

공공API 사용 우편번호 검색

문서 작성일: 2024-3-26

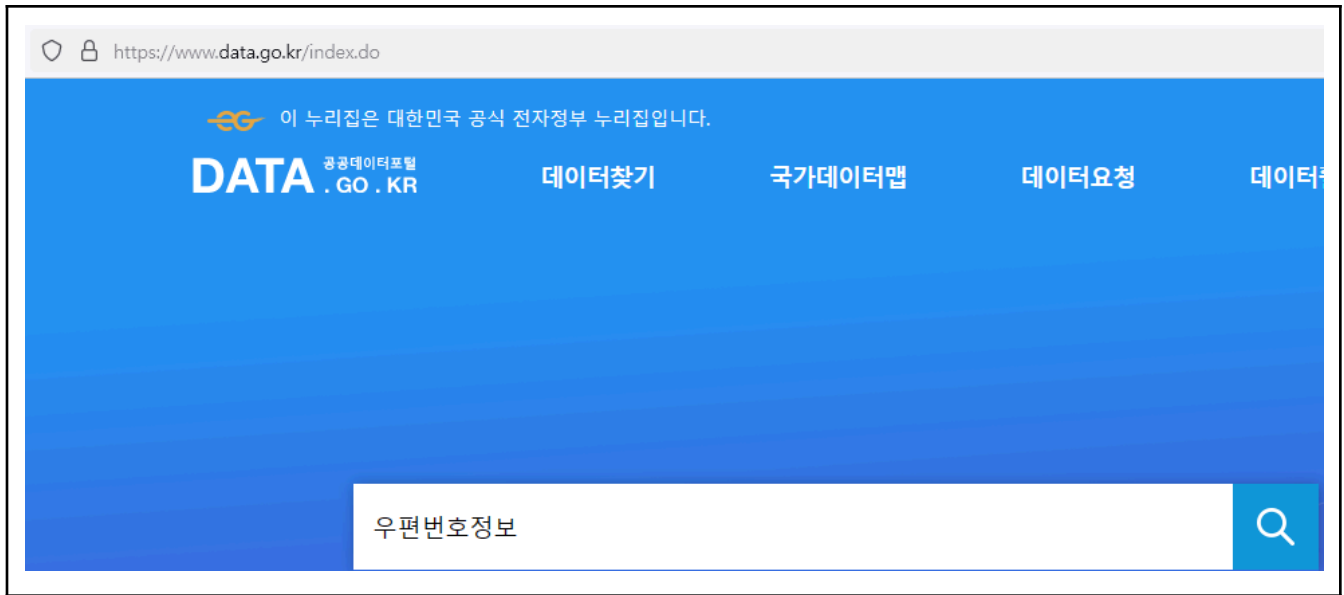
[참고자료-파랑새](#)

목차

- [1. 인증키 확보](#)
- [2. 우편번호 검색화면 설계](#)
 - [a. 봄 부트 프로젝트 생성](#)
 - [f. 주소 검색어 입력 상자](#)
 - [j. 검색 결과 모의 결과 반환](#)
 - [o. 페이지화 기술 별도 개발](#)
 - [g. 주소 목록 페이지화](#)
- [3. 자바 코드 작성](#)
 - [b. 인증키 읽히는지 확인](#)
 - [d. JUnit 시험 메소드에서 작동 확인](#)
 - [f. find, FoundAddress 목록 구성](#)
- [4. 검색 화면을 find\(\)와 연계](#)
 - [a. find\(\) 포함 클래스 코드 개선](#)
 - [b. 주소 페이지 반환 서비스 클래스](#)
 - [c. paging 단편 개선](#)
 - [d. searchform 코드 개선](#)
 - [e. 주소 페이지 서비스 개선](#)
 - [f. 주소 검색 제어기 수정](#)
 - [g. 불필요한 파일 제거](#)
- [5. 화면 개선 - 지번주소](#)

1. 인증키 확보

- a. [사이트](#) 방문
- b. 로그인 후 휴대폰 인증까지해야 됨.
- c. 첫 페이지에서 "우편번호정보" 입력 후 돋보기 아이콘 클릭



- d. 검색 결과 나오면
 - i. 오픈 API 탭 선택 ⇒ *** 우편번호 정보조회 활용신청

"우편번호정보"에 대해 총 555건이 검색되었습니다.

조건검색 초기화

분류체계	서비스유형	제공기관유형	확장자
------	-------	--------	-----

국가중점데이터 분류 조건 추가하기 + 조건열기

전체(555건)

파일데이터(505건)

오픈 API(48건)

표준데이터셋0개(0건)

연계데이터(2건)

정확도순

10개씩

정렬

오픈 API (48건)

교통물류

국가행정기관

미리보기

XML

과학기술정보통신부 우정사업본부_우편번호 정보조회

출력하는 항목(사용자가 응답 받게 되는 데이터) : 우편번호, 도로명주소, 지번주소

제공기관 과학기술정보통신부 우정사업본부 수정일 2021-08-17 조회수 14620 활용신청 4993 키워드 경제/경영,마케팅,물류/유통관리

활용신청

e. 중간 결과

제공기관	과학기술정보통신부 우정사업본부	서비스유형	REST
심의여부	자동승인	신청유형	개발계정 활용신청
처리상태	신청	활용기간	승인일로부터 24개월 간 활용가능

f. 활용목적 입력

*활용목적

☒ 웹 사이트 개발 ☐ 앱개발 (모바일,솔루션등) ☐ 기타 ☐ 참고자료 ☐ 연구(논문 등)

저는 학원 강사 지망생으로 추후 학생들이 주소 검색 기술 개발 실습용 교재 개발을 위하여 요청합니다

g. (아마도 걸어놓은 제한치)

상세기능정보 선택

<input checked="" type="checkbox"/>	상세기능	설명	일일 트래픽
<input checked="" type="checkbox"/>	새우편번호 정보	도로명주소 및 지번주소를 페이지당 출력될 개수와 출력될 페이지에 맞춰 5자리 우편번호와 도로명주소 및 지번주소의 상세내역을 조회해주는 우편번호 정보조회 서비스	10000

라이선스 표시

*이용허락범위

이용허락범위 제한 없음

☒ 동의합니다.

h. 이런게 뜹니다.

www.data.go.kr

신청이 완료되었습니다. 신청된 API는 1~2시간 후 호출 가능합니다.

※ 일부 API에서 24시간 후 인증이 가능한 API가 있습니다.
빠른 시일내에 모든 API가 1시간 이내 인증 가능하도록 개선하겠습니다

OK

총 1건

교통물류

과학기술정보통신부 우정사업본부

활용신청 [승인] 과학기술정보통신부 우정사업본부_우편번호 정보조회

계정 개발 신청일 2024-03-26 만료예정일 2026-03-26

- i. 이제 내일 아침에 이 프로그램 개발하면 되겠군요.
- j. 그런데 인증키는 어디 있는 거지?
- k. 아, 두시간 뒤에 로그인해서 마이페이지로 갔더니 다음 섹션이 생겼습니다.

개인 API인증키

더보기 >

구분	발급일자	인증키
일반	2024-03-26	sroyq3Om/1qtYOjSzX... 인증키 복사(Enc oding) 인증키 복사(Dec oding)

2.우편번호 검색화면 설계

a. bom 부트 프로젝트 생성

- i. 이름: zip_search
- ii. 종속성: 타입리프트, 롬복, 웹, devtools
 - [commit](#)

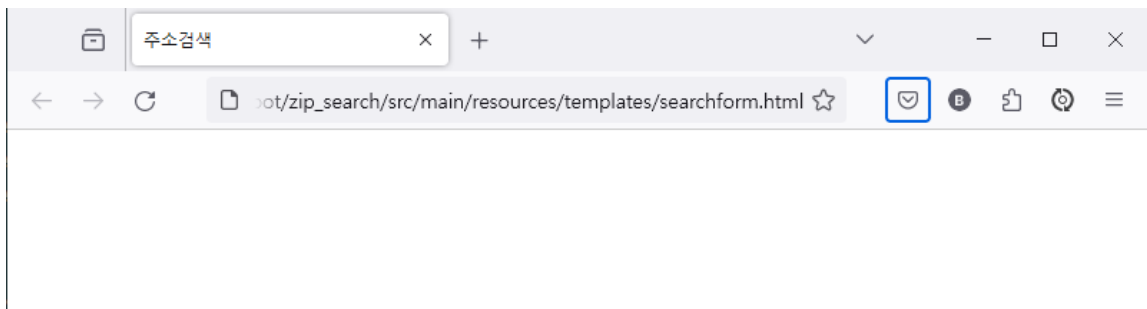
b. searchform.html 생성

- i. 위치: resources/templates
- ii. 내용:

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4 <meta charset="UTF-8">
5 <title>주소검색</title>
6 </head>
7 <body>
8
9 </body>
10 </html>
```

- [commit](#)

c. 브라우저로 확인



d. 제어기 생성

- i. 앞서, 기본 패키지 명칭 변경
 - 으로: `space.bum.zip_search`
 - [commit](#)
- ii. 제어기 클래스 생성
 - 패키지: `space.bum.zip_search.controller`
 - 이름: `SearchController`
 - 내용:

```

6 @Controller
7 public class SearchController {
8     @GetMapping("/search")
9     public String addressSearchForm() {
10         return "searchform";
11     }
12 }

```

- [commit](#)

e. 타임리프 작동 시험

i. html 조금 변경

```

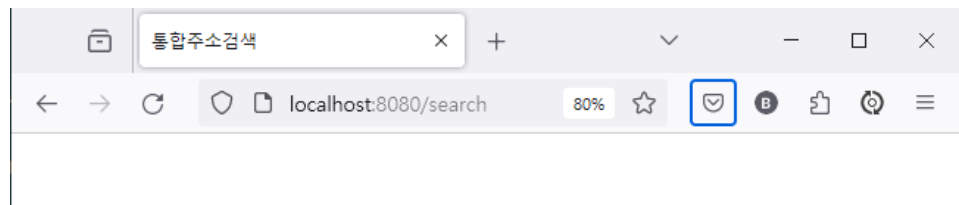
1 <!DOCTYPE html>
2 <html lang="ko" xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <meta charset="UTF-8">
5 <title th:text="'통합주소검색'">주소검색</title>
6 </head>
7 <body>
8
9 </body>
10 </html>

```

- [commit](#)

ii. 실행 및 폼 요청

- <http://localhost:8080/search>



f. 주소 검색어 입력 상자

- i. 일단, 부트스트랩 CSS 및 자바스크립트 코드를 CDN 에서 수입한다.

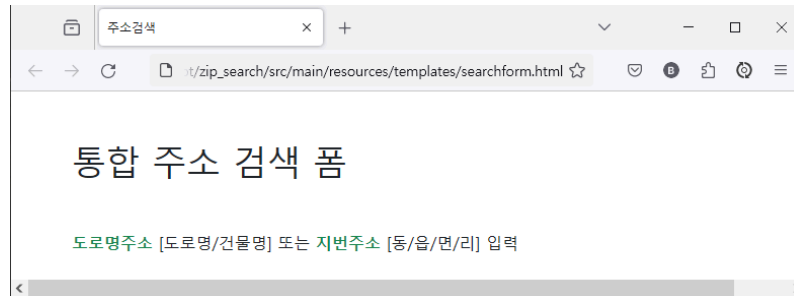
```

4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link
7   href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
8   rel="stylesheet">
9 <script
10   src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
11 <title th:text="'통합주소검색'">주소검색</title>

```

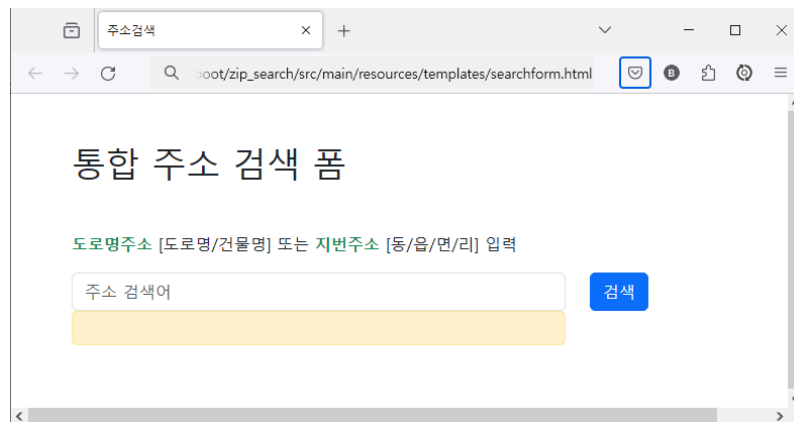
- [commit](#)

ii. 다음 브라우저로 열었을 때 다음의 형태가 되도록 코드를 추가한다.



- [commit](#)

iii. 다음과 같이 "주소 검색어"를 입력하는 상자를 포함하는 폼을 html 에 추가한다.



- [commit](#)

iv. 검색어 입력값 검증 종속성 추가 - pom.xml


```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
```

- [commit](#)

- v. 위 폼에서 입력하는 문자열을 저장하는 도메인 클래스 `SearchKey` 를 정의한다.

```
7 @Data
8 public class SearchKey {
9     @NotEmpty
10    @Size(min = 3, max = 20)
11    private String addrKey;
12 }
```

- [commit](#)

- vi. 검색 폼 반환 제어기 메소드를 정의한다.

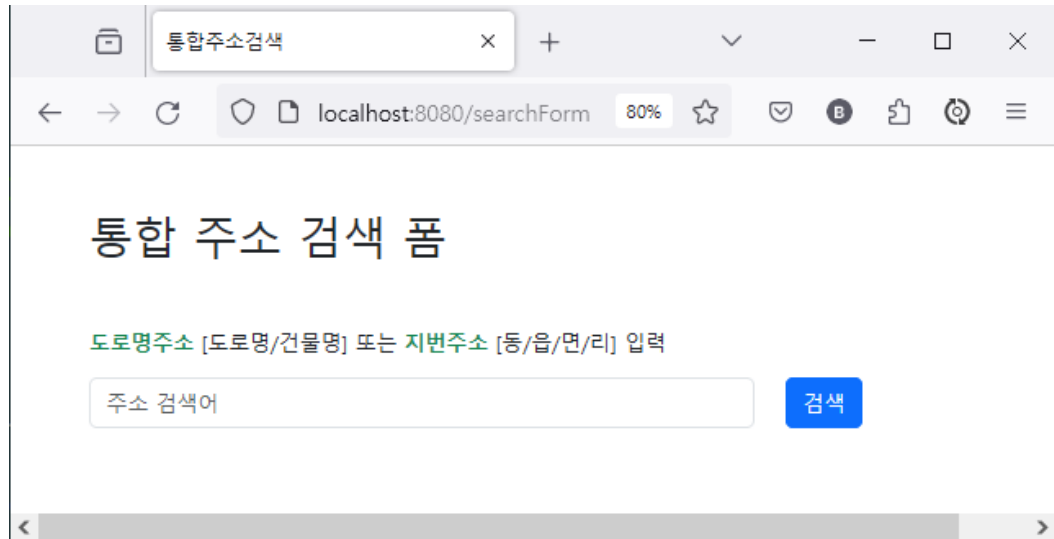
- 종점(endpoint) 명칭 변환 및 모델에 검색 키 객체 추가

```
11 @GetMapping("/searchForm")
12 public String addressSearchForm(Model model) {
13     model.addAttribute("searchKey", new SearchKey());
14     return "searchform";
15 }
```

- [commit](#)

- vii. 봄 부트 및 타립을 통한 검색 폼 호출

- 실행 및 방문 주소 - <http://localhost:8080/searchForm>
- 결과



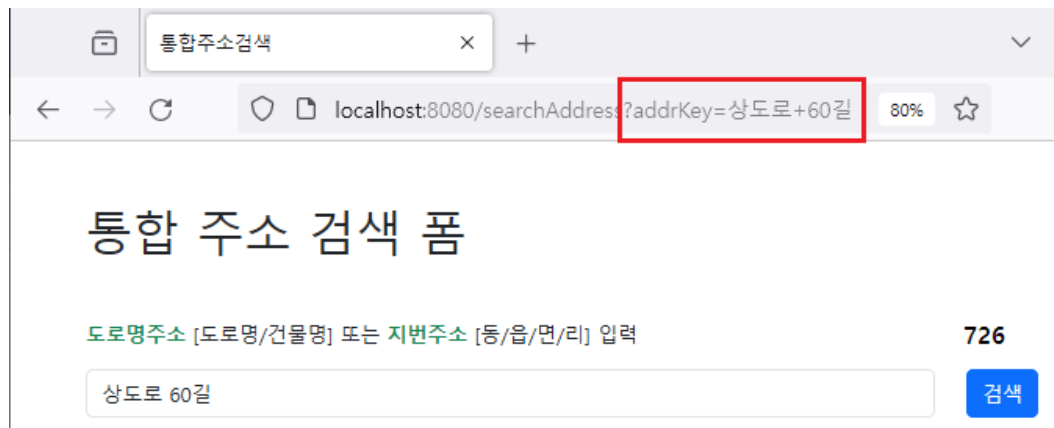
g. 검색 처리 종점 메소드 구현

```
19 @GetMapping("/searchAddress")
20 public String addressSearchForm(@Valid SearchKey searchKey,
21     BindingResult result, Model model) {
22     System.out.println("주소 검색어: " + searchKey.getAddrKey());
23     return "searchform";
24 }
```

- [commit](#)

h. 검색어 입력 및 요청 전송 상자

- 실행, 주소방문, 키 입력 및 검색 버튼 클릭
- 방문 주소: <http://localhost:8080/searchForm>
- 검색 키: 상도로 60길



- 주소반환 페이지(searchAddress 에 "?addrKey=상도로+60길" 이 표시되는 것을 원치 않으면, form 태그의 속성 중 method="post" 로 수정하면 된다.

```
<form class="form-horizontal" th:object="${searchKey}"  
      th:action="@{/searchAddress}" method="post">
```

```
20 @PostMapping("/searchAddress")  
21 public String addressSearchForm(@Valid SearchKey searchKey,
```

- [commit](#)

통합주소검색

통합 주소 검색 폼

도로명주소 [도로명/건물명] 또는 지번주소 [동/읍/면/리] 입력

강남대로 123길

검색

- i. 검색 결과 저장 도메인 클래스 생성

```
6 @Data  
7 @AllArgsConstructor  
8 public class FoundAddress {  
9     private String zipcode;  
10    private String address;  
11 }
```

- [commit](#)

j. 검색 결과 모의 결과 반환

```
26     System.out.println("주소 검색어: " + searchKey.getAddrKey());
27     var addresses = new ArrayList<FoundAddress>();
28
29     createDummyAddresses(addresses);
30     model.addAttribute("addresses", addresses);
31     System.out.println("검색 건수: " + addresses.size());
32     return "searchform";
33 }
34
35 private void createDummyAddresses(ArrayList<FoundAddress> addresses) {
36     addresses.add(new FoundAddress("12345", "서울시 종로구 관철동 134번지"));
37     addresses.add(new FoundAddress("23233", "경기도 하남시 망월동 1050번지"));
38 }
```

- [commit](#)

k. 주소 검색 요청 결과 표시 콘솔 출력 확인

```
주소 검색어: 강남대로 123길
검색 건수: 2
```

l. 주소 검색 모의 결과 표출 - 다음과 같이 '아무거나' 입력하고 검색 요청 때 아래 그림과 같이 출력되게 템플릿을 구성하라.

(검색건수: 2)

우편번호 주소(도로명/지번)

12345	서울시 종로구 관철동 134번지
23233	경기도 하남시 망월동 1050번지

```

39< <table class="table table-striped w-auto caption-top">
40<   <caption class="ps-2">
41<     (검색건수:
42<     <span
43<       th:text="${addresses eq null} ? '- )' : ${addresses.size} + ')">
44<       (검색건수)</span>
45<   </caption>
46<   <thead>
47<     <tr>
48<       <th>우편번호</th>
49<       <th>주소(도로명/지번)</th>
50<     </tr>
51<   </thead>
52<   <tbody>
53<     <tr th:each="address: ${addresses}">
54<       <td th:text="${address.zipcode}">(우편번호)</td>
55<       <td th:text="${address.address}">(도로명/지번-주소)</td>
56<     </tr>
57<     <tr th:remove="all">
58<       <td>(우편번호)</td>
59<       <td>(도로명/지번-주소)</td>
60<     </tr>
61<     <tr th:remove="all">
62<       <td>(우편번호)</td>
63<       <td>(도로명/지번-주소)</td>
64<     </tr>
65<   </tbody>
66< </table>

```

[commit](#)

- m. 처리기 클래스 사적 메소드 헤딩 변경

```

35< private void createDummyAddresses(List<FoundAddress> addresses) {

```

[commit](#)

- n. 오류 메시지 - 입력 안하고 검색 결과

must not be empty

size must be between 3 and 20

- 영어로 표시되니 이를 한글화 필요.

- o. resources 폴더에 messages.properties 파일을 만들고 다음 내용을 입력할 것.

```
messages.properties ×
1 searchKey.addrKey=주소 검색어
2 NotEmpty.searchKey.addrKey=`{0}`를 입력하세요
3 Size.searchKey.addrKey=`{0}` 입력 범위는 {2}에서 {1} 문자입니다.
```

- [commit](#)
- 실행 결과

도로명주소 [도로명/건물명] 또는 지번주소 [동/읍/면/리] 입력

검색

주소 검색어를 입력하세요

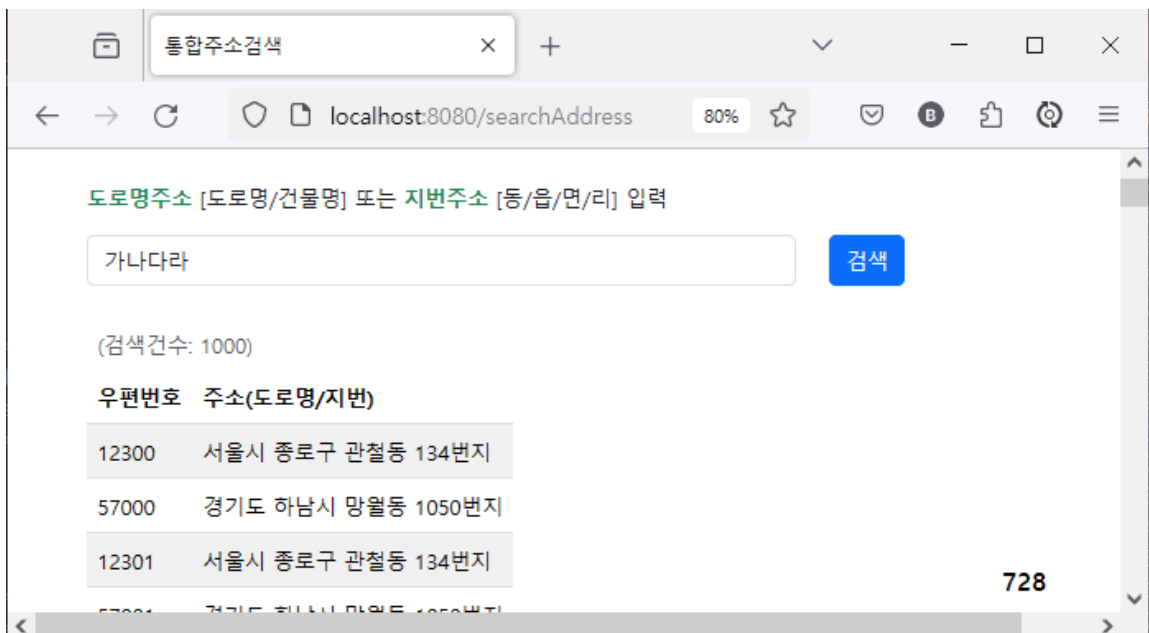
주소 검색어 입력 범위는 3에서 20 문자입니다.

p. 검색 결과 표시 페이지화 구현

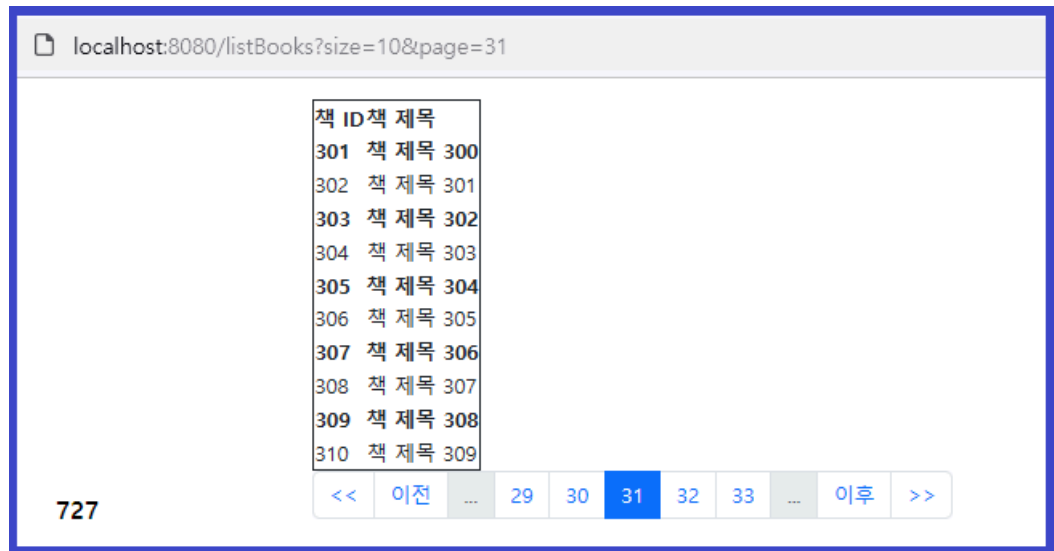
- i. 일단, 모의 주소를 1000건 생성하고 이를 뷰로 보낸다.

```
35 private void createDummyAddresses(List<FoundAddress> addresses) {
36     for (int i = 0; i < 500; i++) {
37         addresses.add(new FoundAddress(Integer.toString(12300 + i),
38             "서울시 종로구 관철동 134번지"));
39         addresses.add(new FoundAddress(Integer.toString(57000 + i),
40             "경기도 하남시 망월동 1050번지"));
41     }
42 }
```

- [commit](#)



- 위에 보다시피 1000건이 검색되어 내용이 한없이 많아 우측 스크롤바가 작아진 것을 볼 수 있다. 이는 바람직하지 않고 다음과 같이 적은 수만 표출하고 아래 페이지링크 목록을 횡으로 제공하는 개발이 필요하다.



○ 페이지화 기술 별도 개발

- [깃허브 - pagination](#)
- [readme.txt 파일](#) // 참고한 두 문서 링크 포함.

q. 주소 목록 페이지화

- 일단, 필요한 종속성을 추가한다.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jdbc</artifactId>
</dependency>
```

- [commit](#)

- h2 디비 종속성을 추가하라

- [commit](#)

- h2 구성 설정 // application.yml 파일 사용

- [commit](#)

- iv. 유용물 클래스 생성
 - AddressUtils.buildAddresses()
 - [commit](#)

- v. 페이지 반환 서비스 빈 클래스
 - AddressService 클래스
 - [commit](#)

- vi. 주소 검색 요청 처리 메소드
 - 서비스 빈 호출
 - GET 방식 요청 접수 // 페이지 크기 및 번호 주소에 표시되는 것도 나쁘지 않음
 - 뷰에서 필요로하는 속성 값 전달
 - [commit](#)

- vii. 페이지 요청 링크 단편 정의
 - 세 군데 이상 재사용되는 단편
 - [commit](#)

- viii. 뷰에 페이지 향해 링크 생성
 - 기본 페이지 크기 : 5
 - 하단 페이지 링크 수: 5
 - [commit](#)

- ix. 표출된 모의 주소 검색 결과



3. 자바 코드 작성

- a. 파랑새의 코드를 복사하여 클래스를 만든다.
 - i. 클래스 이름: `OpenApiController`
 - ii. 소스코드

```

15 public class OpenApiController {
16     // * 공공데이터포털(http://www.data.go.kr) 오픈 API 이용
17
18     // 서비스명 : 통합검색 5자리 우편번호 조회서비스
19     // 새 우편번호(2015-08-01부터) 오픈 API 주소
20     // http://openapi.epost.go.kr/postal/retrieveNewAddressAreaCdSearchAllService/re
21
22     // [in] s : 검색어 (도로명주소[도로명/건물명] 또는 지번주소[동/읍/면/리])
23     // [in] p : 읽어올 페이지(1부터), l : 한 페이지당 출력할 목록 수(최대 50까지)
24     // [out] v[i*3 +0]=우편번호, v[i*3 +1]=도로명주소, v[i*3 +2]=지번주소, v.Count/3=표시할 목록 수
25     // [out] n[0]=검색한 전체 목록(우편번호) 개수, n[1]=읽어온 페이지(1부터)
26     // 반환값 : 예러메시지, null == OK
27     public static String find(String s, int p, int l, List<String> v, int[] n) {
28         HttpURLConnection con = null;
29
30         try {
31             URL url = new URL(
32                 "http://openapi.epost.go.kr/postal/retrieveNewAddressAreaCdSearchAllServ
33                 + "?ServiceKey=발급받은_인증키" // 서비스키
34                 + "&countPerPage=" + l // 페이지당 출력될 개수를 지정(최대 50)
35                 + "&currentPage=" + p // 출력될 페이지 번호
36                 + "&srchwrds=" + URLEncoder.encode(s, "UTF-8") // 검색어
37             );

```

- [commit](#)

iii. 코드의 다음 항목 수정

- 33행 - 발급받은_인증키 (아래 상자 값)

홈 > 마이페이지 > 데이터 활용 > Open API > 인증키 발급현황

인증키 발급현황

총 1건

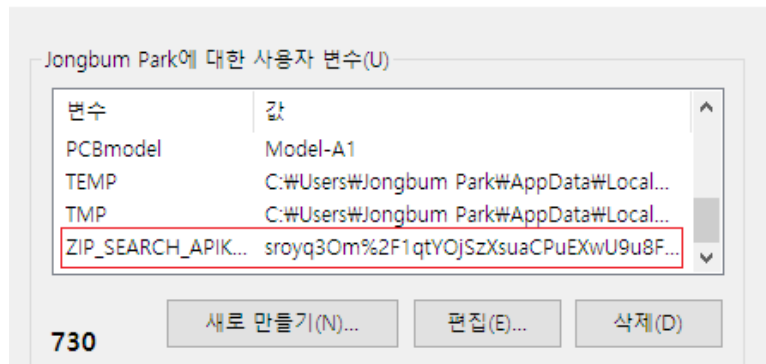
구분	일반
발급일자	2024/03/26
재발급여부	신규발급
인증키	8DKLr50 [redacted]

Z
729

- 단, 인증키는 비밀이므로 이를 숨기는 것은 다음 참고 문서를 따른다. 즉,
 - [API 키 숨기는 방법](#)

(1) [사용자] 환경 변수 ZIP_SEARCH_APIKEY 를 다음과 같이 정의한다.

환경 변수



(2) 다음, 이를 find 메소드에서 사용한다.

```
29 String auth_key = System.getenv("ZIP_SEARCH_APIKEY");
30
31 try {
32     URL url = new URL(
33         "http://openapi.epost.go.kr/postal/retrieveNewAddressAreaCdSe
34         + "?ServiceKey="
35         + auth_key // 서비스키
36         + "&countPerPage=" + 1 // 페이지당 출력될 개수를 지정(최대 50)
37         + "&currentPage=" + p // 출력될 페이지 번호
38         + "&srchwrds=" + URLEncoder.encode(s, "UTF-8") // 검색어
39     );
```

* 제 29 및 35 행 참조

■ [commit](#)

b. 인증키 읽히는지 확인

- 다음 JUnit 시험 메소드를 만들어 호출한다. 그리고 위 29 행을 디버거로 잡고 키가 읽히는지 시험한다.

```
8 class OpenApiControllerTest {
9
10     @Test
11     void testFind() {
12         List<String> vector = new ArrayList<String>();
13         int[] counts = new int[2];
14         OpenApiController.find("미사강변동로", 1, 10, vector, counts);
15     }
16
17 }
```

- [commit](#)

```

29 String auth_key = System.getenv("ZIP_SEARCH_APIKEY");
30
31 try {
32     URL u
33
34     /postal/retrieveNewAd
35

```

auth_key= "sroyq3Om%2F1qtYOjSz"

- coder= 0
- hash= 0
- hashIsZero= false
- value= (id=87)

731

c. find 메소드 작동 확인

- i. 그 전에 Postman 으로 다음 설정하여 자료 검색 시험.
- ii. 주소:

http://openapi.epost.go.kr/postal/retrieveNewAdressAreaCdSearchAllService/retrieveNewAddressAreaCdSearchAllService/getNewAddressListAreaCdSearchAll?srchwrд=미사강변서로127&countPerPage=10¤tPage=1

iii. 인자 : 3 개

GET
http://openapi.epost.go.kr/postal/retrieveNewAdressAreaCdSearchAll...
Send

Params
Auth
Headers (8)
Body
Pre-req.
Tests
Settings
Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	srchwrд	미사강변서로127			
<input checked="" type="checkbox"/>	countPerPage	10			
<input checked="" type="checkbox"/>	currentPage	1			

iv. 인증

- 단, 아래 키 Value('sroyq...')은 : 디코딩된 인증키를 사용함

Params
Auth
Headers (8)
Body
Pre-req.
Tests
Settings
Cookies

Type

API Key

The authorization header will be automatically generated when you send the request. Learn more about [API Key](#) authorization.

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables.](#)

Key

ServiceKey

Value

sroyq3Om/1qtYOjSzXsuaCPuEXv...

Body
200 OK 182 ms 3.39 KB Save as example

디코딩된 인증키

개인 API인증키				더보기 >	
	구분	발급일자	인증키		
732	일반	2024-03-26	sroyq3Om/1qtVOjSzX...	인증키 복사(Enc oding)	인증키 복사(Dec oding)

v. 결과

Body ▾ 200 OK 182 ms 3.39 KB Save as example ...

Pretty Raw Preview Visualize XML ▾

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <NewAddressListResponse>
3   <cmmMsgHeader>
4     <requestMsgId></requestMsgId>
5     <responseMsgId></responseMsgId>
6     <responseTime>20240329:121617294</responseTime>
7     <successYN>Y</successYN>
8     <returnCode>00</returnCode>
9     <errMsg></errMsg>
10    <totalCount>10</totalCount>
11    <countPerPage>10</countPerPage>
12    <totalPage>1</totalPage>
13    <currentPage>1</currentPage>
14  </cmmMsgHeader>
15  <newAddressListAreaCdSearchAll>
16    <zipNo>12915</zipNo>
17    <lnmAdres>경기도 하남시 미사강변서로 127 (망월동, 미사강변도시18단지)
18      1801동~1817동</lnmAdres>
19    <rnAdres>경기도 하남시 망월동 1050 (미사강변도시18단지)</rnAdres>
20  </newAddressListAreaCdSearchAll>
  </newAddressListAreaCdSearchAll>

```

d. JUnit 시험 메소드에서 작동 확인

- testFind 메소드를 일부 수정하고, 15행에서 멈춘 뒤, vector 값을 확인하면
- [commit](#)

```

10 @Test
11 void testFind() {
12     List<String> vector = new ArrayList<String>();
13     int[] counts = new int[2];
14     var result = OpenApiController.find("미사강변동로", 1, 10, vector, counts);
15     System.out.println("결과: " + result);
16 }

```

- 아래와 같이 3 요소가 한 주소 항목을 구성하고 있음을 알 수 있다. 총 30행, 10개 주소 항목이 반환된 것이다.

```
var result = OpenApiController.find("미사강변동로", 1, 10, vector, counts);
System.out.println("vector: " + vector);
vector= ArrayList<E> (id=202)
> ▲ [0]= "12912" (id=201)
> ▲ [1]= "경기도 하남시 미사강변동로 지하 90 (망월동, 미사역)" (id=206)
> ▲ [2]= "경기도 하남시 망월동 1122 (미사역)" (id=212)
> ▲ [3]= "12924" (id=215)
> ▲ [4]= "경기도 하남시 미사강변동로 19 (풍산동, 신안인스빌) 3201동~3208동" (id=216)
> ▲ [5]= "경기도 하남시 풍산동 573 (신안인스빌)" (id=217)
> ▲ [6]= "12924" (id=227)
[12912, 경기도 하남시 미사강변동로 지하 90 (망월동, 미사역), 경기도 하남시 망월동 112
```

- e. 위 vector 를 가지고 주소 객체 목록을 만들려면 일단 FoundAddress 를 다음과 같이 수정해야 된다.

```
7  @Data
8  @AllArgsConstructor
9  @ToString
10 public class FoundAddress {
11     private String zipcode;
12     /**
13      * 도로명 주소
14      */
15     private String roadAddress;
16     /**
17      * 지번 주소
18      */
19     private String zBunAddress;
20 }
```

[commit](#)

- f. find, FoundAddress 목록 구성

```
29 public static String find(String s, int p, int l,
30     List<FoundAddress> pageAddresses, int[] n) {
```

```

82         if (bOk) {
83             ns = doc.getElementsByTagName("newAddressListAreaCdSearchAll");
84             for (p = 0; p < ns.getLength(); p++) {
85                 nd = ns.item(p).getFirstChild();
86                 String zipcode = nd.getTextContent();
87                 nd = nd.getNextSibling();
88                 String roadAddress = nd.getTextContent();
89                 nd = nd.getNextSibling();
90                 String zBunAddress = nd.getTextContent();
91                 pageAddresses
92                     .add(new FoundAddress(zipcode, roadAddress, zBunAddress));
93             }
94         }

```

[commit](#)

- g. 현재, URL 문자열 구성때 문자열 '+' 연산을 과도하게 사용하여 비효율적이므로 이를 `StringBuilder` 를 사용하여 효율화한다.

```

31     HttpURLConnection con = null;
32     | var urlBuilder = new StringBuilder("http://openapi.epost.go.kr/");
33
34     urlBuilder.append("postal/retrieveNewAddressAreaCdSearchAllService/");
35     urlBuilder.append("retrieveNewAddressAreaCdSearchAllService/");
36     urlBuilder.append("getNewAddressListAreaCdSearchAll");
37     urlBuilder.append("?ServiceKey=");
38     urlBuilder.append(System.getenv("ZIP_SEARCH_APIKEY")); // API 인증키
39     urlBuilder.append("&countPerPage=");
40     urlBuilder.append(1); // 페이지당 출력될 개수를 지정(최대 50)
41     urlBuilder.append("&currentPage=");
42     urlBuilder.append(p); // 출력될 페이지 번호
43     urlBuilder.append("&srchwrds=");
44
45     try {
46         urlBuilder.append(URLEncoder.encode(s, "UTF-8")); // 주소 검색어
47         URL url = new URL(urlBuilder.toString());
48
49         con = (HttpURLConnection) url.openConnection();

```

[commit](#)

- h. 바뀐 `find` 메소드에 맞게 시험 메소드를 수정

```

12 @Test
13 void testFind() {
14     List<FoundAddress> pageAddresses = new ArrayList<FoundAddress>();
15     int[] counts = new int[2];
16     OpenApiController.find("미사강변서로", 6, 10, pageAddresses, counts);
17     System.out.println("결과: " + pageAddresses);
18 }

```

[commit](#)

i. 실행 결과 콘솔 출력

```
결과: [FoundAddress(zipcode=12915, roadAddress=경기도 하남시 미사강변서로 127-4 (망월동,
미사강변센텀팰리스(CentumPalace)), zBunAddress=경기도 하남시 망월동 1050
(미사강변센텀팰리스(CentumPalace))), FoundAddress(zipcode=12915, roadAddress=경기도 하남시
미사강변서로 127-4 (망월동, 미사강변도시18단지), zBunAddress=경기도 하남시 망월동 1050
(미사강변도시18단지)), FoundAddress(zipcode=12915, roadAddress=경기도 하남시 미사강변서로 151
(망월동, 윤솔초등학교), zBunAddress=경기도 하남시 망월동 1046 (윤솔초등학교)), FoundAddress
(zipcode=12915, roadAddress=경기도 하남시 미사강변서로 167 (망월동, 미사강변도시17단지),
zBunAddress=경기도 하남시 망월동 1044 (미사강변도시17단지)), FoundAddress(zipcode=12915,
roadAddress=경기도 하남시 미사강변서로 167-1 (망월동, 미사강변도시17단지), zBunAddress=경기도
하남시 망월동 1044 (미사강변도시17단지)), FoundAddress(zipcode=12915, roadAddress=경기도
하남시 미사강변서로 167-2 (망월동, 미사강변도시17단지), zBunAddress=경기도 하남시 망월동 1044
(미사강변도시17단지)), FoundAddress(zipcode=12915, roadAddress=경기도 하남시 미사강변서로 167
-3 (망월동, 미사강변도시17단지), zBunAddress=경기도 하남시 망월동 1044 (미사강변도시17단지)),
FoundAddress(zipcode=12915, roadAddress=경기도 하남시 미사강변서로 167-4 (망월동,
미사강변도시17단지), zBunAddress=경기도 하남시 망월동 1044 (미사강변도시17단지))]
```

4. 검색 화면을 find()와 연계

a. find() 포함 클래스 코드 개선

- 클래스명: ZipcodeUtil
- find 메소드 형식인자명칭 개선: s ⇒ addrSearchKey 등

```
17 public class ZipcodeUtil {
18     /**
19      * 공공데이터포털(http://www.data.go.kr) 오픈 API 이용 서비스명 : 통합검색 5자리 우편번호
20      * 조회서비스 새 우편번호(2015-08-01부터) 오픈 API 주소
21      *
22      * @param addrSearchKey : 주소 검색어 (도로명주소[도로명/건물명] 또는 지번주소[동/읍/면/리])
23      * @param pageIndex : 읽을 페이지 번호(1: 첫 페이지)
24      * @param pageSize : 페이지 크기(최대 50)
25      * @param pageAddresses : 읽은 주소(우편번호, 도로주소, 지번주소) 한 페이지
26      * @param result : 검색 결과[전체 주소항 개수, 읽은 페이지 번호]
27      * @return 오류 메시지, 성공한 경우 null
28      */
29     public static String find(String addrSearchKey, int pageIndex, int pageSize,
30                               List<FoundAddress> pageAddresses, int[] result) {
31         HttpURLConnection con = null;
```

- [commit](#)

b. 주소 페이지 반환 서비스 클래스

```
18 @Service
19 public class ZipcodeService {
20
21     public Page<FoundAddress> findPaginated(@Valid SearchKey searchKey,
22         Pageable pageable, int[] counts) {
23         int pageSize = pageable.getPageSize();
24         int currentPage = pageable.getPageNumber();
25         List<FoundAddress> pageAddresses = new ArrayList<FoundAddress>();
26
27         ZipcodeUtil.find(searchKey.getAddrKey(), currentPage + 1, pageSize,
28             pageAddresses, counts);
29         Page<FoundAddress> addressPage = new PageImpl<FoundAddress>(pageAddresses,
30             PageRequest.of(currentPage, pageSize, counts[0]));
31
32         return addressPage;
33     }
34 }
```

- [commit](#)

c. paging 단편 개선

- 주소 검색어 인자 추가 및 활용

```
<a th:fragment="paging(addrKey, size, page, label, tooltip)" th:text=${label}
th:class="page-link"
th:href="@{searchAddress(addrKey=${addrKey}, size=${size}, page=${page})}"
th:title="${tooltip}" rel="tooltip"> [[${label}]] </a>
```

- [commit](#)

d. searchform 코드 개선

- paging 단편 호출 때, 검색어 인자 추가

```
73     <a th:replace="~{fragments/paging :: paging(${searchKey.addrKey},
74         ${addressPage.size}, 1, '<<', '첫 페이지')}"></a>
```

- 방법 get 으로...
- 속성명 roadAddress 사용
- [commit](#)

e. 주소 페이지 서비스 개선

```
public Page<FoundAddress> findPaginated(@Valid SearchKey searchKey,
    Pageable pageable) {
    int pageSize = pageable.getPageSize();
    int currentPage = pageable.getPageNumber();
    List<FoundAddress> pageAddresses = new ArrayList<FoundAddress>();
    int[] result = new int[2];
```

- [commit](#)

f. 주소 검색 제어기 수정

- 모의 주소 서비스 대신 우편 번호 서비스 사용

```
31 @Autowired
32 private ZipcodeService zipcodeService;
```

- [commit](#)

g. 불필요한 파일 제거

- [commit](#)

5. 화면 개선 - 지번주소

- 도로명 주소 및 지번 주소 동시 표현

12916	경기도 하남시 미사강변서로 85-3 (풍산동, 미사강변동일하이빌)
	경기도 하남시 풍산동 470 (미사강변동일하이빌)

```

<thead>
  <tr>
    <th>우편번호</th>
    <th>도로명 주소 / <span class="fw-normal">지번 주소</span></th>
  </tr>
</thead>
<tbody th:if="${addressPage != null}">
  <tr th:each="address,iStat: ${addressPage.content}"
      th:if="${addressPage != null}" class="align-middle">
    <td th:text="${address.zipcode}">(우편번호)</td>
    <td>
      <p th:text="${address.roadAddress}" class="mb-0 fw-bold">(도로 주소)</p>
      <p th:text="${address.zBunAddress}" class="mb-0">(지번 주소)</p>
    </td>
  </tr>
  <tr th:remove="all" class="align-middle">
    <td>(우편번호)</td>
    <td>
      <p class="mb-0">(도로 주소)</p>
      <p class="mb-0">(지번 주소)</p>
    </td>
  </tr>
</tbody>

```

- [commit](#)
- 실행 결과

통합주소검색

localhost:8080/searchAddress?addrKey

80%

통합 주소 검색 폼

도로명주소 [도로명/건물명] 또는 지번주소 [동/읍/면/리] 입력

미사강변중앙로

검색

(검색건수: 113)

우편번호 도로명 주소 / 지번 주소

12920	경기도 하남시 미사강변중앙로 143 (풍산동, 대한예수교장로회우리들교회) 경기도 하남시 풍산동 541 (대한예수교장로회우리들교회)
12913	경기도 하남시 미사강변중앙로 157 (망월동, 리버테라스) 경기도 하남시 망월동 1104-2 (리버테라스)
12913	경기도 하남시 미사강변중앙로 160 (망월동, 미사리버파크) 경기도 하남시 망월동 1108-3 (미사리버파크)
12913	경기도 하남시 미사강변중앙로 161 (망월동, 미사파크뷰) 경기도 하남시 망월동 1104-1 (미사파크뷰)
12913	경기도 하남시 미사강변중앙로 164 (망월동, 미타워) 경기도 하남시 망월동 1108-1 (미타워)

<<

직전

...

3

4

5

6

7

...

다음

>>