```

GraphQL 주요 개념

Resolver

필드에 대한 데이터를 불러오는 함수

```
query {
  getPersons {
    name
    age
  }
}
```



모든 필드는 Resolver를 가져야 한다

```
// resolvers.js

const getPersons = (parent, args, context) => {
  return context.db.persons.find()
}

const name = (parent, args, _) => {
  return parent.name
}

const age = (parent, args, _) => {
  return parent.age
}
```

AWS 소개



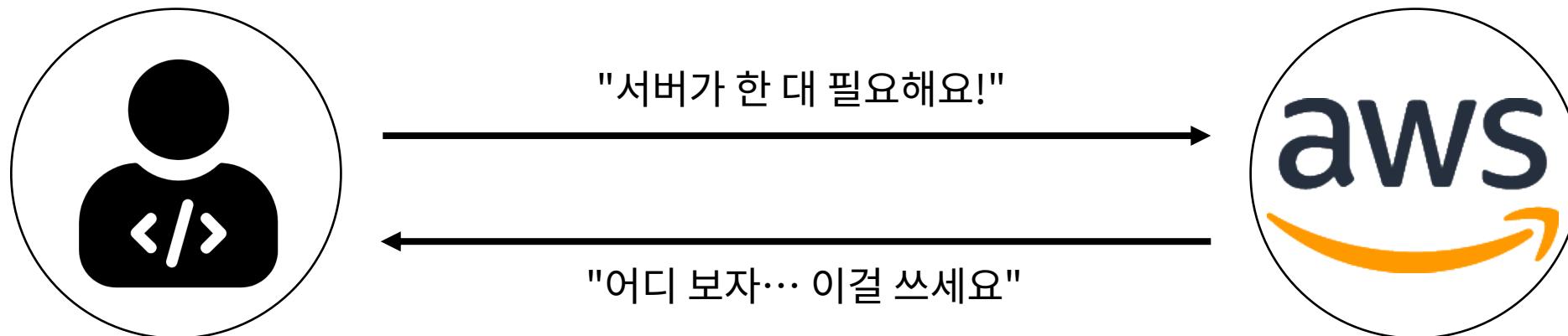
AWS 소개



AWS란

Cloud Computing Resource Platform

- 클라우드 컴퓨팅 자원을 호스팅하는 플랫폼
ex) 개인 PC, 서버용 컴퓨터들, 라즈베리파이, ...
- 컴퓨팅 자원: 서비스를 제공하는 데에 사용되는 컴퓨팅 머신
- 클라우드: 바로 눈앞이 아닌 다른 먼 곳에 위치



AWS란



(출처: <https://aws.amazon.com/ko/compliance/data-center/data-centers/>)

AWS 소개



On-Premise 와 Cloud

On-Premise 와 Cloud

On-Premise

- on the premise : …를 전제로
 - 자체 보유한 전산실에 서버를 구축하여 운영하는 방식
 - 어떤 방식이 더 유리한지 검토 필요
- 비용, 안정성, 보안, 학습 부담, 기존 시스템과의 통합 등

On-Premise 와 Cloud



돈은 중대 사항입니다

AWS 소개

관리형 서비스

관리형 서비스

Managed Service

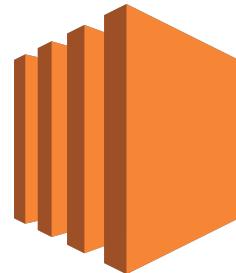
AWS가 제공하는 대부분의 서비스는 모두 관리된다

- AWS가 서비스의 **확장성, 가용성, 내결함성을 관리**

- 특정 사용 용도에 집중한 서비스 제공

ex) Lambda, S3, RDS, … → 만들고자 하는 서비스 구성에 따라 조립

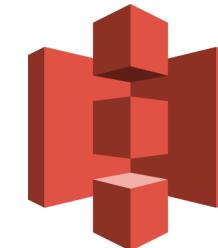
- 개발자는 앱 로직에 집중



Amazon EC2



Lambda



S3



Route 53



API Gateway



DynamoDB

AWS 소개



사용할 AWS 서비스 알아보기

사용할 AWS 서비스 알아보기

AWS IAM

Identity and Access Management

- AWS 자원을 사용하려면 적절한 자격이 필요
- 자원에 따라 각기 자격 관리 가능
- aws-cli 를 통하여 등록 및 사용

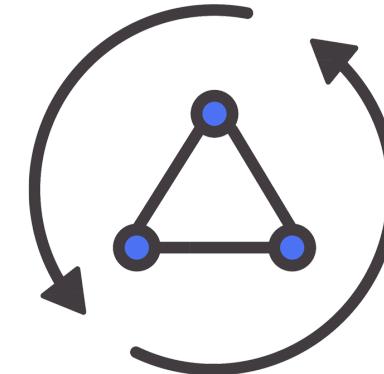


사용할 AWS 서비스 알아보기

AWS AppSync

Managed GraphQL API Service

- Schema만 제공하면 간단하게 GraphQL API를 생성
- 실습에서 사용할 API 생성 및 배포



Amazon DynamoDB

Managed NoSQL Database Service

- AppSync의 Data Source로 사용
- 실습에서 사용할 데이터를 저장



사용할 AWS 서비스 알아보기

AWS Amplify

Fastest Way to Build Mobile and Web Application

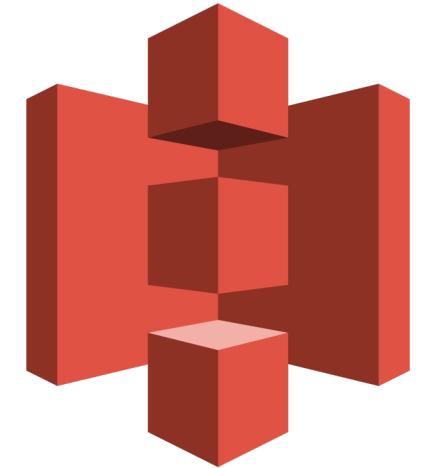
- 간단하게 모바일 및 웹 앱을 배포할 수 있도록 해주는 프레임워크
- 내부적으로 다양한 AWS 서비스들을 사용



Amazon S3

Simple Storage Service

- 정적 데이터를 저장할 수 있는 스토리지
- Amplify, AppSync의 설정 파일 저장



Hands-On

감사합니다

설문조사 참여 부탁드립니다.

