

20' CMD

# 자바스크립트의 이해

컴퓨터교육과 정유찬, 김태현



2020. 07. 24

# 목차



## 1. JSON

JSON의 개념 및 이해, Object란, JSON  
특징 JSON stringify, parse

## 2. API

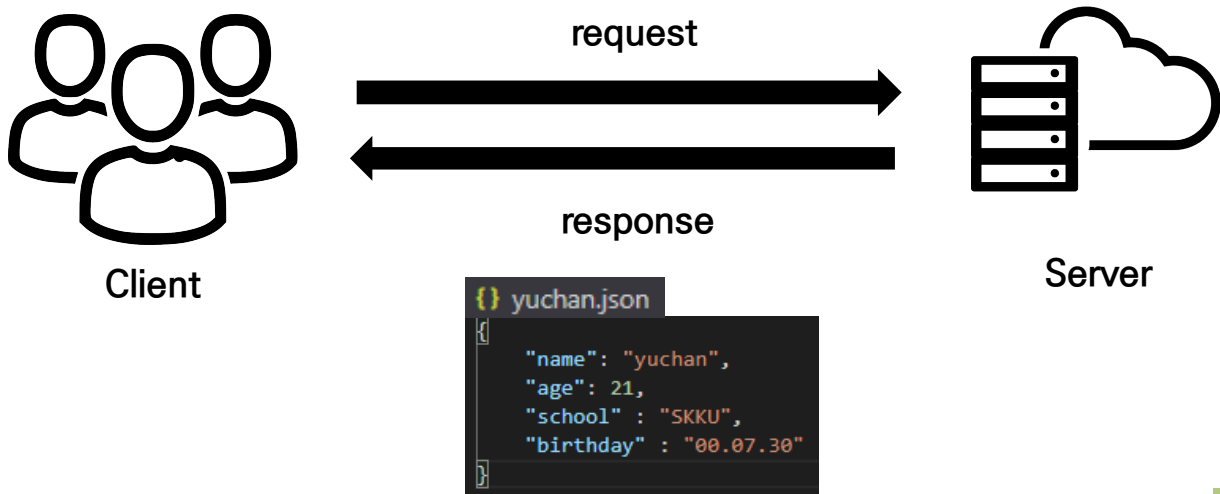
API의 개념, 오픈 API 실습



# 1. JSON

- 개념

- 1 데이터를 표시하는 표현 방법
- 2 서버와 클라이언트 사이에서 데이터를 교환할 때의 형식 또는 양식 “통신 방법? 프로그래밍 문법? 아니다!”



# 1. JSON

- Object (잠깐 짚고 넘어가자!)

- 1 JSON 세부 의미: JavaScript Object Notation = JS에서 object 만들 때 사용하는 표현식
- 2 Object: Class에 데이터를 넣은 것 (instance of a class)

Class 없이도 바로 만들 수 있음

```
class Person{  
  constructor(name, age){  
    this.name = name;  
    this.age = age;  
  }  
}  
  
const yuchan = new Person("yuchan", 21);  
const taehyun = new Person("taehyun", 50);
```

```
const dog = {  
  name = "또리",  
  age = 5,  
  size = 15  
}
```

# 1. JSON

- 특징

- 이름에서 알 수 있듯이 JavaScript의 Object 표현식과 유사함

{ Key : Value }

```
const dog = {  
  name = "또리",  
  age = 5,  
  size = 15  
}
```

```
{ } text.json > ...  
1 {  
2   "name": "yuchan",  
3   "age": 21  
4 }
```

- 프로그래밍 언어, 플랫폼 상관없이 사용가능 (라이브러리 많음)
- 자료형: 수, 문자열, Boolean, 배열, 객체, null

# 1. JSON

- Server-Client 데이터 교환

Example)

Client에서 “토끼” object를 server로 전송할 때, json형식  
( **key, value 형식**)으로 바꾸어 전송 (serialize)

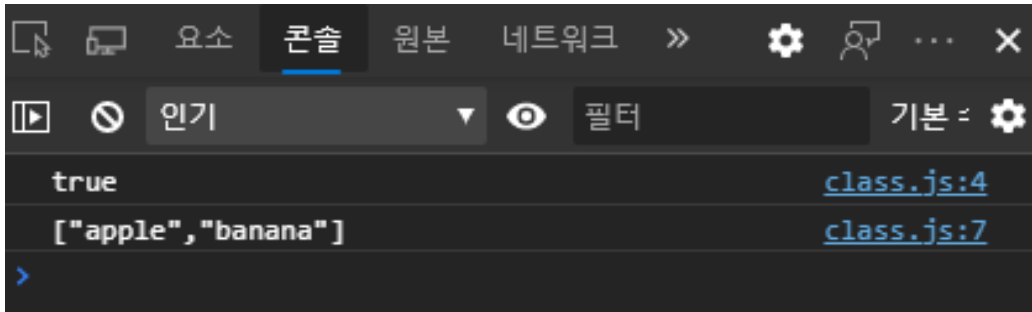
Server에서 다시 받아올 때, 전송 받은 json형식의 데이터를 토끼 object로 변환하여 웹 브라우저에 표시(deserialize)



# 1. JSON

- JSON stringify 실습

```
let json = JSON.stringify(true);  
console.log(json);  
  
json = JSON.stringify(['apple', 'banana']);  
console.log(json);
```



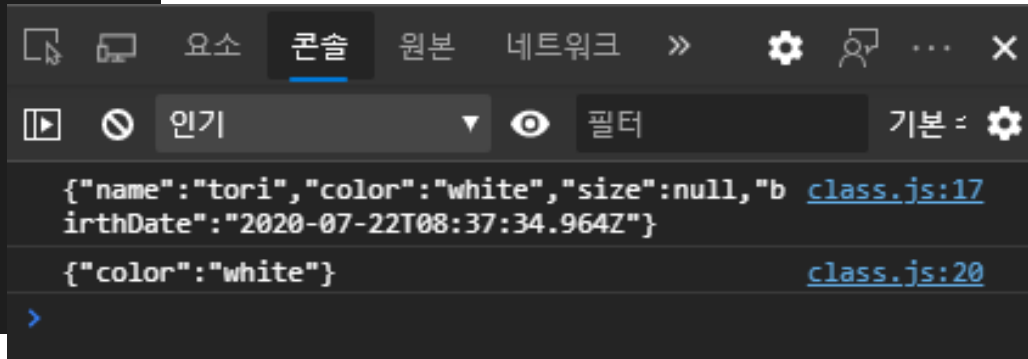
# 1. JSON

- JSON stringify 실습

```
const rabbit = {  
  name : 'tori',  
  color : 'white',  
  size : null,  
  birthDate: new Date(),  
  jump: () => { // 함수는 data 아니기에 json에 포함 안됨.  
    console.log(`${name} can jump!`);  
  },  
};
```

```
json = JSON.stringify(rabbit);  
console.log(json);
```

```
json = JSON.stringify(rabbit, ["color"]);  
console.log(json);
```

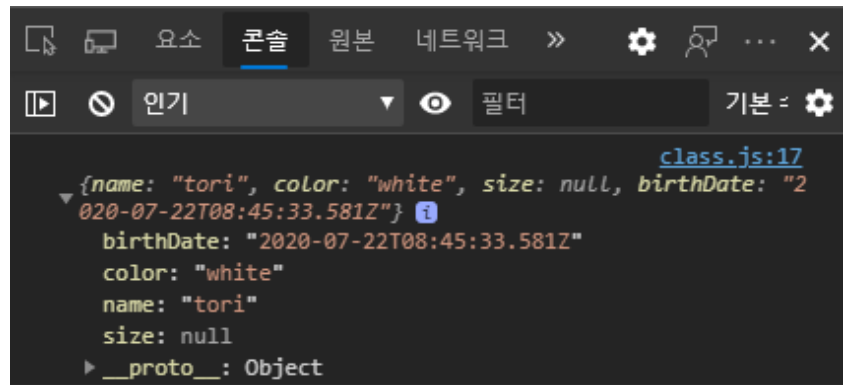




# 1. JSON

- JSON parse 실습

```
const rabbit = {  
  name : 'tori',  
  color : 'white',  
  size : null,  
  birthDate: new Date(),  
  jump: () => { // 함수는 data 아니기에 json에 포함 안됨.  
    console.log(`${name} can jump!`);  
  },  
};  
  
json = JSON.stringify(rabbit);  
const obj = JSON.parse(json);  
console.log(obj);
```



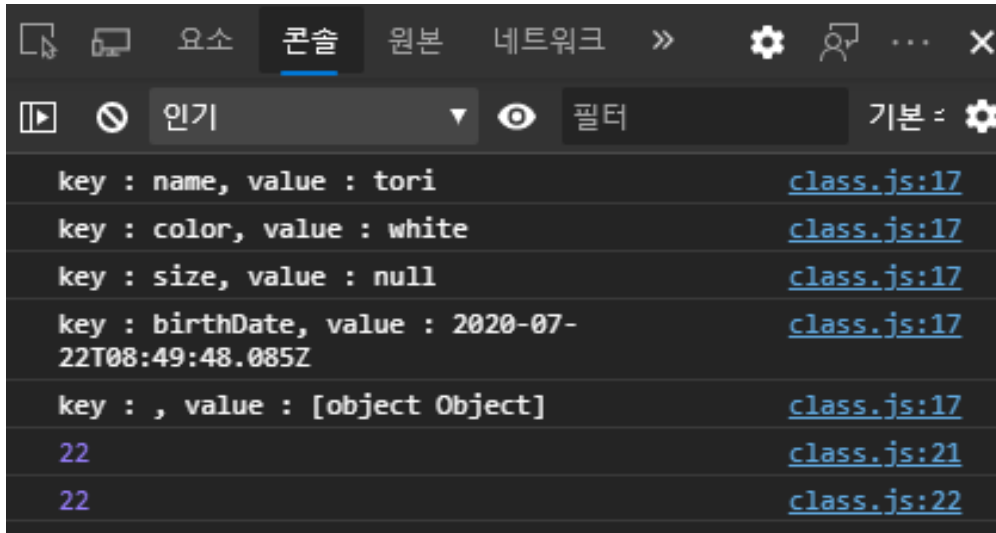
# 1. JSON

- JSON parse 실습

```
const rabbit = {
  name : 'tori',
  color : 'white',
  size : null,
  birthDate: new Date(),
  jump: () => { // 함수는 data 아니기에 json에 포함 안됨.
    console.log(`${name} can jump!`);
  },
};

json = JSON.stringify(rabbit);
const obj = JSON.parse(json, (key, value) => {
  console.log(`key : ${key}, value : ${value}`);
  return key === 'birthDate' ? new Date(value) : value;
});
console.log(obj);

console.log(rabbit.birthDate.getDate());
console.log(obj.birthDate.getDate());
```



# 1. JSON



- 추가 내용

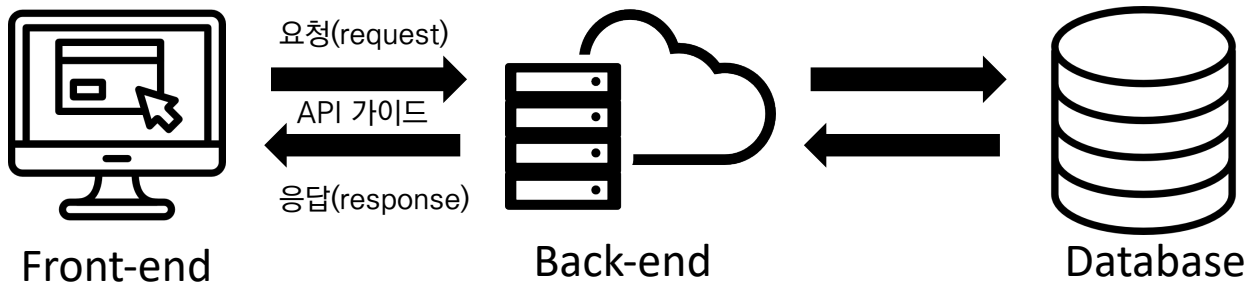
- 1 데이터 표현하는 방식 중 xml
- 2 현재는 json을 많이 쓰는 추세(인간, 기계 모두 이해하기 쉬움)
- 3 Ajax 구현에 사용
- 4 유용한 사이트 (JSON Diff, JSON Beautifier, JSON Parser, JSON Validator)



## 2. API

- API의 개념

- 1 API: Application Programming Interface의 약자
- 2 어떠한 응용프로그램에서 운영체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있도록 만든 인터페이스
- 3 특정 사이트에서 특정 데이터를 공유할 경우 어떤 방식으로 정보를 요청해야 하는지, 어떠한 데이터를 제공 받을 수 있는지에 대한 규격
- 4 기업이나 정부에서는 방대한 데이터를 활용할 수 있도록 API 형태로 데이터를 제공 Ex) 카카오, 페이스북, 구글 등



## 2. API

- API 가이드: 요청과 응답의 형식이 작성된 문서

- 1 요청(request)과 응답(response)은 사이트마다 정해진 형식에 맞춰서 전달 됨
- 2 요청(request): 서버에 정보를 달라고 하는 것
- 3 응답(response): 찾은 정보를 내 사이트가 받는 것

### 요청 (request)

1. 주소
2. 전송방식: GET or POST
3. 보낼 것

### 응답 (response)

1. 형식: JSON
2. 응답 내용

〈 공통적으로 가지는 내용 〉

## 2. API

### ▪ Open API

- 1 기업이나 정부에서는 방대한 데이터를 활용할 수 있도록 API 형태로 데이터를 제공
- 2 Back-end를 구성해 놓고, 주소와 사용 규칙을 공개한 것
- 3 serverless: 공개된 Back-End를 사용하고, Front-end만 만들어 사용하는 것
- 4 지도, 결제, 채팅 등 다양한 기능 이용 가능

- 네이버 디벨로퍼: <https://developers.naver.com/main>
- 카카오 디벨로퍼: <https://developers.kakao.com>
- 구글 디벨로퍼: <https://developers.google.com>
- 공공 데이터 포털: <https://www.data.go.kr>

## 2. API

### ▪ Open API 실습

- 1 준비 1 – API 키 발급받기    – 카카오 디벨로퍼: <https://developers.kakao.com>
- 2 준비 2 – 소스코드(뼈대)준비    – 깃허브: <https://github.com/Taehyeon-Kim/js-api-ex>

## 2. API

- 준비 1 – 카카오 API 키 발급받기
  - 1 카카오 디벨로퍼 사이트 로그인
  - 2 상단의 내 어플리케이션 클릭
  - 3 + 어플리케이션 추가하기 버튼 클릭
  - 4 아이콘 생략 / 앱 이름, 회사 이름 = “API연습”로 등록
  - 5 만들어진 어플리케이션 클릭 – REST API 키 사용



## 2. API

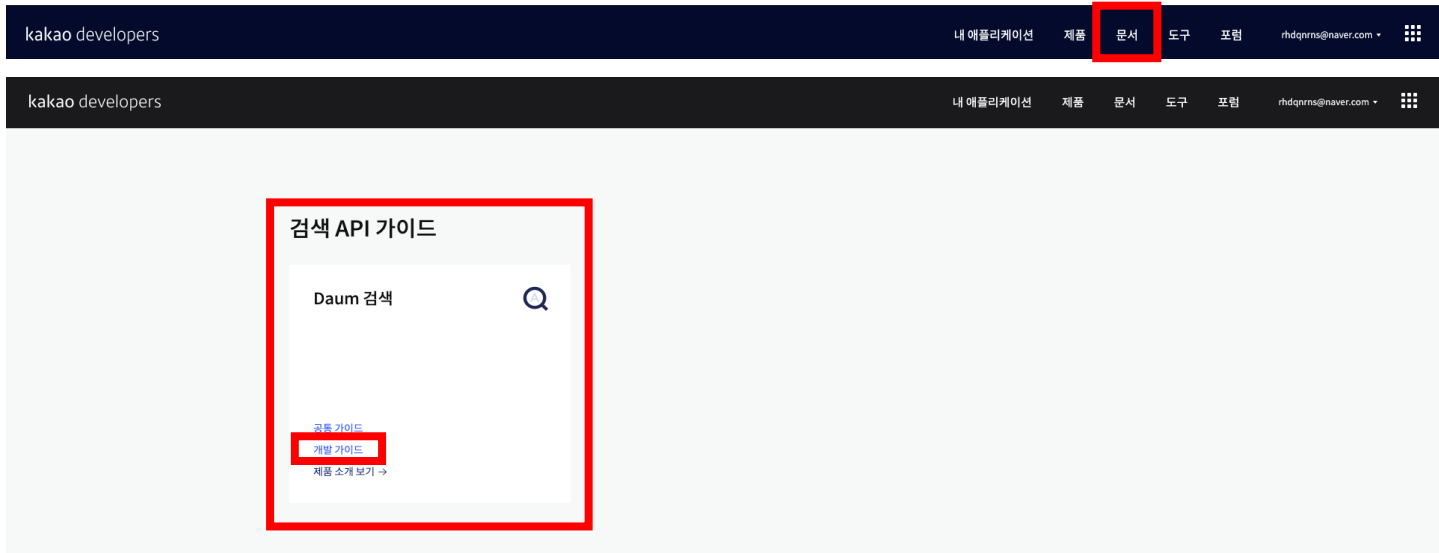
- 준비 2 – 소스코드 준비

- 1 깃허브에서 만들어진 소스(뼈대)코드 copy
- 2 로컬에 clone
- 3 Editor에서 소스코드 열기



## 2. API

- 참고 - 책 검색 API 가이드



# 2. API

## ■ 책 검색 API 가이드

kakao developers

내 애플리케이션

제품

문서

도구

포럼

rhdgqms@naver.com

문서 > Daum 검색 > 개발 가이드

푸시 알림

Daum 검색

공통 가이드

개발 가이드

웹문서 검색

동영상 검색

이미지 검색

블로그 검색

책 검색

카툰 검색

지도

로컬

카카오내비

비전

포즈

번역

음성

SDK 다운로드

유용한 참고 정보

검색을 원하는 질의어

결과 문서 정렬 방식, accuracy(정확도순) 또는 recency(최신순), 기본 값 accuracy

결과 페이지 번호, 1~50 사이의 값, 기본 값 1

한 페이지에 보여질 문서 수, 1~50 사이의 값, 기본 값 10

검색 필터 제한  
사용 가능한 값: title(제목), isbn (ISBN), publisher(출판사), person(인명)

Request

URL

```
GET /v3/search/book HTTP/1.1
Host: dapi.kakao.com
Authorization: KakaoAK {app_key}
```

Parameter

Name	Type	Description	Required
query	String	검색을 원하는 질의어	O
sort	String	결과 문서 정렬 방식, accuracy(정확도순) 또는 recency(최신순), 기본 값 accuracy	X
page	Integer	결과 페이지 번호, 1~50 사이의 값, 기본 값 1	X
size	Integer	한 페이지에 보여질 문서 수, 1~50 사이의 값, 기본 값 10	X
target	String	검색 필터 제한 사용 가능한 값: title(제목), isbn (ISBN), publisher(출판사), person(인명)	X

Response

meta

↑



자바스크립트 열심히!

**감사합니다**



2020. 07. 24