

비개발자를 위한

API 활용 강좌

2019.04.30 Ken





아.. 시원한 콜라 마시고 시프다 ㅠㅠ





건축정보



교통사고정보



국민건강정보



상권정보



수산정보



실시간 수도정보



농수축산가격정보



등산로 정보



부동산종합정보



통합재정정보



지방행정정보



부동산거래정보



식의약품종합정보



지방재정정보



법령정보



+ 더보기

전체(29,145)

파일데이터(25,986)

오픈API(3,067)

표준데이터(92)

오픈API 3,067건을 찾았습니다.

기관별 검색

오픈API [3,067건]

정확도

날 짜

제 목

조회수

활용신청

한국문학 번역출간도서정보 조회수 : 263 바로가기 횟수 : 282

수정일 : 2017.04.13 기관 : 한국문학번역원 서비스유형 : LINK

한국문학 번역출간도서정보(한국어->외국어)

XML

교육

우체국위치 정보 조회수 : 335 바로가기 횟수 : 511

수정일 : 2017.08.22 기관 : 과학기술정보통신부 우정사업본부 서비스유형 : LINK

우체국이나 365코너, 무인창구, 우체통 등의 운영시간 주소 위치정보등을 공개함으로써 가까운 시설물 및 정보를 안내하여 ...

XML

과학기술

표준화 교육자료 조회수 : 17 바로가기 횟수 : 100

수정일 : 2015.05.06 기관 : 산업통상자원부 서비스유형 : LINK

"표준의활용과 필요성에 대한 쉬운접근을위해 스토리카툰형식으로 개발된 자료로서 소비자 및 수출기업이 표준의 중요성에 ...

XML

산업고용

농산물 주요품목 가격 조회수 : 20 바로가기 횟수 : 468

수정일 : 2015.09.10 기관 : 서울특별시농수산물공사 서비스유형 : LINK

ㅇ가락시장 농산물 주요품목 가격(등급별가격)을 제공하는 오픈API입니다. ㅇ신청URL : http://www.garak.co.kr/gongsa/...

XML

농축수산



Hacking or Magic

자... **1분** 안에
튜터링 튜터들의 모든 정보를
엑셀로 옮겨 보도록 하겠습니다.



Goal

API를 이용하여

팔딱팔딱 살아있는 데이터를

더 풍성하게 수집하여

각종 기획 및 보고서에 활용한다

Contents

- 1) API가 무엇인지 어떻게 활용되는지 이해한다.
- 2) API를 이용하여 실시간 데이터를 수집하는 법을 실습한다.
- 3) API통신에서 주로 사용되는 JSON 데이터 포맷을 알아본다.
- 4) 구글 스프레드시트 상에 API 데이터를 연계하는 법을 실습한다.

API란 무엇인가?

API?

- Application Programming Interface
- 이종 시스템간에
응용 프로그래밍이 가능하도록
정의된 명세
- 예) Win32, OpenGL, DirectX ...

API via HTTP

- 웹 애플리케이션 개발에서 다른 서비스에 요청을 보내고 응답을 받기 위해 **정의된 명세**를 일컫는다.
- 특정 정보를 제공할 목적으로 웹서비스 형태로 제공
- 요청 내용: **정보(Data)** 또는 행위(Action)

데이터 vs 정보



데이터

추상화



정보

- 날것 그대로의 식재료 (고기, 회, 야채)
- 고객 구매/환불 내역

- **의도**에 의해 조리된 요리 (삼계탕, 동태찌개, 샤브샤브)
- 4월 무빙패키지 재구매자 환불율

서비스 / 앱



(정보화된)
데이터



UI /UX

- 플레이팅
- BGM
- 인테리어
- 서빙

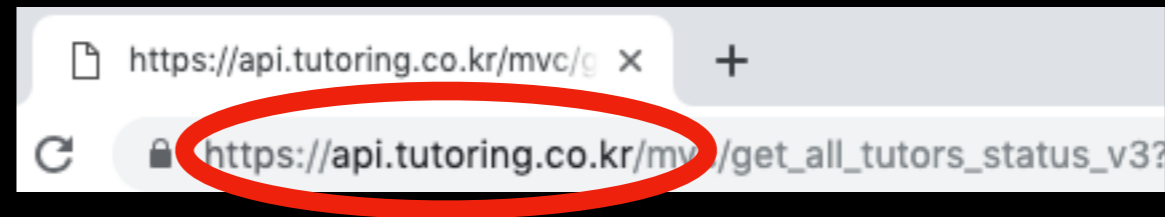
API의 효용

- 싱싱한 실시간 데이터가 필요함
- Raw(날것) 데이터는 활용하기 어려움
- 정보화된 데이터만 손쉽게 가져다 활용할 수 있으면 참 좋겠음
- UI/UX 적용하는 것은 고비용 (개발은 대고객 서비스 우선)
- 내부 활용을 위해서는 데이터만 손쉽게 갖다 쓸수 있어야함

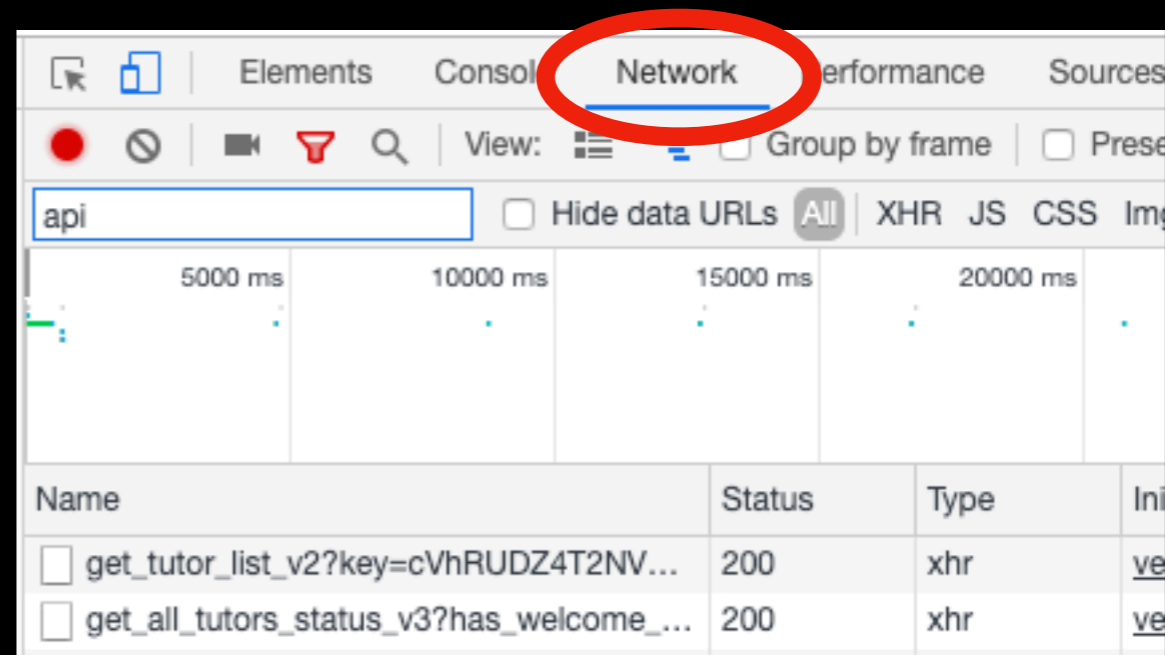


Keywords

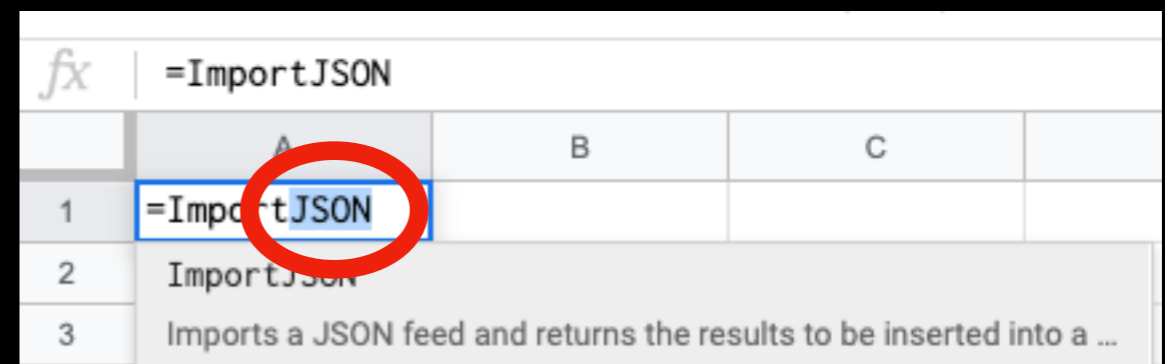
URL



Network



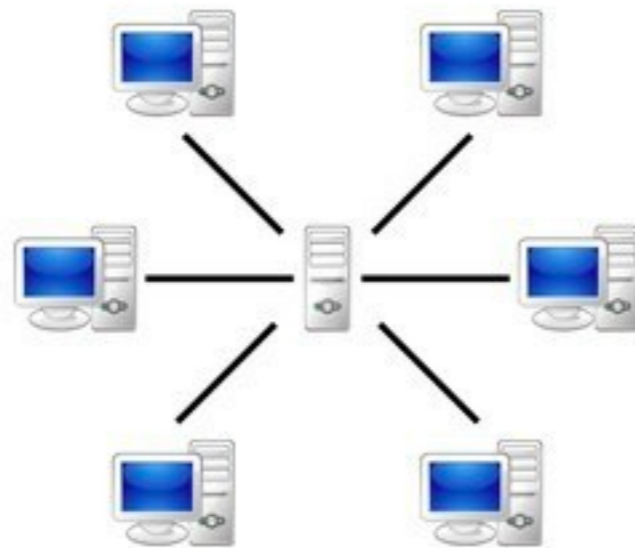
JSON



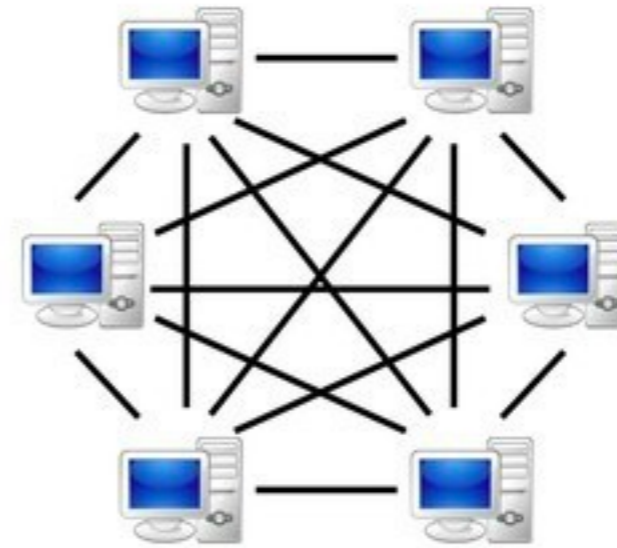
네트워크 기초 다지기

네트워크

- 물리적으로 떨어져 있는 Peer 간의 연결 기술



Server-based



P2P-network

Internetworking

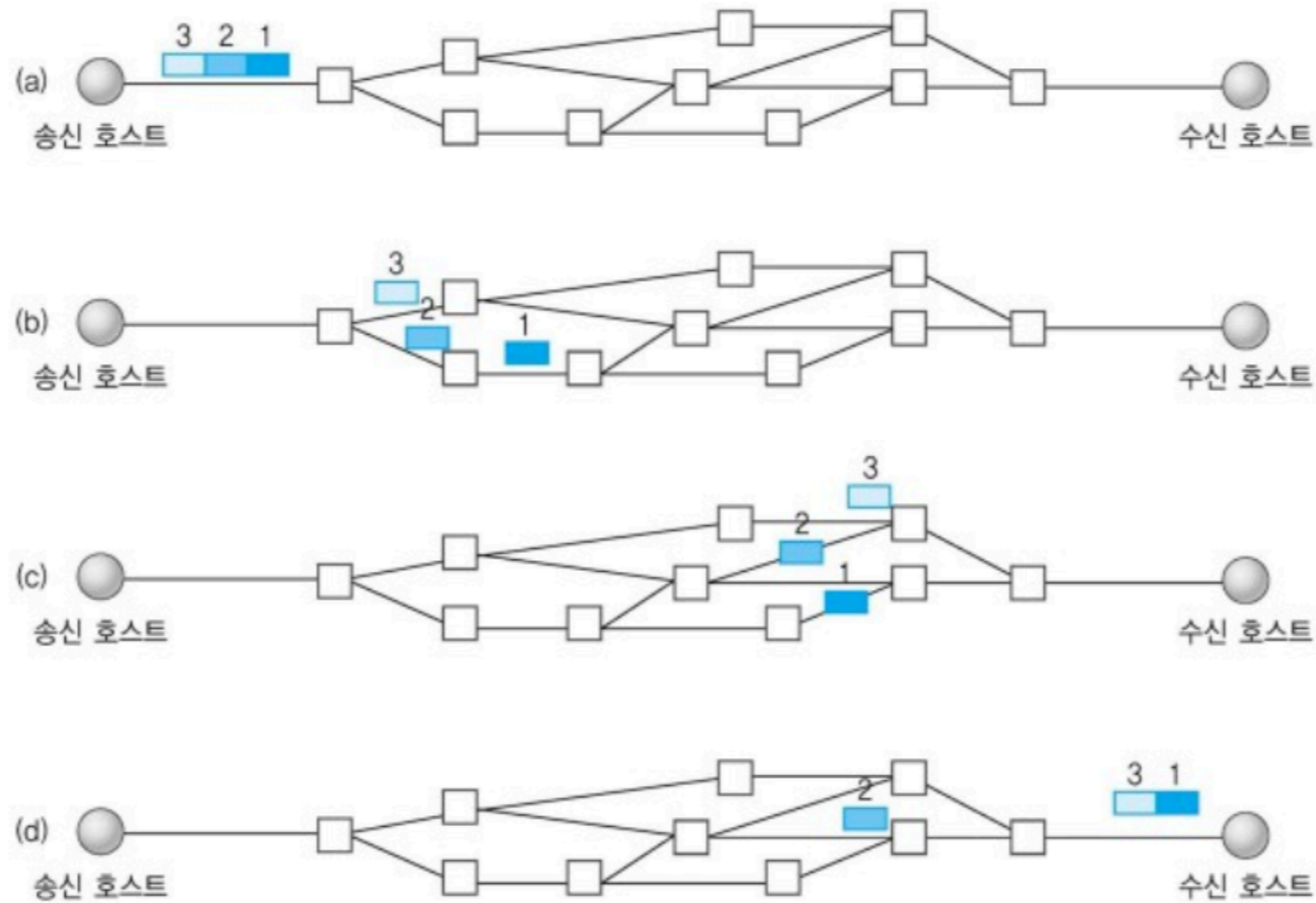
ARPAnet (1969년)

- 군사용
- 소수 대학 및 연구소 네트워크 간 연결
- 패킷교환방식: NCP

Internet (1983년)

- 민간용
- 모든 기관, 개인의 네트워크 간 연결
- 패킷교환방식: TCP / IP

패킷교환방식



< 데이터그램 패킷 교환 방식 >

패킷 = 오토바이

네트워크 계층

OSI 7 Layer	
L7	응용 계층 (Application Layer)
L6	표현 계층 (Presentation Layer)
L5	세션 계층 (Session Layer)
L4	전송 계층 (Transport Layer)
L3	네트워크 계층 (Network Layer)
L2	데이터 링크 계층 (Data Link Layer)
L1	물리 계층 (Physical Layer)

TCP/IP 4 Layer	
L4	응용 계층 (Application Layer)
L3	전송 계층 (Transport Layer)
L2	인터넷 계층 (Internet Layer)
L1	네트워크 액세스 (Network Access Layer)

**L4. 응용 프로그램 구현시 사용
(프로토콜: HTTP, FTP, SMTP, SSH)**

L3. 패킷 송수신 (프로토콜: TCP, UDP)

**L2. 네트워크간 식별 및 라우팅
(프로토콜: IP)**

L1. 물리적 식별 (MAC 주소)

프로토콜

- 통신 상호간에 약속된 송수신 규칙
- 여보세요~ / 모시모시~ / 알았다 오버~
- 암호호 / 모스부호 / 바코드 / QR코드

TCP & UDP

- 전송 계층에서 사용 되는 프로토콜

TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
연결형	비연결형
신뢰도 높음	신뢰도 낮음
속도 느림	속도 빠름
1:1	1:1, 1:N, N:N
전화	라디오/무전기

HTTP

- HyperText Transfer Protocol
- 응용 계층에서 사용되는 프로토콜
- 서버 - 클라이언트 사이에 요청과 응답으로 통신
- 인터넷상에서 주로 HTML 문서를 주고 받기 위해 쓰임
- HTML 이외에 다양한 데이터를 요청/응답 받을수 있음
 - 텍스트 기반: XML, JSON 등
 - 바이너리 기반: 이미지, 동영상, PDF 등
- HTTPS
 - HTTP = 일반우편
 - HTTPS = 등기 + 내용증명

Client - Server

클라이언트

- 서비스를 요청하는 Peer
- API 클라이언트 = HTTP 클라이언트 (브라우저, 포스트맨 등)

서버

- 클라이언트로 부터 받은 요청을 처리 및 응답하는 Peer
- API 서버 = 웹서버 (Web Application Server)

인터넷 브라우저

- HTTP 클라이언트
 - HTML 수신
 - API 데이터 수신 (동적 데이터)
 - 기타 리소스 파일 수신
- HTML/CSS 해석 (정적 UI)
- Javascript 해석 (동적 UI, 동적 데이터 바인딩)
- UI 렌더링 엔진 (화면상에 프린팅)

URL 넌 누구냐!

https://api.tutoring.co.kr:8080/eng/tutor/profile?email=ken%40tutoring.co.kr&t_type=native

URL : Uniform Resource Locator

정형화된 자원 위치

(예전에는 네트워크상의 물리적인 파일위치 주로 사용)

URI : Uniform Resource Identifier

정형화된 자원 식별자

(API등에서 사용되는 가상자원의 고유 ID로 사용)

프로토콜 (protocol)

https://api.tutoring.co.kr:8080/eng/tutor/profile?email=ken%40tutoring.co.kr&t_type=native

포트 (port)

https://api.tutoring.co.kr:8080/eng/tutor/profile?email=ken%40tutoring.co.kr&t_type=native

- **고유 서비스별로 할당된 숫자**
 - 예) 전화선 / 인터넷선 / 전력선 / 수도관 / 가스관
- **프로토콜마다 표준 포트가 지정되어 있음**
 - HTTP = 80
 - HTTPS = 443
 - FTP = 21
 - SSH = 22
 - TELNET = 23
 - MySQL = 3306

호스트 (host)

https://api.tutoring.co.kr:8080/eng/tutor/profile?email=ken%40tutoring.co.kr&t_type=native

- 도메인 = 건물 주소
- IP = 건물 위도/경도 좌표

Path

https://api.tutoring.co.kr:8080/eng/tutor/profile?email=ken%40tutoring.co.kr&t_type=native

- **path** = 층/호실
- **filename** = 사람이름

QueryString

https://api.tutoring.co.kr:8080/eng/tutor/profile?email=ken%40tutoring.co.kr&t_type=native

- **key = value**
- 구분자 & 사용 : **key1=value1&key2=value2&...**
- 스페이스, 특수문자 사용시 **URLEncoding/URLDecoding** 필요
- **<http://www.convertstring.com/ko/EncodeDecode/UrlDecode>**

REST API & JSON

REST API

- <https://meetup.toast.com/posts/92>
- HTTP 프로토콜의 **메소드**를 활용하여 웹서비스에 CRUD 서비스를 요청

CRUD	Method
Create	POST
Read	GET
Update	PUT
Delete	DELETE

JSON

- JavaScript Object Notation
- 이종 시스템간 데이터를 효과적으로 전달하기 위해 고안된 데이터 표기법
- CSV: 복잡하고 재귀적인 구조의 데이터는 표현 불가
- XML: 장황하고 복잡, 무겁고 느림
- JSON: 유연하고 빠름, 모든 구조의 데이터 표현 가능
- CSV -> XML -> JSON

JSON 포맷

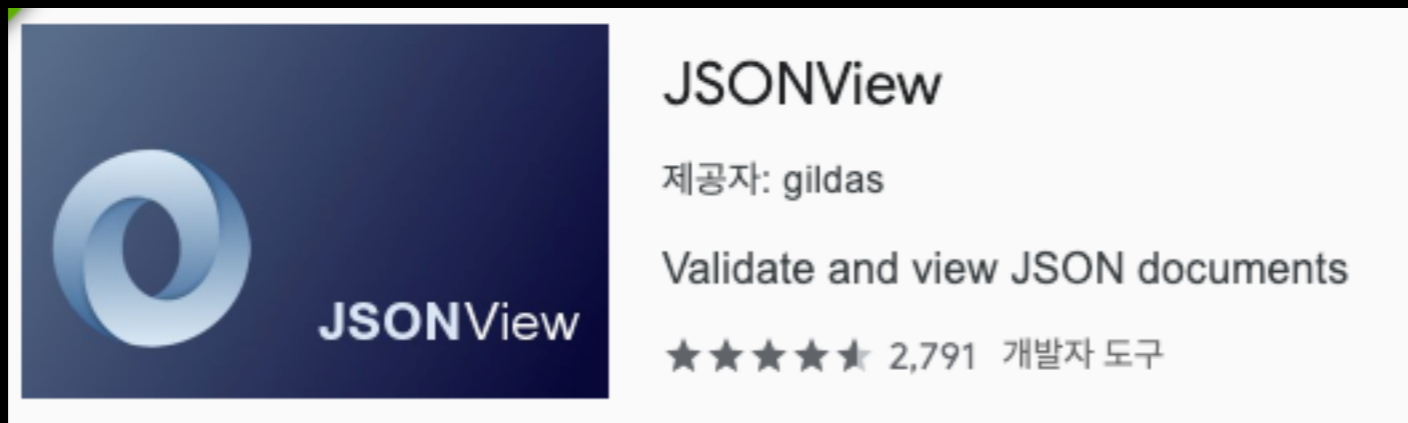
- 루트요소는 단일 object 또는 array로 이루어져 있음
- `array` = [object, object,]
- `object` = {attribute, attribute, ...}
- `attribute` = key : value
- key = 문자열
- value = 수, 문자열, `object`, `array` (nested 구조)
- <https://ko.wikipedia.org/wiki/JSON>

휴식

크롬 브라우저 JSON 뷰어 플러그인 추가하는 법

1. <https://chrome.google.com/webstore> 크롬 웹스토어 이동

2. "JSONView" 검색



3.

Chrome에 추가

4. 이후, JSON 리턴하는 URL 진입시 **아름답게** 표시됨
(테스트 URL : <https://api.androidhive.info/contacts/>)

Postman 맛보기

메모 보내기

No.	이름	Endpoint (path)	Method	Params(명세)	Params(예제)
	호스트	https://api.tutoring.co.kr			
1	Create a memo	/lab/memo	POST	fr: 송신자ID to: 수신자ID message: 메세지	fr: ken to: jin message: 하나들 테스트
2	Read a memo ----- Read list	/lab/memo	GET	fr: 송신자ID to: 수신자ID ----- id: 나의 ID list: inbox (받은 메세지) or sent (보낸 메세지)	fr: ken to: jin ----- id: ken list: inbox
3	Update a memo	/lab/memo	PUT	fr: 송신자ID to: 수신자ID message: 메세지	fr: ken to: jin message: 하나들 테스트
4	Delete a memo	/lab/memo	DELETE	fr: 송신자ID to: 수신자ID	fr: ken to: jin

POP Quiz

No.	이름	Endpoint (path)	Method	Params(명세)	Params(예제)
	호스트	https://api.tutoring.co.kr			
1	답 제출	/lab/quiz	POST	id: 제출자ID q: 문제번호 a: 답	id: ken q: 1 a: C
2	답 수정	/lab/quiz	PUT	id: 제출자ID q: 문제번호 a: 답	id: ken q: 1 a: B
3	제출답안 조회	/lab/quiz	GET	id: 제출자ID	id: ken
4	등수 조회	/lab/rank	GET		

POP Quiz

1. attribute

2. POST

3. IP

4. HTTP

5. 패킷

6. 프로토콜

7. HTTPS

8. READ

9. object

10. Path

A. 여보세요

B. {}

C. 80

D. 층/호실

E. CREATE

F. key : value

G. 오토바이

H. 위도/경도

I. GET

J. 등기+내용증명

POP Quiz (정답)

-
1. attribute
2. POST
3. IP
4. HTTP
5. 패킷
6. 프로토콜
7. HTTPS
8. READ
9. object
10. Path
- A. 여보세요
B. {}
C. 80
D. 층/호실
E. CREATE
F. key : value
G. 오토바이
H. 위도/경도
I. GET
J. 등기+내용증명
- Connections:
1. attribute -> A. 여보세요
2. POST -> E. CREATE
3. IP -> H. 위도/경도
4. HTTP -> C. 80
5. 패킷 -> G. 오토바이
6. 프로토콜 -> F. key : value
7. HTTPS -> J. 등기+내용증명
8. READ -> I. GET
9. object -> B. {}
10. Path -> D. 층/호실

Open API
Google spread sheets
연동 실습

API 보안 이슈

- 키 : Ken
- 암호화된키 : Tas3aQNvSTVsMURHTzJQQT07
- 보안토큰
 - HTTP 헤더에 사용
 - 유효기간 지정
 - 공개키/개인키(키페어)를 이용한 송수신 정보 암호화

API 성능 이슈

- API 키 발급을 통해 사용자 식별
- IP별 / 사용자별 요청콜수 제한
 - 예) 사용자별 분당 최대 100회 제한
IP별 초당 최대 5회 제한
- Endpoint별 컴퓨팅 시간에 따른 가중치 부여
 - 예) get_user : score 1, get_all_users : score 5
사용자별 분당 최대 50 score까지 제한

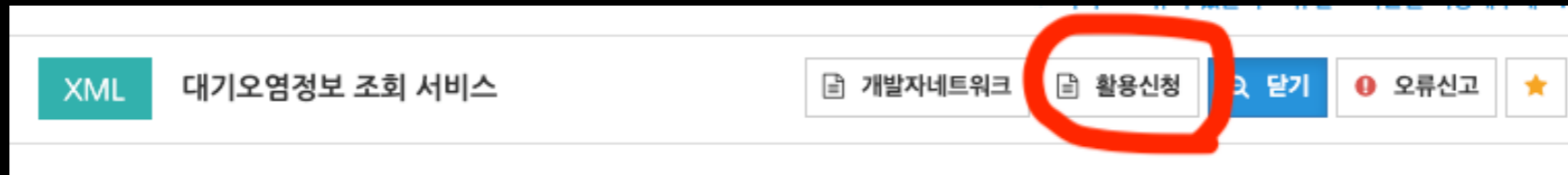
공공데이터 포털

미세먼지 데이터 가져오기

<http://data.go.kr>

공공데이터 서비스 활용 신청

<https://www.data.go.kr/dataset/15000581/openapi.do>



개발계정 신청

기본정보

서비스명	대기오염정보 조회 서비스	서비스 유형	REST
심의여부	자동승인	신청유형	개발계정 활용신청
처리상태	신청	활용기간	승인일로부터 2개월 간 활용가능

시스템유형 선택

* 시스템 유형 일반 서버 구축

* 일반 : OpenAPI 서비스를 호출하여 응답받은 결과값을 서버에 저장하지 않고 사용할 경우 (서버 미구축)
 * 서버 구축 : OpenAPI 서비스를 호출하여 응답받은 결과값을 서버에 저장하거나 DB화 하여 사용할 경우

안내

*** 위치기반서비스 사업자 확인**

공공데이터중 위치정보를 포함한 서비스를 사용하고자 하는 사업자는 '위치정보의 보호 및 이용 등에 관한 법률'에 따라 방송통신위원회에 '위치정보서비스 허가'를 받거나 '위치기반 서비스사업 신고'를 하여야 합니다. 이에 해당하는 사업자인 경우에는 첨부파일에 '위치기반서비스사업신고필증'을 첨부해 주시기 바랍니다. 활용신청 시 '위치기반서비스사업신고필증'이 등록되지 않으면 반려가 될 수 있으니 참고하시기 바랍니다.

활용정보

* 활용목적 웹 사이트 개발 앱개발 (모바일,솔루션등) 기타 참고자료 연구(논문 등)

첨부파일 **한 개의 파일만 첨부 할 수 있습니다.**
 선택된 파일 없음

상세기능정보 필수 입력 정보입니다. *자동승인 상세기능은 신청과 동시에 활용 가능합니다.

<input checked="" type="checkbox"/>	상세기능	설명	일일 트래픽
<input checked="" type="checkbox"/>	시군구별 실시간 평균정보 조회	시도의 각 시군구별 측정소목록의 일반 항목에 대한 시간대별 평균농도를 제공하는 시군구별 실시간 평균정보 조회	500
<input checked="" type="checkbox"/>	시도별 실시간 평균정보 조회	시도별 측정소목록에 대한 일반 항목의 시간 및 일평균 자료 및 지역 평균 정보를 제공하는 시도별 실시간 평균정보 조회	500
<input checked="" type="checkbox"/>	대기질 예보통보 조회	통보코드와 통보시간으로 예보정보와 발생 원인 정보를 조회하는 대기질(미세먼지/오존) 예보통보 조회	500
<input checked="" type="checkbox"/>	시도별 실시간 측정정보 조회	시도명을 검색조건으로 하여 시도별 측정소목록에 대한 일반 항목과 CAI 최종 실시간 측정값과 지수 정보 조회 기능을 제공하는 시도별 실시간 측정정보 조회	500
<input checked="" type="checkbox"/>	통합대기환경지수 나쁨 이상 측정소 목록조회	통합대기환경지수가 나쁨 등급 이상인 측정소명과 주소 목록 정보를 제공하는 통합대기환경지수 나쁨 이상 측정소 목록조회	500
<input checked="" type="checkbox"/>	측정소별 실시간 측정정보 조회	측정소명과 측정데이터 기간(일, 한달, 3개월)으로 해당 측정소의 일반 항목 측정정보를 제공하는 측정소별 실시간 측정정보조회	500

라이선스표시

이용허락범위 저작자표시 동의합니다 (사유: 동의합니다)

공공데이터 포털에서 JSON 포맷으로 API URL 추출하는 법

(한국환경공단 미세먼지 경보 발령 현황 예시)


1. data.go.kr 에서 회원가입 (회원가입시 이메일 인증 필요)
2. 데이터셋 > 오픈 API 에서 "미세먼지"로 검색
3. "한국환경공단_미세먼지 경보 발령 현황" 선택
4. "활용신청" 클릭
5. 시스템 유형 : 일반
활용목적: 참고자료
상세기능정보: 모두선택
라이선스표시: 동의
선택후 "신청" 클릭
6. 마이페이지 > 오픈API > 개발계정 리스트 > "미세먼지 경보 정보" 이동
7. "일반 인증키 발급" 클릭
8. 상세기능정보 > 실행 > 미리보기 클릭
(SERVICE ACCESS DENIED ERROR 가 나올경우 10분 정도 경과후 다시 시도, 인증키 활성화까지 최대 1시간 정도 소요)
9. XML포맷의 데이터가 표시되는 것을 확인
10. URL 마지막에 "&_returnType=json" 를 추가하여 새로고침 (JSON 포맷으로 표시되는 것을 확인)
11. 해당 URL을 Google Spread Sheets에서 import 하여 활용

Google spread sheets 활용

ImportJSON 설치

<https://github.com/bradjasper/ImportJSON>

ImportJSON 설치

1. <https://raw.githubusercontent.com/bradjasper/ImportJSON/master/ImportJSON.gs> 이동
2. 코드 전체를 복사하여 둠
(win:ctrl+A ctrl+C, mac: cmd+A cmd+C)
3. 구글 스프레드시트 > 도구 > 스크립트편집기
4. 기존 내용(myFunction) 삭제
5. 2. 에서 복사하여둔 코드 붙여넣기
6. 파일 > 저장 또는  클릭
7. 프로젝트 이름 수정 "아무거나" > 확인
8. 이후 해당 스프레드시트에서만 importJSON 함수 사용 가능
(새로운 스프레드시트 생성시마다 1~7작업을 반복하여야 함)

데이터 갱신

- URL 캐싱 회피 (timestamp 활용)
- 버튼 트리거 (수동갱신)
- 오픈시 트리거
- 타이머 트리거 (자동갱신)

매번 갱신된 데이터를 가져오는 코드 (URL캐싱 회피 코드 예제)

```
function refresh() {  
    var timestamp = new Date().getTime();  
    var cell = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet().getRange("F3");  
    cell.setFormula('=ImportJSON("https://...(해당URL)...&'+timestamp+'")');  
}
```

URL 입력부분 뒤에 **&'+timestamp+'** 을 추가

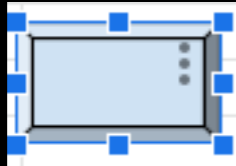
* 주의 : **&**를 누락하면 절대안됨

버튼 트리거

1. 삽입 > 그림 > 도형 선택



2. 버튼모양 도형 선택 > 드래그하여 그림 > "저장 후 닫기"



3. 삽입된 버튼그림 선택 > ⋮ > 스크립트 할당

4. 스크립트 할당 입력창에 앞에서 작성한 갱신함수명 "refresh" 입력함

5. 이후 버튼 클릭시마다 데이터 갱신하여 다시 가져옴

오픈시 트리거

```
function onOpen() {  
    refresh();  
}
```

1. onOpen 이라는 이름의 함수 추가
 2. 앞에서 작성한 refresh함수 실행코드 작성
- * 주의) 스프레드시트를 다시 열때마다 실행되어야하나

크롬브라우저 자체를 꺾다가 다시켜서 열때만 작동함

타이머 트리거

1. 스크립트 편집기에서 시계모양 아이콘

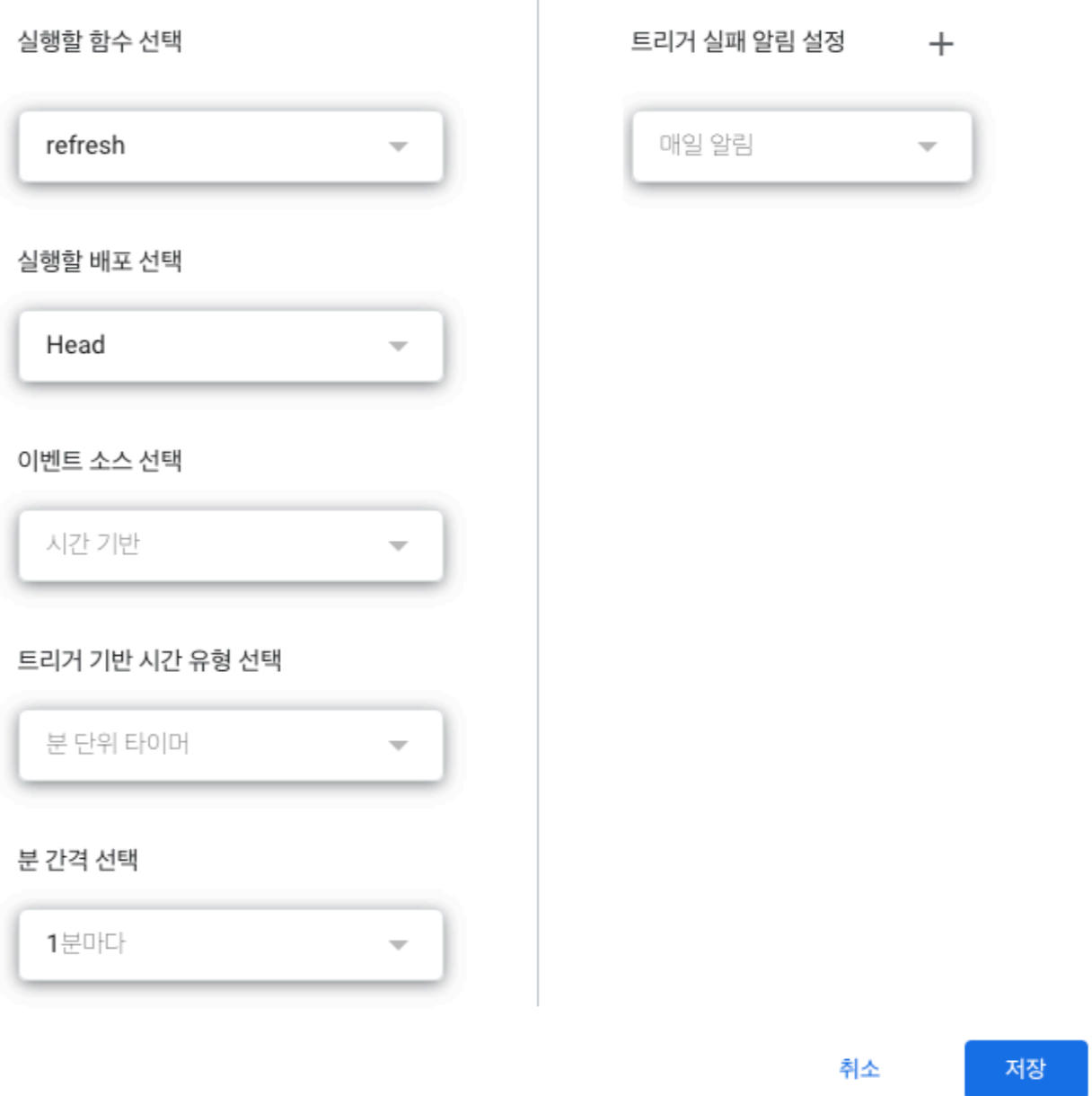
() 클릭

2. 트리거 화면 좌측 하단에

"+ 트리거 추가" 버튼 클릭

3. 우측 예시화면 참고하여 작성후 "저장"

4. 이후, 1분마다 데이터가 새로고침 됨



실행할 함수 선택

refresh

실행할 배포 선택

Head

이벤트 소스 선택

시간 기반

트리거 기반 시간 유형 선택

분 단위 타이머

분 간격 선택

1분마다

트리거 실패 알림 설정 +

매일 알림

취소 저장

실습

- 모든 튜터 정보
- 주요 KPI 데이터
- 시간별 매출 현황
- 일간 전환율 데이터

끝

집으로~