

# 공공API 활용가이드

## API 신청

### 홈페이지 접속

DATA 공공데이터포털  
.GO.KR

로그인 회원가입 사이트맵 ENGLISH

데이터셋

제공신청

활용사례

정보공유

이용안내

### 로그인

로그인

아이디/비밀번호찾기

공인인증서등록/안내

로그인 > 로그인

### 로그인



국민과 함께 하는 공공데이터포털에 오신 것을 환영합니다.

회원가입을 하시면 공공데이터포털을 통해 국가가 보유하고 있는 유용한 공공데이터와 오픈API를 활용하실 수 있습니다.

- 개인회원 로그인 페이지입니다.
- ID/PW 방식이나 인증서 방식 중 한 가지로 로그인이 가능합니다.
- 공공데이터포털은 개인정보보호법 및 공공데이터 개인정보처리방침에 의거하여 회원가입 후 2년이 경과된 회원에 대하여 개인정보수집에 대한 재동의를 받고 있습니다. 미동의 시, 포털 관리자에 의해 회원탈퇴 및 회원정보가 삭제 처리될 수 있습니다. 아이디/비밀번호를 정상적으로 입력했으나 로그인이 되지 않는 경우, 관리자에 의해 회원 정보가 삭제된 경우이므로 불편하시더라도 회원 재가입을 부탁드립니다.
- 개인정보보호 관련 법제

<https://www.data.go.kr> 홈페이지에 로그인을 합니다.

### 경락가격정보서비스로 접속

## 경락가격정보서비스

ENGLISH

경락가격정보를 조회하기 위한 서비스로서 농산물, 축산물, 수산물, 화훼류, 산지공판장, 종합유통센터별로 조회할 수 있다.

활용신청 (바로가기) 건수 : 217

※ 서비스 오류가 있을시 오류신고 버튼을 이용해주세요.

XML

경락가격정보서비스



활용신청



닫기



오류신고



<https://www.data.go.kr/dataset/15000355/openapi.do>

활용신청을 누릅니다.

### API 신청서 제출

## > 시스템유형 선택 ?

☒ 일반    ☐ 서버 구축

\* 시스템 유형
 

- \* 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

테스트용이기 때문에 일반을 선택하며

## > 활용정보

* 활용목적	<input checked="" type="radio"/> 웹 사이트 개발 <input type="radio"/> 앱개발 (모바일,솔루션등) <input type="radio"/> 기타 <input type="radio"/> 참고자료 <input type="radio"/> 연구(논문 등)	
첨부파일	한 개의 파일만 첨부 할 수 있습니다. <div> <input type="button" value="파일 선택"/> <input type="button" value="선택된 파일 없음"/> <input type="button" value="삭제"/> </div>	

활용목적에 연구를 선택한 후 신청을 완료합니다.

## API키 확인

 총 1 건

[승인] 경락가격정보서비스      신청일:2018-09-27 [활용신청] 만료예정 :2020-09-27  
 서비스유형 :REST      분류 :교통및물류 > 물류등기타      제공기관 :농림수산식품교육문화정보원

활용이 되었다면 하단처럼 리스트에 노출이 되며

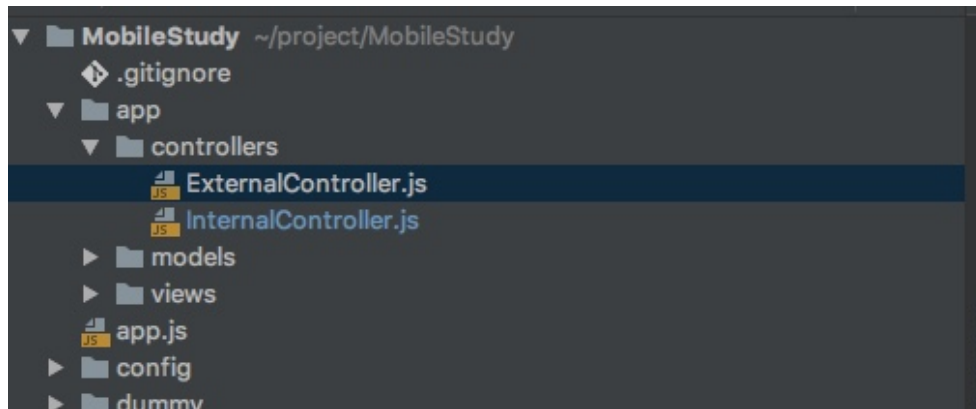
기본정보					
서비스명	경락가격정보서비스		상세설명		
서비스 유형	REST	일일트래픽	0	평균응답속도(초)	0
심의여부	자동승인	신청유형	운영계정   활용신청	처리상태	승인
활용기간	2018-09-27 ~ 2020-09-27				

활용정보가 표시됩니다. 이때 API키를 복사하여줍니다.

# 서버구축

서버는 크게 2가지로 나뉘게 됩니다. 첫번째가 나의 운영서버에서 신청한 후 디비에 저장되는 방식 하나는 디비에 저장하지 않은채 바로 결과를 가져오는 방식입니다.

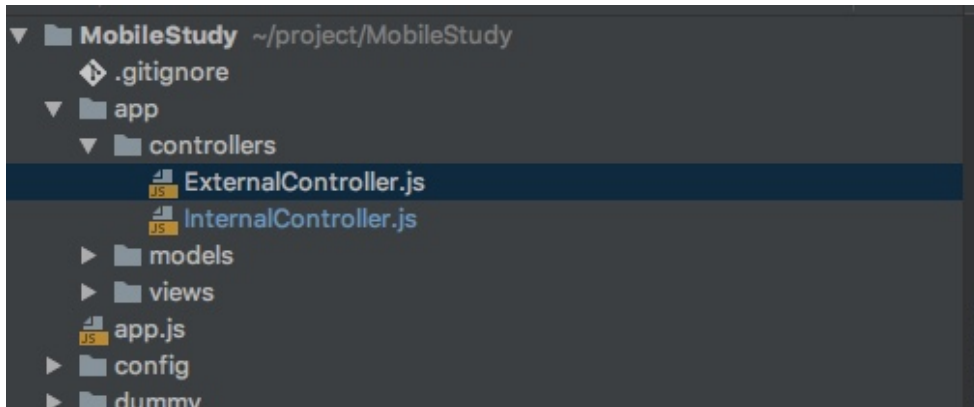
## 직접 가져오는 방식



ExternalController.js파일을 확인합니다.

```
1 module.exports.result = async function (req, res) {
2
3   //request.js는 Restful을 사용하기위한 라이브러리이며 JQuery.Ajax와 크게 다르지 않습니다.
4
5
6   //async await 패턴은 C++의 yeild와 같이 콜백이 끝날때까지 다음 내용을 실행하지 않고 기다립니다.
7   const marketResult = await request.get({
8     url: API_URL,
9     timeout: 10000,
10    json:true,
11    qs: {
12      // 서비스에 발급받은 Service Key를 요청합니다.
13      'ServiceKey': SERVICEKEY,
14      //Dates에 날짜를 입력합니다.
15      'dates': "20180829",
16      //locde와 mcdoe는 농산물 상품코드이며 파일에 함께 첨부드리었습니다.
17      'lcode': req.query.mcode.substring(0,2),
18      'mcode': req.query.mcode,
19      //리턴받을 형식은 json으로 가져옵니다.
20      '_type': "json",
21    },
22  }).then((result)=>{
23    /*데이터를 성공적으로 가져온경우 필요한 item 리스트만 가져옵니다.*/
24    return result.response.body
25  }).catch(e =>{
26    /* 에러처리*/
27    console.error("request Error : "+e)
28  });
29
30
31
32  //리턴받은 데이터를 handlebar.hbs를 통해 출력하여줍니다.
33  res.render("external/result", {
34    fruit:getNameByCode(req.query.mcode),
35    marketResult,
36    query:req.query,
37    pageTitle
38  });
```

## 디비에 저장한 후 저장값을 읽는 방식



InternalController.js 파일을 확인합니다.

```

1  const marketResult = await request.get({
2      url: API_URL,
3      timeout: 10000,
4      json: true,
5      qs: {
6          'ServiceKey': SERVICEKEY,
7          'dates': req.body.date.split("-").join(' '), //정규식을 사용하려 하였지만 이편이 조금 더 이
해가 쉬울것같아 split과 join으로 사용하였습니다.
8          'lcode': req.body.mcode.substring(0, 2),
9          'mcode': req.body.mcode,
10         'numOfRows': req.body.count,
11         '_type': "json",
12     },
13 }).then((result) => {
14     /*데이터를 성공적으로 가져온경우 필요한 item 리스트만 가져옵니다.*/
15     return result.response.body
16 }).catch(e => {
17     /* 에러처리*/
18     console.error("request Error : " + e);
19     res.status(500).send();
20 });
21
22
23 /* 사용자에게 받은 mcode(상품카테고리)와 리턴받은 mcode(상품카테고리)가 일치하는
24 데이터만을 필터링합니다. */
25 let result = marketList.filter(data => data.mcode == req.body.mcode);
26
27 /* 0보다 많을경우 데이터베이스에 넣어줍니다.*/
28 /* 그렇지 않을경우 데이터가 없다는 값을 리턴합니다. */
29 if(marketResult.totalCount > 0){
30     /* 리턴받은 값을 데이터베이스에 저장하여줍니다. */
31     new MarketHistory({
32         fruit:getNameByCode(req.body.mcode),
33         date:req.body.date,
34         result:marketResult.items.item
35     }).save(err =>{
36         if(err) res.status(500).send();
37         /* 데이터가 있는 경우 204 Empty Success 메시지를 보내줍니다. */
38         else res.status(204).send() //

```

```
39         })
40     }else{
41         /* 검색된 결과가 0개인경우*/
42         /* 이경우 200 status값과 함께 리턴 메시지를 보내줍니다. */
43         res.status(200).send("데이터가 없습니다.");
44     }
45 }
```