

WoT and HTTP Open API

Introduction to Internet and Web



부산대학교 정보·의생명 공학대학
정보컴퓨터공학부



실습 개요

❖ OPEN API 사용법 실습

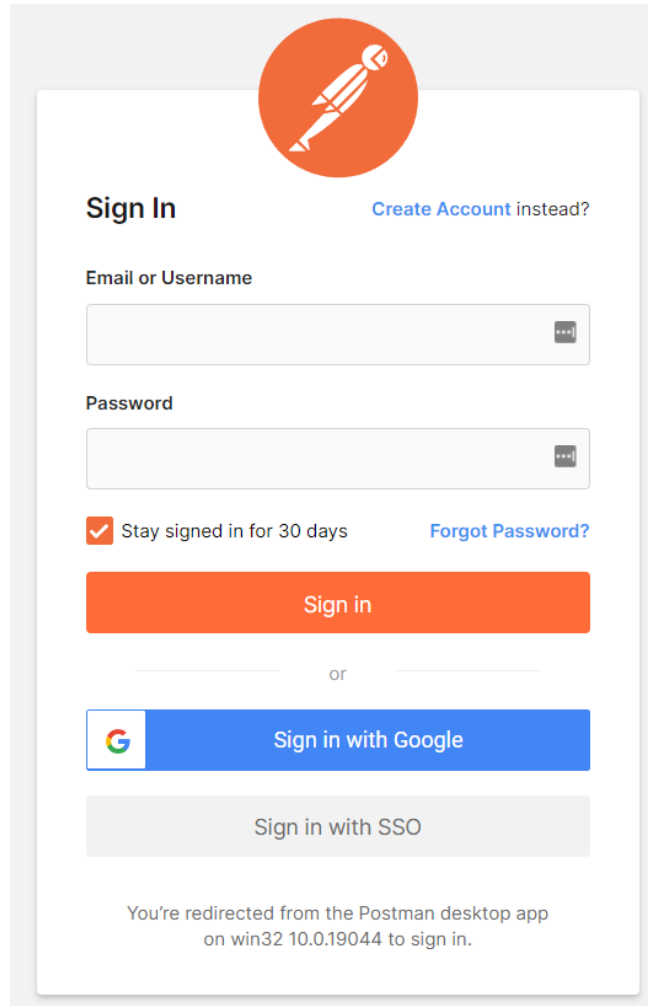
- PostMan 설치 및 사용법
- 공공데이터 API 사용 예제
- KAKAO, NAVER Search API 사용 예제

❖ WoT 예제

- Philips HUE API 사용

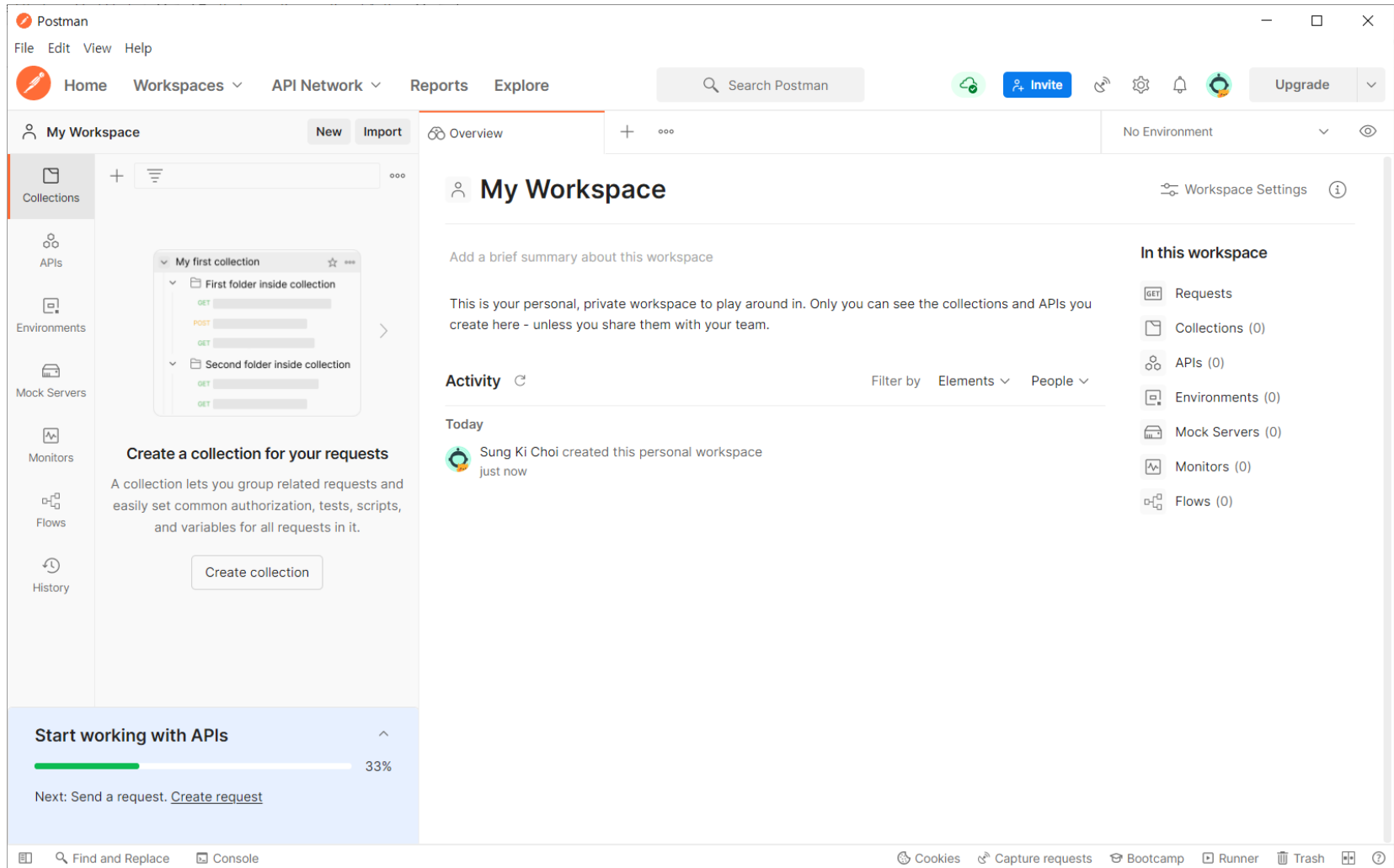
실습 준비 – Web API Test Client 프로그램 설치

❖ <https://www.postman.com/downloads/>



The image shows the Postman Sign In interface. At the top is the Postman logo (an orange circle with a white rocket). Below it, the text "Sign In" is displayed, with a link "Create Account instead?" to its right. There are two input fields: "Email or Username" and "Password", both with a small eye icon to toggle visibility. Below the password field is a checkbox labeled "Stay signed in for 30 days" and a link "Forgot Password?". A large orange "Sign in" button is positioned below these options. Below the button is a horizontal line with the word "or" in the center. Underneath the line are two buttons: "Sign in with Google" (with the Google logo) and "Sign in with SSO" (disabled). At the bottom, a message states: "You're redirected from the Postman desktop app on win32 10.0.19044 to sign in."

Postman 실행



API 호출 Test 준비

❖ Collection 추가 -> Request 추가

My Workspace

New Import

Overview New Collection

Collections

1

2

New Collection

This collection is empty

3

Add a request to start working.

Auth Pre-req. Tests Variables

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

Type

No Auth

This collection does not use any authorization.

[Learn more about authorization](#)



New Collection / New Request

GET Enter request URL

Send

Params Auth Headers (6) Body Pre-req. Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION		Bulk Edit
Key	Value	Description		

API 호출 Test (1)

❖ Web API 호출에 필요한 3가지 기본

요소

① 서버와 서비스 명 / API URL :

Ex) <https://postman-echo.com/get>

② HTTP Request Method

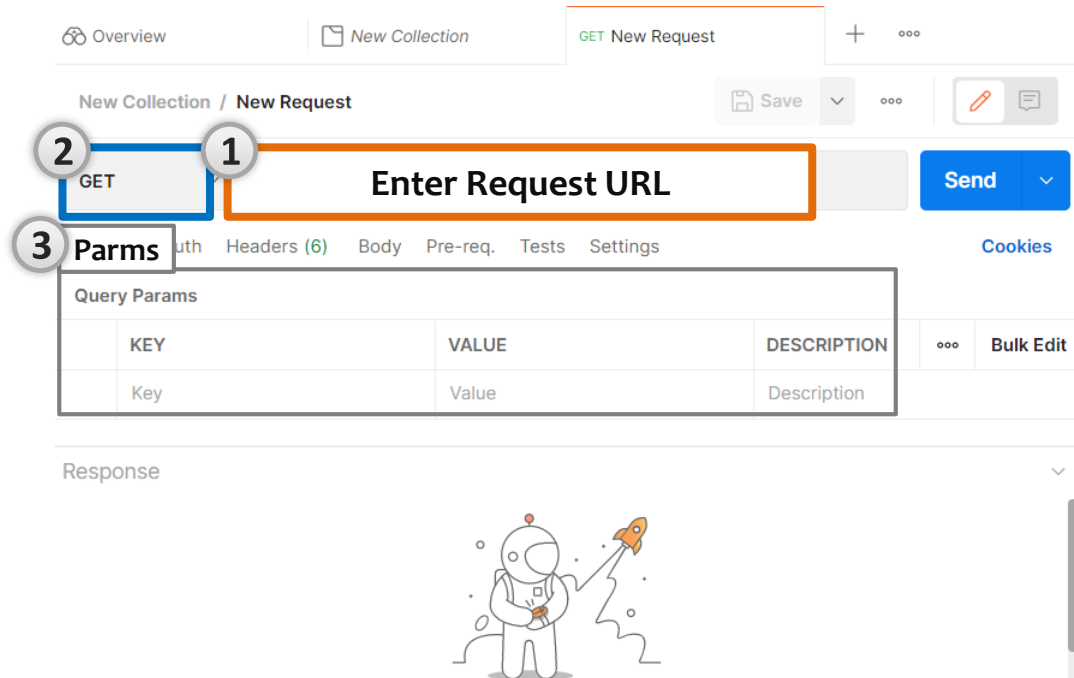
- GET, PUT, POST, DELETE, ...
- 당분간은 GET 만 사용

③ Parameters :

- API 종속적
- Key, Value로 구성

❖ Example

- <https://postman-echo.com/get>
- GET
- 2 Query Parameters (Key, Value)
 - foo1, bar1
 - foo2, bar2



API 호출 Test (2)

1. ①,②,③에 적절한 값을 입력/선택

2. "Send" Click

- API 호출
- Query Request

3. 결과 확인

- Query Response
- API 호출

The screenshot shows the Postman interface for a new request. The URL is `https://postman-echo.com/get`. The request method is GET. The query parameters are defined in a table:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	foo1	bar1	
<input checked="" type="checkbox"/>	foo2	bar2	
	Key	Value	Description

The response is shown in the 'Body' tab, displaying a JSON object:

```
{  "args": {    "foo1": "bar1",    "foo2": "bar2"  },  "headers": {  }}
```

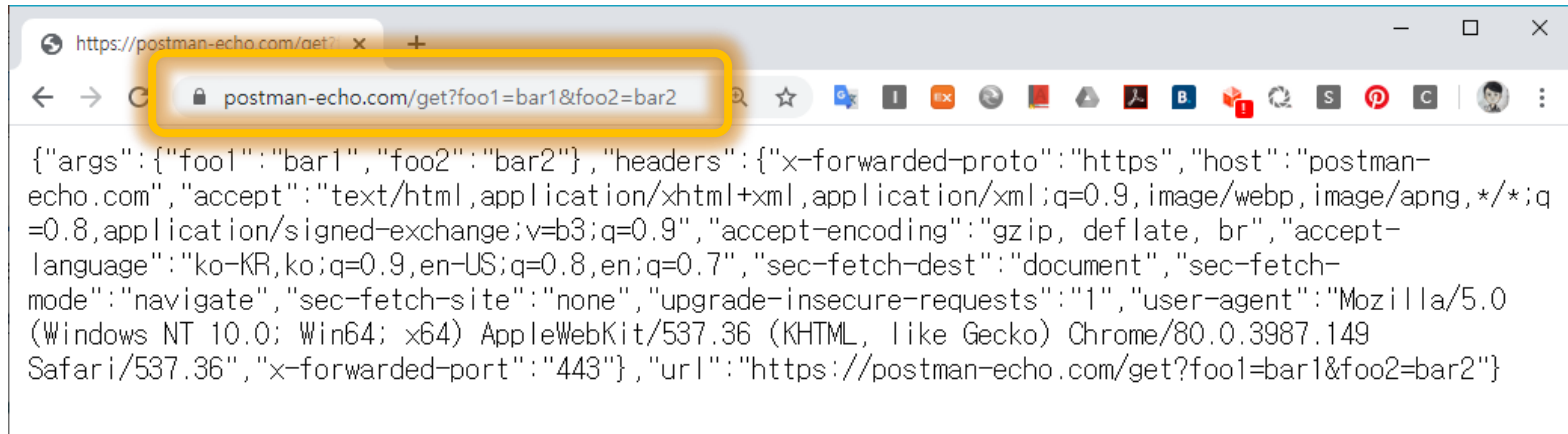
Annotations 1, 2, and 3 highlight the URL, the GET method, and the response body respectively.

HTTP Query URL

❖ Params 입력에 따른 URL 변화를 확인하라

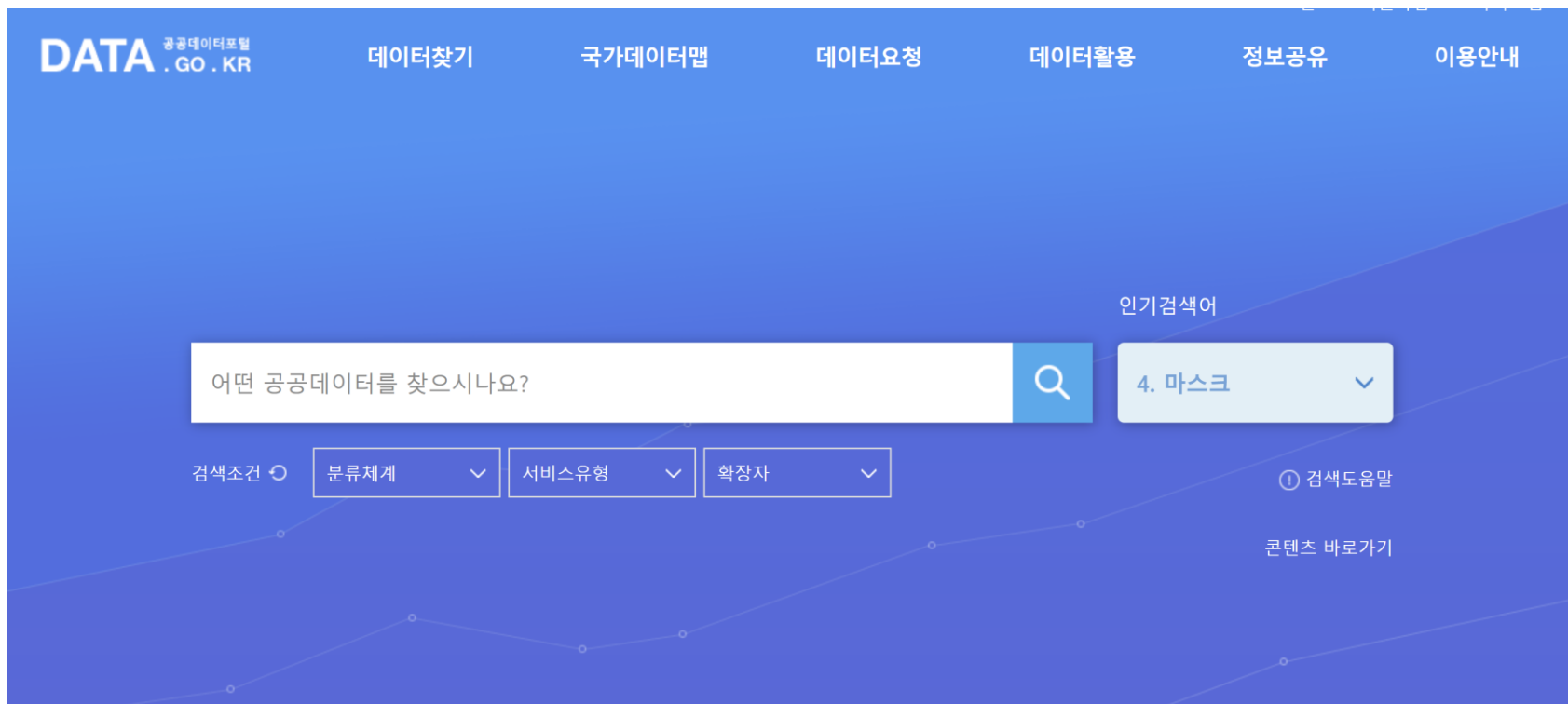
- `https://postman-echo.com/get`
- → `https://postman-echo.com/get?foo1=bar1&foo2=bar2`
 - "?" 다음부터 Params 시작
 - key=value의 형식으로 인자 표현
 - 인자의 구분은 "&"

❖ URL을 복사하여 일반 Browser의 주소창에 입력하고 결과를 Postman의 결과와 비교해 보라



공공데이터포털 계정 획득 및 Key 발급

❖ 공공데이터포털 – www.data.go.kr



❖ “기상청 단기예보”, “부산버스정보시스템” 검색

공공데이터 Open API 활용 신청과 참고문서 확인

오픈API 상세



XML JSON 기상청_단기예보 ((구)_동네예보) 조회서비스

초단기실황, 초단기예보, 단기((구)동네)예보, 예보버전 정보를 조회하는 서비스입니다. 초단기실황정보는 예보 구역에 대한 대표 AWS 관측값을, 초단기예보는 예보시점부터 6시간까지의 예보를, 단기예보는 예보기간을 글피까지 확장 및 예보단위를 상세화(3시간→1시간)하여 시공간적으로 세분화한 예보를 제공합니다.

47

4

관심

활용신청

오류신고 및

담당자 문의

OpenAPI 정보

메타데이터 다운로드

분류체계	과학기술 - 과학기술연구	제공기관	기상청
관리부서명	기상융합서비스과	관리부서 전화번호	042-481-7502
API 유형	REST	데이터포맷	JSON+XML
활용신청	7865	키워드	단기예보,초단기실황,초단기예보
등록	2021-06-28	수정	2022-03-31
심의유형	개발단계 : 허용 / 운영단계 : 허용		
비용부과유무	무료		
이용허락범위	  공공저작물_출처표시		
참고문서	기상청41_단기예보 조회서비스_오픈API활용가이드_최종.zip		

마이페이지를 통한 API 활용 승인 여부 확인

마이페이지

오픈API

개발계정

운영계정

인증키 발급현황

DATA

나의 문의

나의 관심

나의 제공신청

나의 분쟁조정

회원정보 수정

개발계정

신청 0건 >

신청중인 단계

· 보류 0건
· 반려 0건

활용 1건 >

승인되어 활용중인 단계

· 변경신청 0건

중지0

중지신청하여 운영

상세검색 열기

총1건

과학기술

기상청

활용신청 [승인] 기상청_단기예보 ((구)_동네예보) 조회서비스

신청일 2022-04-29 만료예정일 2024-04-29

Click

개발 계정 상세 보기

기본정보

데이터명	기상청_단기예보 ((구)_동네예보) 조회서비스 상세설명		
서비스유형	REST	심의여부	자동승인
신청유형	개발계정 활용신청	처리상태	승인
활용기간	2022-04-29 ~ 2024-04-29		

서비스정보

참고문서	기상청41_단기예보 조회서비스_오픈API활용가이드_최종.zip
데이터포맷	JSON+XML
End Point	http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0
<p>API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다.</p> <p>포털에서 제공되는 Encoding/Decoding 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다.</p> <p>* 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.</p>	
일반 인증키 (Encoding)	Vnrso4ayvW2S%2B26mo5JHOJpGzpXdp4H88RzhvTm0MoSVUKDszT5983m0mC0gkZ3ktiBKAyUYmzJjfzOU%2FP%2BK6Q%3D%3D
일반 인증키 (Decoding)	Vnrso4ayvW2S+26mo5JHOJpGzpXdp4H88RzhvTm0MoSVUKDszT5983m0mC0gkZ3ktiBKAyUYmzJjfzOU/P+K6Q==

Manual의 중요성 - 읽고 이해하기, 작성하기

[출처] 공공데이터포털 - 기상청 동네예보 조회서비스 오픈API활용가이드

API 서비스 정보	API명(영문)	VilageFcstInfoService_2.0		
	API명(국문)	단기예보 조회서비스(2.0)		
	API 설명	초단기실황, 초단기예보, 동네예보, 예보버전 정보를 조회하는 서비스		
API 서비스 보안적용 기술 수준	서비스 인증/권한	[O] ServiceKey [] 인증서 (GPKI/NPKI) [] Basic (ID/PW) [] 없음		
	메시지 레벨 암호화	[] 전자서명 [] 암호화 [O] 없음		
	전송 레벨 암호화	[] SSL [O] 없음		
	인터페이스 표준	[] SOAP 1.2 (RPC-Encoded, Document Literal, Document Literal Wrapped) [O] REST (GET) [] RSS 1.0 [] RSS 2.0 [] Atom 1.0 [] 기타		
	교환 데이터 표준 (중복선택가능)	[O] XML [O] JSON [] MIME [] MTOM		
	서비스 URL	http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0		
API 서비스 배포정보	서비스 명세 URL (WSDL 또는 WADL)	N/A		
	서비스 버전	1.0		
	서비스 시작일	2021-07-01	서비스 배포일	2021-07-01
	서비스 이력	2021-07-01 : 서비스 시작		
	메시지 교환유형	[O] Request-Response [] Publish-Subscribe [] Fire-and-Forgot [] Notification		
	데이터 갱신주기	수시 (일 8회)		

번호	API명(국문)	상세기능명(영문)	상세기능명(국문)
1	단기예보 조회서비스	getUltraSrtNcst	초단기실황조회
2		getUltraSrtFcst	초단기예보조회
3		getVilageFcst	단기예보조회
4		getFcstVersion	예보버전조회

상세 기능 내역 - [초단기실황조회]

상세 기능 내역

상세기능 번호	1	상세기능 유형	조회 (목록)
상세기능명(국문)	초단기실황조회		
상세기능 설명	실황정보를 조회하기 위해 발표일자, 발표시각, 예보지점 X 좌표, 예보지점 Y 좌표의 조회 조건으로 자료 구분코드, 실황값, 발표일자, 발표시각, 예보지점 X 좌표, 예보지점 Y 좌표의 정보를 조회하는 기능		
Call Back URL	http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtNcst		
최대 메시지 사이즈	[1764] byte		
평균 응답 시간	[100] ms	초당 최대 트랙잭션	[30] tps

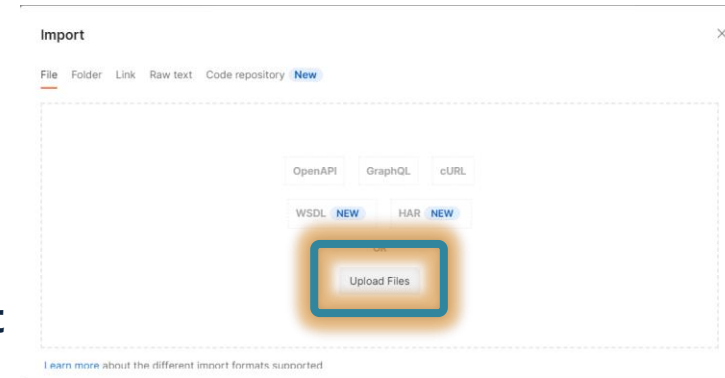
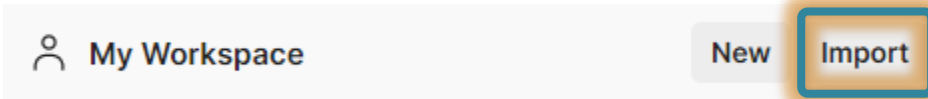
요청 메시지 명세

항목명(영문)	항목명(국문)	항목크기	항목구분	샘플데이터	항목설명
serviceKey	인증키	100	1	인증키 (URL Encode)	공공데이터포털에서 발급받은 인증키
numOfRows	한 페이지 결과 수	4	1	10	한 페이지 결과 수 Default: 10
pageNo	페이지 번호	4	1	1	페이지 번호 Default: 1
dataType	응답자료형식	4	0	XML	요청자료형식(XML/JSON) Default: XML
base_date	발표일자	8	1	20210628	'21년 6월 28일 발표
base_time	발표시각	4	1	0600	06시 발표(정시단위) -매시각 40분 이후 호출
nx	예보지점 X 좌표	2	0	55	예보지점의 X 좌표값 *별첨 엑셀 자료 참조
ny	예보지점 Y 좌표	2	0	127	예보지점의 Y 좌표값 *별첨 엑셀 자료 참조

※ 항목구분 : 필수(1), 옵션(0), 1건 이상 복수건(1..n), 0건 또는 복수건(0..n)

공공 API 호출 Test - Postman 활용

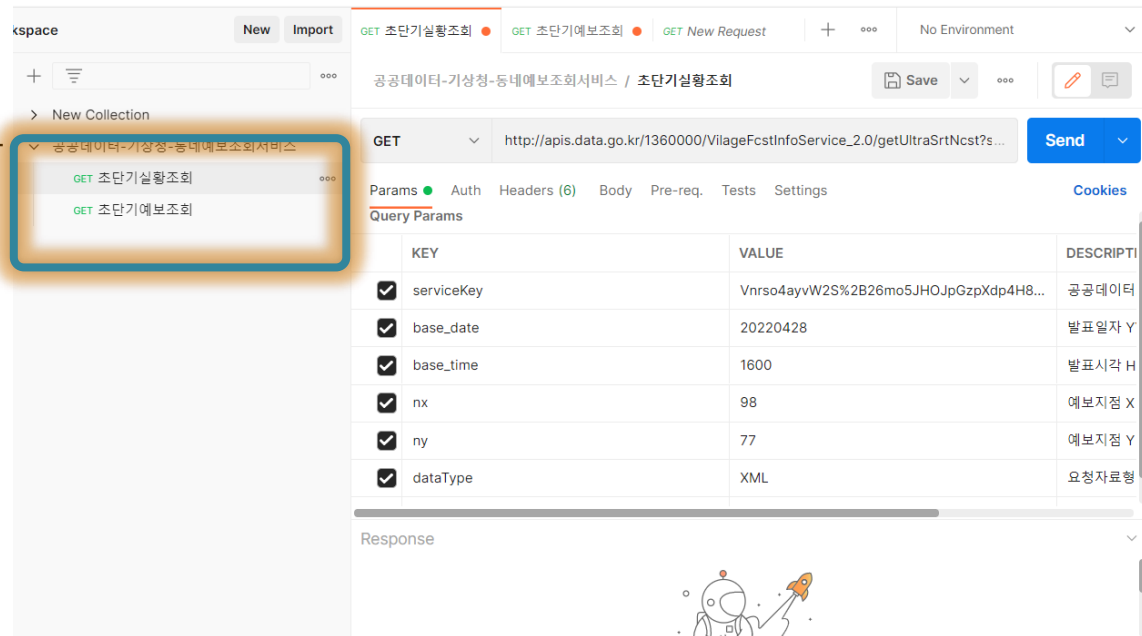
❖ Postman Collection File download & Import



❖ 공공데이터-기상청-단기예보조회서비스 API Test

- 초단기실황조회 API를 Test
- Response를 해석해보자
 - Category “RN1”의 의미는?
- Query Params의 의미를 이해하고 값을 변경하며 Test해보자
- Nx와 ny를 자신이 거주하는 지역으로 변경해보자
- 인증키를 자신이 할당 받은 인증키로 변경해보자

- 참고) 공공데이터 포털에서의 “활용 신청”이 성공하여도 API 활용이 즉각적으로 되지 않을 수 있음. 인증 정보가 실제 API 서버에 등록되기까지 어느 정도의 시간이 소요됨



공공 API 호출 Test - Postman 활용

❖ dataType이 “XML”인 경우와 “JSON”인 경우의 Response를 비교해보기

- 본 강좌에서는 가능한 “JSON” 형식을 활용하여 실습할 예정임

❖ 공공데이터-부산버스정보시스템 API를 Test 해보기

- 본인이 자주 이용하는 정류장의 버스 도착 정보를 획득해보자

❖ 공공데이터-기상청-단기예보조회 API의 다른 서비스 Test

- Collection에 포함된 “초단기실황조회”, “초단기예보조회” 기능 외에 “단기예보조회” 서비스를 호출하는 호출문을 API 문서를 참고하여 작성해보기
 - 자기가 사는 지역을 기반으로 parameter 작성 (나머지 parameter는 제한없음) 후, API 동작 확인

네이버/Kakao 개발자 계정과 이용 Key 얻기

❖ 네이버 계정 생성하기 → <http://developer.naver.com>

- API 이용 신청: <https://developers.naver.com/apps/#/register>

❖ Kakao 계정 생성하기 → <https://developers.kakao.com/>

- 내 APP 추가: <https://developers.kakao.com/console/app>

The image displays two web portals side-by-side. The left portal is 'kakao developers' with a dark blue theme. It features a large banner with the text 'Build with Kakao Developers' and a '시작하기' (Get Started) button. Below the banner, there are sections for '추천 제품' (Recommended Products) including '카카오싱크', '카카오 로그인', and '카카오톡 소셜'. The right portal is 'NAVER Developers' with a white theme. It has a navigation bar with links like 'Products', 'Documents', and 'Application'. Below the navigation bar, there are several columns of links categorized under 'API 이용 안내', 'CLOVA', '네이버 로그인', '파파고', and '서비스 API'. At the bottom of the Naver portal, there is a section titled '사전 검수 신청 방법' (Pre-review Application Method) with two bullet points: '✓ 적용 가이드 확인 후 개발' and '✓ 필수 확인 항목과 사전 검수 요청 가이드 확인 후 검수 등록'. Below the portals, there is a row of icons for various services: CLOVA, 네이버 로그인, papago, 서비스 API, and 네이버 클라우드 플랫폼.

kakao developers 내 애플리케이션 제품 문서 도구 포럼 로그인 Q KOR ENG

Build with Kakao Developers

시작하기

추천 제품

카카오싱크 카카오 로그인 카카오톡 소셜

NAVER Developers Products Documents Application NAVER D2 Support Forum API 상태 Search Here 로그인

API 이용 안내	CLOVA	네이버 로그인	파파고	서비스 API
API 소개		네이버 로그인 API	Papago 번역	데이터랩
운영 정책		카페	언어 감지	검색
FAQ		캘린더	한글인명-로마자 변환	단축URL
BI 가이드		네이버페이 배송지 정보		웹차
이용약관		적용 가이드		네이버 공유하기
상표사용 가이드		적용 사례		모바일앱 연동
				네이버 오픈메인

사전 검수 신청 방법

- ✓ 적용 가이드 확인 후 개발
- ✓ 필수 확인 항목과 사전 검수 요청 가이드 확인 후 검수 등록

products/intro/plan/plan.md CLOVA 네이버 로그인 papago 서비스 API 네이버 클라우드 플랫폼

Naver/Kakao 검색 API 호출 Test

❖ PLATO에서 Naver/Kakao 검색 API Collection File download & Import

- Query Parameter “query”의 값을 바꿔가며 검색 API를 Test해보기

The screenshot displays the PLATO API client interface. On the left, a sidebar shows a collection named 'Naver_Kakao Search API' with two items: 'GET Naver Search OpenAPI' and 'GET KAKAO Daum Search OpenAPI'. The main panel shows the details of the 'GET Naver Search OpenAPI' endpoint. The URL is 'https://openapi.naver.com/v1/search/webkr.json?query=BIFF'. The 'Query Params' section is highlighted, showing a table with a checked checkbox for the 'query' parameter with the value 'BIFF'. The 'Body' section shows the JSON response in 'Pretty' format.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> query	BIFF	검색을 원하는 문자열로서 UTF-8로 ...

```
1 {
2   "lastBuildDate": "Fri, 29 Apr 2022 15:27:09 +0900",
3   "total": 838224,
4   "start": 1,
5   "display": 10,
6   "items": [
7     {
```

Naver/Kakao의 인증 정보

❖ Naver/Kakao API 호출에는 인증 정보가 필요하지 않은가?

- 공공데이터 API를 호출 하려면 Service Key 정보가 필요하였음
 - serviceKey 라는 인자 이름에 대해 자신의 Service Key 값을 입력
- 취약점 : Service Key 정보가 URL에 그대로 노출됨

❖ Query Parameter가 아닌 HTTP Request Header를 통한 인증 정보 전달

- Params가 아닌 “Headers” Tab을 선택해보기
 - Naver의 경우 X-Naver-Client-Id, X-Naver-Client-Secret이라는 두 개의 Header {Key, Value}
 - Kakao의 경우 Authorization이라는 하나의 Header {Key, Value}
- 본인의 인증 정보로 변경해보기

The screenshot shows a REST client interface with the following details:

- URL:** `https://openapi.naver.com/v1/search/webkr.json?query=BIFF`
- Method:** GET
- Headers Tab:** Selected and highlighted with a blue box. It shows 8 headers, with 6 hidden.
- Header List:**

	KEY	VALUE	DE	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	X-Naver-Client-Id	HH2fJNFYyJKtPz_WyAl8				
<input checked="" type="checkbox"/>	X-Naver-Client-Secret	TymPEM45jw				
	Key	Value				Description
- Status Bar:** 200 OK, 224 ms, 1.73 KB
- Response Format:** JSON

HTTP API를 이용한 Thing 제어 : Philips HUE

❖ HUE made by philips

❖ 구성요소

- 브릿지
- 전구
- 어플리케이션



❖ What we do?

- 상황에 따른 색깔 및 밝기의 연출
 - 분위기에 따른 연출
 - 사용자의 행동패턴, 혹은 외부의 신호에 따른 연출
 - 예) 사용자가 집에 들어오고 나감을 감지하여 조명 제어
이메일, 문자 등이 왔음을 조명을 통하여 알림
 - 예술적 표현
 - 예) 미리 프로그래밍된 제어 시퀀스를 통하여 많은 수의 전구를 제어하는 예술표현

<http://www2.meethue.com/ko-kr/>

<http://www.developers.meethue.com/>

Philips HUE 이용 예시

❖ YouTube Example

- <https://www.youtube.com/watch?v=v-R3otgPZNI>
- <https://www.youtube.com/watch?v=PH4djvY4Vqo>

❖ 사용자의 위치에 따른 HUE 전구 제어 (Geo-fencing)



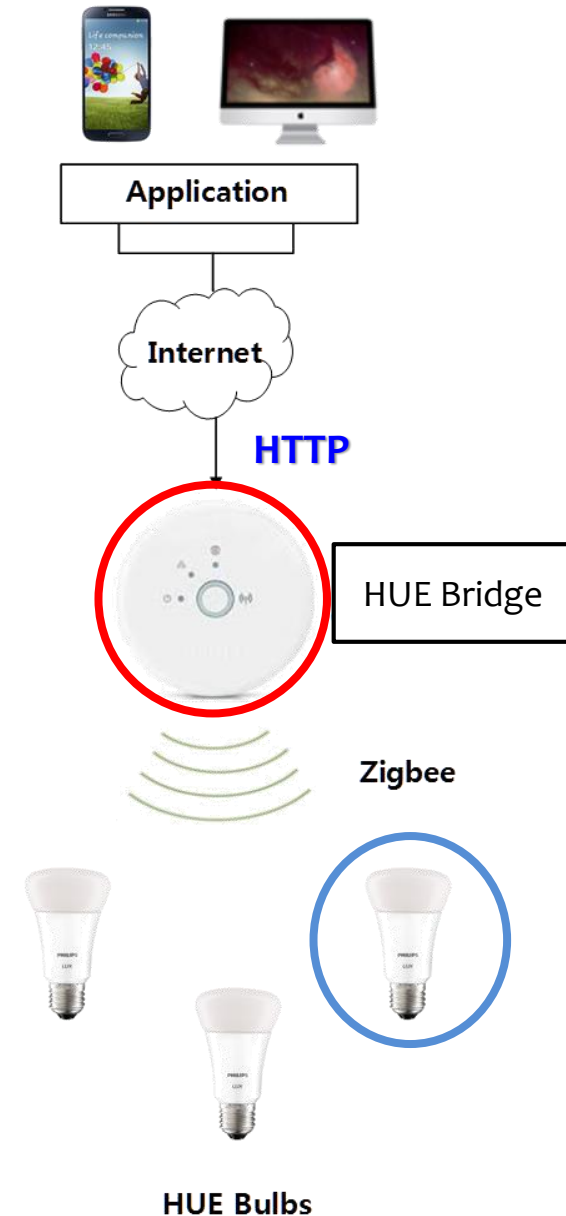
Philips Hue API

❖ API 문서 접근을 위해서는 사용자 등록이 필요

- <http://www.developers.meethue.com/philips-hue-api>

❖ Lights API

- Hue Bridge : <test ip> - 192.168.1.16
- Get All lights
 - URL : /api/<username>/lights **GET**
 - <username> : ZV3BGkv7HMGKFwlBvQmd3osl5SndQUgNNCXqritl
 - No Parameters
- Get light attributes and state
 - URL : /api/<username>/lights/<id> **GET**
 - No Parameters



Philips Hue API

❖ Lights API : Set Light State

- URL : /api/<username>/lights/<id>/state **PUT**
 - <username> : ZV3BGkv7HMGKFwlBvQmd3osI5SndQUgNNCXqritl
 - No Parameters
- JSON Body !! → Header 필요 : Content-Type : application/json

HTTP PUT



UPDATE

▶ http://192.168.0.104/api/ZV3BGkv7HMGKFwlBvQmd3osI5SndQUgNNCXqritl/lights/1/state

PUT ▾	http://192.168.0.104/api/ZV3BGkv7HMGKFwlBvQmd3osI5SndQUgNNCXqritl/lights/1/state	Params	Send ▾
Authorization	Headers (1)	Body ●	Pre-request Script Tests

<input checked="" type="checkbox"/>	Content-Type	application/json	⋮ ×	Bulk Edit
	key	value		

▶ http://192.168.0.104/api/ZV3BGkv7HMGKFwlBvQmd3osI5SndQUgNNCXqritl/lights/1/state

PUT ▾	http://192.168.0.104/api/ZV3BGkv7HMGKFwlBvQmd3osI5SndQUgNNCXqritl/lights/1/state	Params	Send ▾
Authorization	Headers (1)	Body ●	Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary **JSON (application/json)** ▾

1 {"on" : true}

Automatic Turn On/Off

The screenshot displays the 'Collection Runner' application window. The top bar includes 'File', 'Edit', 'View', and 'Help' menus. Below the bar, the 'Run Results' tab is active, showing a summary for 'Automatic On/Off' with '0 PASSED' and '0 FAILED' results. The test suite is titled 'Automatic On/Off' and is associated with 'No Environment'. Buttons for 'Run Summary', 'Export Results', 'Retry', and 'New' are visible. The test results table shows 'Iteration 1' with three 'PUT' requests: 'Turn off No.1', 'Turn off No.2', and 'Turn off No.3'. Each request is marked as '200 OK' with a response time of '18 ms' and a body size of '42 B'. The table also indicates that 'This request does not have any tests.' for each entry. On the right side of the table, there is a vertical bar with a red '1' indicating the total number of failed tests.

Iteration	Method	Request	URL	Response	Time	Size
Iteration 1	PUT	Turn off No.1	http://192.168.0.104/api/...	Automatic On/Off / Turn off No.1	200 OK	18 ms
					42 B	
		This request does not have any tests.				
	PUT	Turn off No.2	http://192.168.0.104/api/...	Automatic On/Off / Turn off No.2	200 OK	18 ms
					42 B	
		This request does not have any tests.				
	PUT	Turn off No.3	http://192.168.0.104/api/...	Automatic On/Off / Turn off No.3	200 OK	18 ms
					42 B	
		This request does not have any tests.				
	PUT	Turn On No.1	http://192.168.0.104/api/...	Automatic On/Off / Turn On No.1	200 OK	18 ms
					42 B	
		This request does not have any tests.				
	PUT	Turn On No.2	http://192.168.0.104/api/...	Automatic On/Off / Turn On No.2	200 OK	18 ms
					42 B	
		This request does not have any tests.				
	PUT	Turn On No.3	http://192.168.0.104/api/...	Automatic On/Off / Turn On No.3	200 OK	18 ms
					42 B	
		This request does not have any tests.				

The photograph shows a desk setup with a white router, two red heat lamps, and a black desk lamp. The router is positioned in the center, with two red heat lamps on either side. A black desk lamp is on the right. The desk is covered with a red cloth. The background is a wooden wall.