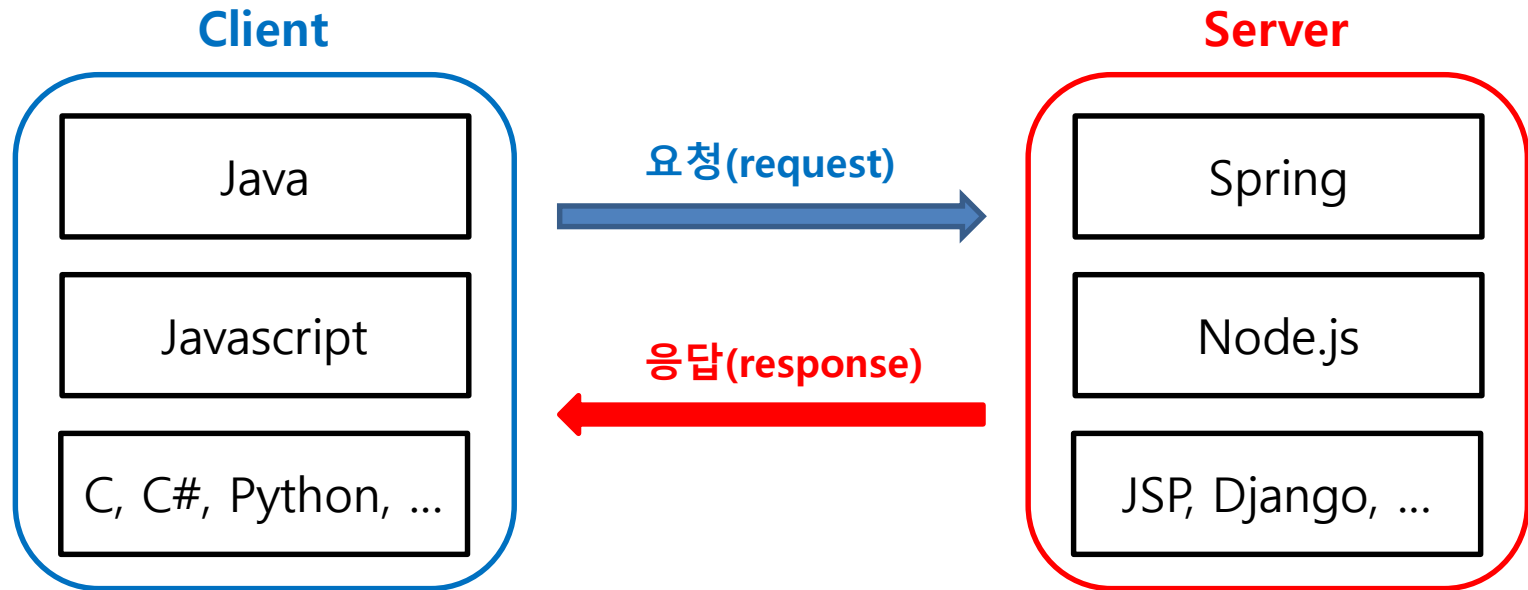


■ HTTP 통신

● 통신 흐름



■ HTTP 통신

● Client 에서 Server 로 요청

```
URL url = new URL("http://ggoreb.com/http/list1.jsp");  
URLConnection urlConnection = url.openConnection();
```

● Server 에서 받은 응답을 문자열(String) 형태로 변환

```
InputStream in = urlConnection.getInputStream();  
InputStreamReader isr = new InputStreamReader(in, "utf-8");  
BufferedReader reader = new BufferedReader(isr);  
  
StringBuffer response = new StringBuffer();  
  
String data = null;  
while(true) {  
    data = reader.readLine();  
    if(data == null) break;  
    response.append(data + "\n");  
}
```

■ HTTP 통신

● 전체 소스코드 (1 / 2)

```
package http;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;

public class Http1 {
    public static void main(String[] args) throws IOException {
        URL url = new URL("http://ggoreb.com/http/list1.jsp");
        URLConnection urlConnection = url.openConnection();

        InputStream in = urlConnection.getInputStream();
        InputStreamReader isr = new InputStreamReader(in, "utf-8");
        BufferedReader reader = new BufferedReader(isr);

        StringBuffer response = new StringBuffer();
```

■ HTTP 통신

● 전체 소스코드 (2 / 2)

```
String data = null;
while(true) {
    data = reader.readLine();
    if(data == null) break;
    response.append(data + "\n");
}
reader.close();
isr.close();
in.close();

System.out.println(response);
}
}
```

■ HTTP 통신

● 브라우저로 요청한 결과

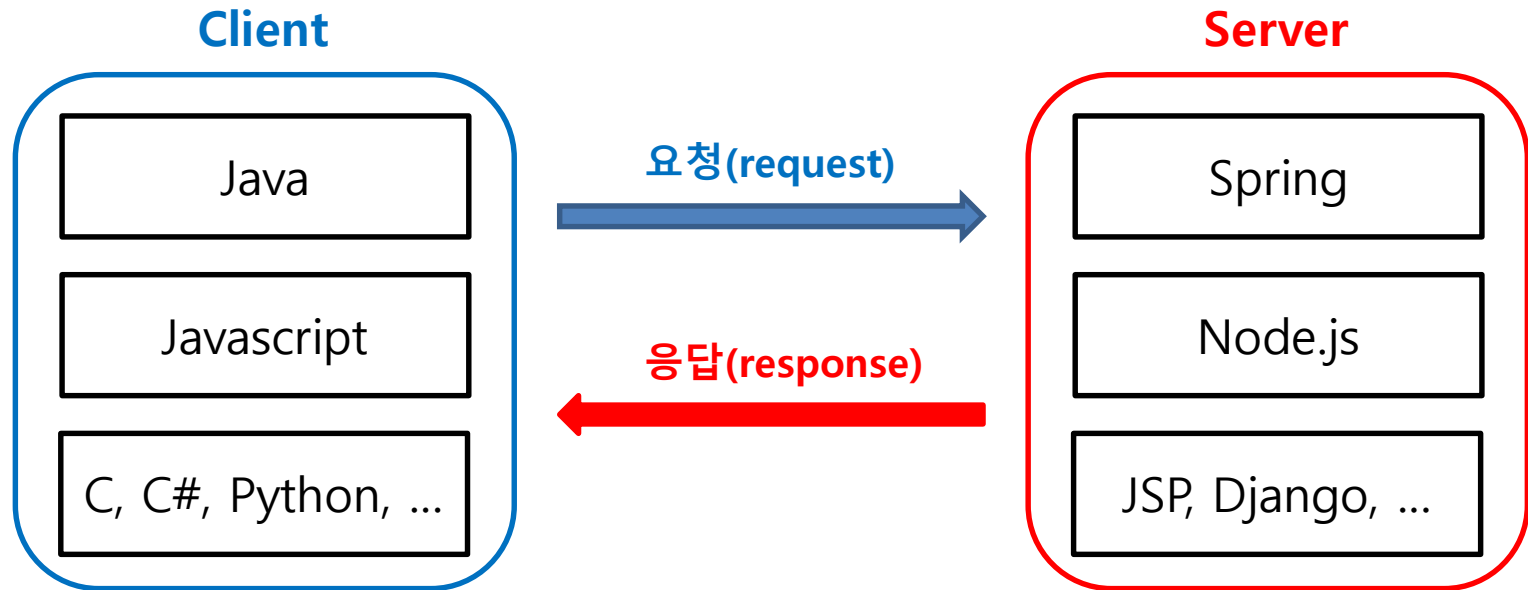


● 자바 코드로 요청한 결과

```
["java", "android", "spring", "html", "javascript"]
```

■ HTTP 통신 (POST)

● 통신 흐름



■ HTTP 통신 (POST)

● Client 에서 Server 로 요청

```
URL url = new URL("http://ggoreb.com/http/list2.jsp");  
URLConnection urlConnection = url.openConnection();  
HttpURLConnection con = (HttpURLConnection) urlConnection;
```

● POST 요청 설정

```
con.setRequestMethod("POST");  
con.setDoOutput(true);
```

● (필요시) 파라미터 입력

```
OutputStream out = con.getOutputStream();  
OutputStreamWriter osr = new OutputStreamWriter(out, "utf-8");  
BufferedWriter writer = new BufferedWriter(osr);
```

■ HTTP 통신 (POST)

● 전체 소스코드 (1 / 3)

```
package http;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;

public class Http2 {
    public static void main(String[] args) throws IOException {
        URL url = new URL("http://ggoreb.com/http/list2.jsp");
        URLConnection urlConnection = url.openConnection();
        HttpURLConnection con = (HttpURLConnection) urlConnection;
```


■ HTTP 통신 (POST)

● 전체 소스코드 (2 / 3)

```
con.setRequestMethod("POST");
con.setDoOutput(true);

OutputStream out = con.getOutputStream();
OutputStreamWriter osr = new OutputStreamWriter(out, "utf-8");
BufferedWriter writer = new BufferedWriter(osr);

writer.close();
osr.close();
out.close();

InputStream in = con.getInputStream();
InputStreamReader isr = new InputStreamReader(in, "utf-8");
BufferedReader reader = new BufferedReader(isr);

StringBuffer response = new StringBuffer();

String data = null;
```

■ HTTP 통신 (POST)

● 전체 소스코드 (3 / 3)

```
while(true) {  
    data = reader.readLine();  
    if(data == null) break;  
    response.append(data + "\n");  
}  
  
reader.close();  
isr.close();  
in.close();  
  
System.out.println(response);  
}  
}
```

■ Data Parsing

● HTML



```
..<!doctype html> == $0
<html>
  <head>...</head>
  <body>
    <div class="info">
      
      
      
      
    </div>
  </body>
</html>
```

● XML

```
<person>
  <number>1</number>
  <name>kim</name>
</person>
```

● JSON

```
{
  "age": 10,
  "name": "GGoReb"
}
```

■ HTML – Jsoup 사용

- Client 에서 Server 로 요청

```
Connection con =  
    Jsoup.connect("http://ggoreb.com/http/html1.jsp");
```

- 응답 받은 HTML 문서를 Document 형식으로 변환

```
Document doc = con.get();
```

- Document 중 모든 img 요소 추출

```
Elements imgs = doc.getElementsByTag("img");
```

- 순서대로 img 요소의 속성값 추출

```
for(int i = 0; i < imgs.size(); i++) {  
    Element img = imgs.get(i);  
    String src = img.attr("src");  
    String title = img.attr("title");  
    System.out.printf("%s, %s\n", src, title);  
}
```

■ HTML – Jsoup 사용

● 전체 소스코드 (1 / 2)

```
package http;

import java.io.IOException;

import org.jsoup.Connection;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class Http3 {
    public static void main(String[] args) throws IOException {
        Connection con =
            Jsoup.connect("http://ggoreb.com/http/html1.jsp");

        Document doc = con.get();

        Elements imgs = doc.getElementsByTag("img");
```

■ HTML – Jsoup 사용

● 전체 소스코드 (2 / 2)

```
for(int i = 0; i < imgs.size(); i++) {  
    Element img = imgs.get(i);  
    String src = img.attr("src");  
    String title = img.attr("title");  
    System.out.printf("%s, %s\n", src, title);  
}  
}  
}
```

```
/images/luffy.jpg, 루피  
/images/nami.jpg, 나미  
/images/robin.jpg, 로빈  
/images/zoro.jpg, 조로
```

■ XML – Jsoup 사용

- Client 에서 Server 로 요청

```
Connection con =  
    Jsoup.connect("http://ggoreb.com/http/data1.xml");
```

- 응답 받은 HTML 문서를 Document 형식으로 변환

```
Document doc = con.get();
```

- Document 중 모든 person 요소 추출

```
Elements persons = doc.getElementsByTag("person");
```

- 순서대로 img 요소의 속성값 추출

```
for(int i = 0; i < persons.size(); i++) {  
    Element person = persons.get(i);  
    Element number = person.getElementsByTag("number").get(0);  
    Element name = person.getElementsByTag("name").get(0);  
    System.out.printf("%s, %s\n", number.text(), name.text());  
}
```

■ XML – Jsoup 사용

● 전체 소스코드 (1 / 2)

```
package http;

import java.io.IOException;

import org.jsoup.Connection;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class Http4 {
    public static void main(String[] args) throws IOException {
        Connection con =
            Jsoup.connect("http://ggoreb.com/http/data1.xml");

        Document doc = con.get();

        Elements persons = doc.getElementsByTag("person");
    }
}
```


■ XML – Jsoup 사용

● 전체 소스코드 (2 / 2)

```
for(int i = 0; i < persons.size(); i++) {  
    Element person = persons.get(i);  
    Element number = person.getElementsByTag("number").get(0);  
    Element name = person.getElementsByTag("name").get(0);  
    System.out.printf("%s, %s\n", number.text(), name.text());  
}  
}
```

1, kim

■ JSON – Json 사용

● Client 에서 Server 로 요청

```
NetworkUtil nu = new NetworkUtil();  
String result = nu.get("http://ggoreb.com/http/json1.jsp");
```

● 응답 받은 JSON 문자열을 JSONObject로 변환

```
JSONObject obj = new JSONObject(result); ← {}로 시작하는 경우
```

● key를 이용하여 value 추출

```
int age = obj.getInt("age"); ← 숫자 getInt(key)  
String name = obj.getString("name"); ← 문자 getString(key)  
System.out.printf("%d, %s", age, name);
```

■ JSON – Json 사용

● NetworkUtil.java (1 / 2)

```
package http;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;

public class NetworkUtil {
    public String get(String address) {
        StringBuffer result = new StringBuffer();
        try {
            URL url = new URL(address);
            URLConnection con = url.openConnection();

            InputStream is = con.getInputStream();
            InputStreamReader isr = new InputStreamReader(is, "utf-8");
            BufferedReader reader = new BufferedReader(isr);
```

■ JSON – Json 사용

● NetworkUtil.java (2 / 2)

```
String separator = "";

while (true) {
    String data = reader.readLine();
    if (data == null) {
        break;
    }
    result.append(separator + data);
    separator = "\\n";
}
} catch (IOException e) {
    e.printStackTrace();
}
return result.toString();
}
}
```

■ JSON – Json 사용

● 전체 소스코드

```
package http;

import org.json.JSONObject;

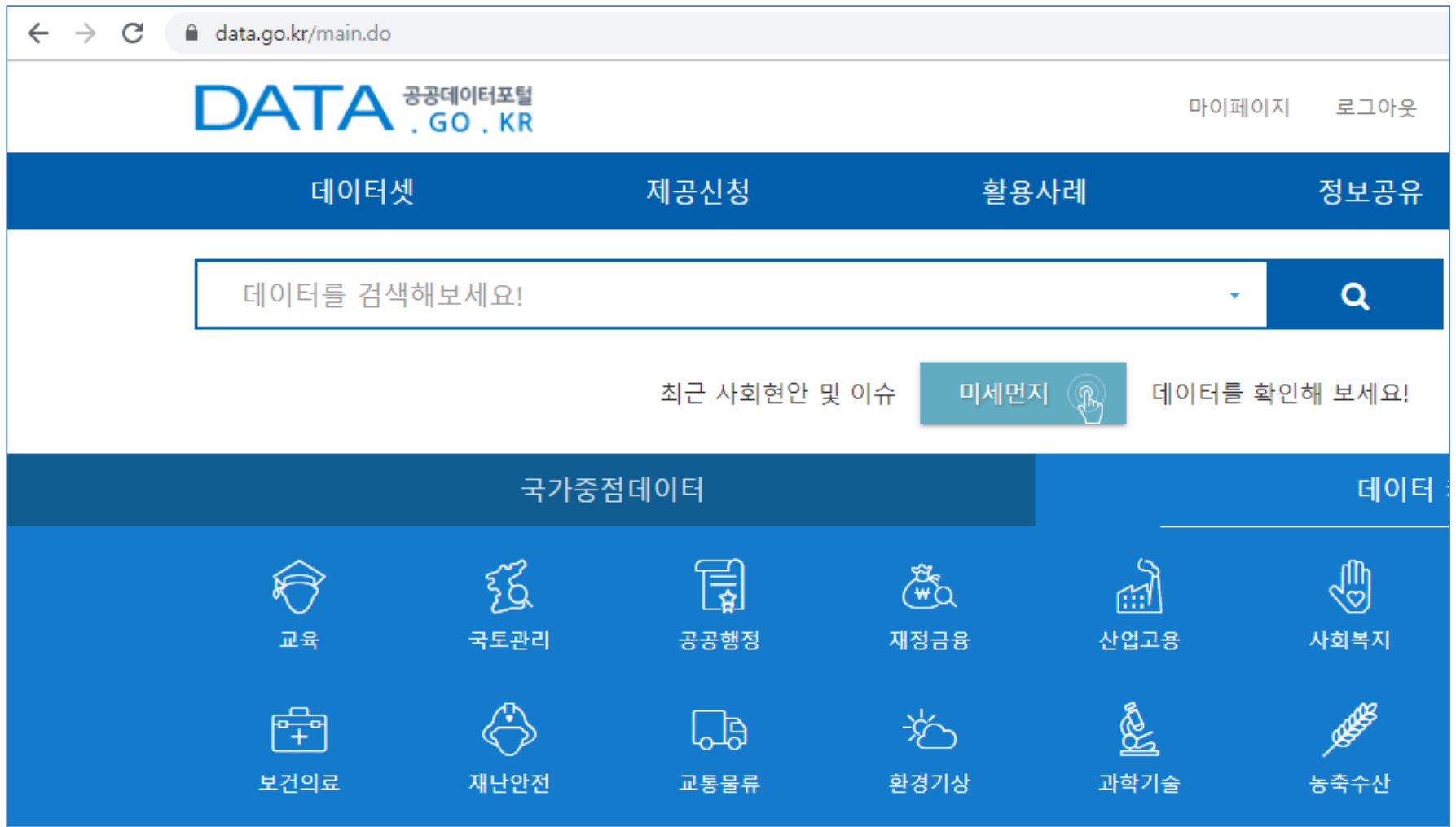
public class Http5 {
    public static void main(String[] args) {
        NetworkUtil nu = new NetworkUtil();
        String result = nu.get("http://g goreb.com/http/json1.jsp");

        JSONObject obj = new JSONObject(result);
        int age = obj.getInt("age");
        String name = obj.getString("name");
        System.out.printf("%d, %s", age, name);
    }
}
```

■ 공공데이터 (data.go.kr) 활용

● 공공데이터 포털 접속 (회원가입 필요)

- <http://data.go.kr>



The screenshot shows the homepage of the Korea Public Data Portal (data.go.kr). The browser address bar displays "data.go.kr/main.do". The page header includes the "DATA . GO . KR" logo and links for "마이페이지" (My Page) and "로그아웃" (Logout). A blue navigation bar contains the following menu items: "데이터셋" (Data Set), "제공신청" (Request for Provision), "활용사례" (Use Cases), and "정보공유" (Information Sharing). Below the navigation bar is a search bar with the placeholder text "데이터를 검색해보세요!" (Search for data) and a magnifying glass icon. Under the search bar, there is a section titled "최근 사회현안 및 이슈" (Recent Social Issues and Issues) with a button labeled "미세먼지" (Fine Dust) and a hand icon, followed by the text "데이터를 확인해 보세요!" (Check the data!). The main content area is divided into two sections: "국가중점데이터" (National Key Data) and "데이터" (Data). The "국가중점데이터" section features a grid of 12 icons representing various data categories: 교육 (Education), 국토관리 (Land Management), 공공행정 (Public Administration), 재정금융 (Finance), 산업고용 (Industry and Employment), 사회복지 (Social Welfare), 보건의료 (Health and Medical Care), 재난안전 (Disaster and Safety), 교통물류 (Transportation and Logistics), 환경기상 (Environment and Climate), 과학기술 (Science and Technology), and 농축수산 (Agriculture, Forestry, and Fisheries).

data.go.kr/main.do

DATA . GO . KR

마이페이지 로그아웃

데이터셋 제공신청 활용사례 정보공유

데이터를 검색해보세요!

최근 사회현안 및 이슈 미세먼지 데이터 확인해 보세요!

국가중점데이터 데이터

교육 국토관리 공공행정 재정금융 산업고용 사회복지

보건의료 재난안전 교통물류 환경기상 과학기술 농축수산

■ 공공데이터 (data.go.kr) 활용

● 활용 데이터 선정 후 신청

☰ > OPEN API > 개발계정 신청

개발계정 신청



> 기본정보

서비스명	국문 관광정보 서비스	서비스 유형	REST
심의여부	자동승인	신청유형	개발계정 활용신청
처리상태	신청	활용기간	승인일로부터 24개월 간 활용가능

> 시스템유형 선택 ?

* 시스템 유형

☒ 일반 ☐ 서버 구축

* 일반 : OpenAPI 서비스를 호출하여 응답받은 결과값을 서버에 저장하지 않고 사용할 경우 (서버 미 구축)

* 서버 구축 : OpenAPI 서비스를 호출하여 응답받은 결과값을 서버에 저장하거나 DB화 하여 사용할 경우

■ 공공데이터 (data.go.kr) 활용

● 활용 데이터 선정 후 신청

> 상세기능정보 필수 입력 정보입니다.

*자동승인 상세기능은 신청과 동시에 활용 가능합니다.

<input type="checkbox"/>	상세기능	설명	일일 트래픽
<input type="checkbox"/>	공통정보조회	관광정보에 매핑되는 서브이미지목록을 조회하는 기능	1000
<input type="checkbox"/>	이미지정보조회	추가 관광정보 상세내역을 조회한다. 파라미터형식에 따라 숙박정보가 포함될 수 있다	1000
<input type="checkbox"/>	반복정보조회	상세소개쉬는날, 개장기간 등 내역을 조회하는 기능	1000
<input type="checkbox"/>	소개정보조회	타입별공통정보기본정보,약도이미지,대표이미지,분류정보,지역정보,주소정보,좌표정보,개요정보,길안내정보,이미지정보,연계관광정보목록을 조회하는 기능	1000
<input type="checkbox"/>	위치기반 관광정보조회	위치기반 관광정보파라미터 타입에 따라서 제목순,수정일순,등록일순,인기순 정렬검색목록을 조회하는 기능	1000
<input type="checkbox"/>	지역기반 관광정보조회	지역기반 관광정보파라미터 타입에 따라서 제목순,수정일순,등록일순,인기순 정렬검색목록을 조회하는 기능	1000
<input type="checkbox"/>	숙박정보조회	숙박정보 굿스테이,베니키아,한옥 검색목록을 조회한다. 타입이 '숙박'일 경우에만 유효하다	1000
<input type="checkbox"/>	행사정보조회	행사정보목록을 조회한다. 콘텐츠 타입이 '행사'일 경우에만 유효하다	1000
<input type="checkbox"/>	키워드조회	키워드로 검색을하며 전체별 타입정보별 목록을 조회한다	1000
<input type="checkbox"/>	서비스분류코드조회	서비스분류코드목록을 대,중,소분류로 조회한다	1000
<input type="checkbox"/>	지역코드조회	지역코드목록을 지역,시군구,읍면동 코드목록을 조회하는 기능	1000

System Message

개발계정 신청이 완료되었습니다.
'마이페이지>OPEN API>개발계정'에서 신청
내역을 확인하실 수 있습니다.

확인

■ 공공데이터 (data.go.kr) 활용

● 서비스 키 발급

> 기본정보					
서비스명	국문 관광정보 서비스		상세설명		
서비스 유형	REST	일일트래픽	0	평균응답속도(초)	0
심의여부	자동승인	신청유형	개발계정 활용신청		처리상태
					승인
활용기간	2019-10-30 ~ 2021-10-30				
> 서비스정보					
일반 인증키 (UTF-8)	A2usf4txsthrBPXx7%2FckSjlSrlOOF3DRCXlpwxFKcl1KPyvOOWSD%2FUhcapvCkM51AdZOieoolvkMdz2XQCt33w%3D%3D 복사				
End Point	http://api.visitkorea.or.kr/openapi/service				
데이터포맷	XML				
참고문서	TourAPI_Guide_(Korean)v3.4_20181120.zip				

http://api.visitkorea.or.kr/openapi/service/rest/KorService/searchFestival?serviceKey=A2usf4txsthrBPXx7%2FckSjlSrlOOF3DRCXlpwxFKcl1KPyvOOWSD%2FUhcapvCkM51AdZOieoolvkMdz2XQCt33w%3D%3D&MobileOS=ETC&MobileApp=AppTest&eventStartDate=20191101&_type=json

■ 공공데이터 (data.go.kr) 활용

● 행사정보 조회 API – 요청 메시지 (request parameter)

14) [행사정보 조회] 오퍼레이션 명세

가) 오퍼레이션 내용

내용	행사/공연/축제 정보를 날짜로 조회하는 기능입니다. 콘텐츠 타입이 “행사/공연/축제” 인 경우만 유효합니다. 파라미터에 따라 제목순, 수정일순(최신순), 등록일순, 인기순 정렬검색을 제공합니다.
형식	http://api.visitkorea.or.kr/openapi/service/rest/KorService/searchFestival

나) 요청 메시지 (Request Parameter)

항목명(영문)	항목명(국문)	항목크기	항목구분	샘플데이터	항목 설명
numOfRows	한 페이지 결과 수	4	0	10	한 페이지 결과 수
pageNo	페이지 번호	4	0	1	현재 페이지 번호
MobileOS	OS 구분	30	1	ETC	IOS (아이폰), AND (안드로이드), WIN (윈도우폰), ETC
MobileApp	서비스명	60	1	AppTest	서비스명=어플명
ServiceKey	인증키 (서비스키)	100	1	인증키 (URL- Encode)	공공데이터포털에서 발급받은 인증키
listYN	목록 구분	1	0	Y	목록 구분 (Y=목록, N=개수)
arrange	정렬 구분	1	0	A	(A=제목순, B=조회순, C=수정일순, D=생성일순) 대표이미지가 반드시 있는 정렬 (O=제목순, P=조회순, Q= 수정일순, R=생성일순)
areaCode	지역코드	10	0		지역코드
sigunguCode	시군구코드	10	0		시군구코드(areaCode 필수)
eventStartDate	행사 시작일	8	0	20170901	행사 시작일 (형식: YYYYMMDD)
eventEndDate	행사 종료일	8	0	20170931	행사 종료일 (형식: YYYYMMDD)

■ 공공데이터 (data.go.kr) 활용

● 행사정보 조회 API – 응답 메시지 (response)

항목명	항목 설명	항목크기	항목구분	샘플데이터	항목 설명
<u>resultCode</u>	결과코드	4	1	0000	응답 결과코드
<u>resultMsg</u>	결과메시지	4	1	OK	응답 결과메시지
<u>numOfRows</u>	한 페이지 결과 수	4	1	10	한 페이지 결과 수
<u>pageNo</u>	페이지 번호	4	1	1	현재 페이지 번호
<u>totalCount</u>	전체 결과 수	5	1	483	전체 결과 수
<u>addr1</u>	주소	400	0	경기도 가평군 가평읍 달전리 1-1	주소(예, 서울 중구 다동)를 응답
<u>addr2</u>	상세주소	400	0	-	상세주소
<u>areacode</u>	지역코드	10	0	31	지역코드
<u>booktour</u>	교과서 속 여행지 여부	1	0	0	교과서 속 여행지 여부 (1=여행지, 0=해당없음)
<u>cat1</u>	대분류	12	0	A02	대분류 코드
<u>cat2</u>	중분류	12	0	A0207	중분류 코드
<u>cat3</u>	소분류	12	0	A02070200	소분류 코드
<u>contentid</u>	콘텐츠 ID	12	1	2039819	콘텐츠 ID
<u>contenttypeid</u>	콘텐츠타입 ID	2	1	15	관광타입(관광지, 숙박 등) ID
<u>createdtime</u>	등록일	14	1	20151109143241	콘텐츠 최초 등록일
<u>eventstartdate</u>	행사 시작일	8	1	20170901	행사 시작일 (형식: <u>YYYYMMDD</u>)
<u>eventenddate</u>	행사 종료일	8	1	20170903	행사 종료일 (형식: <u>YYYYMMDD</u>)
<u>firstimage</u>	대표이미지(원본)	100	0	<u>http://</u> <u>tong.visitkorea.or.kr/</u> <u>cms/resource/</u> <u>39/2503539_image2_</u> <u>1.png</u>	원본 대표이미지 (약 500*333 size) URL 응답

■ 공공데이터 (data.go.kr) 활용

● 전체 소스코드 (1 / 2)

```
package http;

import org.json.JSONArray;
import org.json.JSONObject;

public class Http6 {
    public static void main(String[] args) {
        NetworkUtil nu = new NetworkUtil();
        String url1 = "http://api.visitkorea.or.kr";
        String url2 = "/openapi/service/rest/KorService/searchFestival";
        String param = "?serviceKey=A2usf4txsthrBPXx7%2FckSjISr100F3DR";
        param += "CXlpwxFKcl1KPyv00WSD%2FUhcapvCkM51AdZ0ieoo";
        param += "IvkMdz2XQCt33w%3D%3D";
        param += "&MobileOS=ETC";
        param += "&MobileApp=AppTest";
        param += "&eventStartDate=20191101";
        param += "&_type=json";

        String result = nu.get(url1 + param);
    }
}
```

■ 공공데이터 (data.go.kr) 활용

● 전체 소스코드 (2 / 2)

```
JSONObject obj = new JSONObject(result);
JSONObject response = obj.getJSONObject("response");
JSONObject body = response.getJSONObject("body");
JSONObject items = body.getJSONObject("items");
JSONArray item = items.getJSONArray("item");
for(int i = 0; i < item.length(); i++) {
    JSONObject ele = item.getJSONObject(i);
    String address = ele.getString("addr1");
    String title = ele.getString("title");
    int startDate = ele.getInt("eventstartdate");
    int endDate = ele.getInt("eventenddate");
    System.out.printf("%s, %s, %d ~ %d\n",
        address, title, startDate, endDate);
}
```

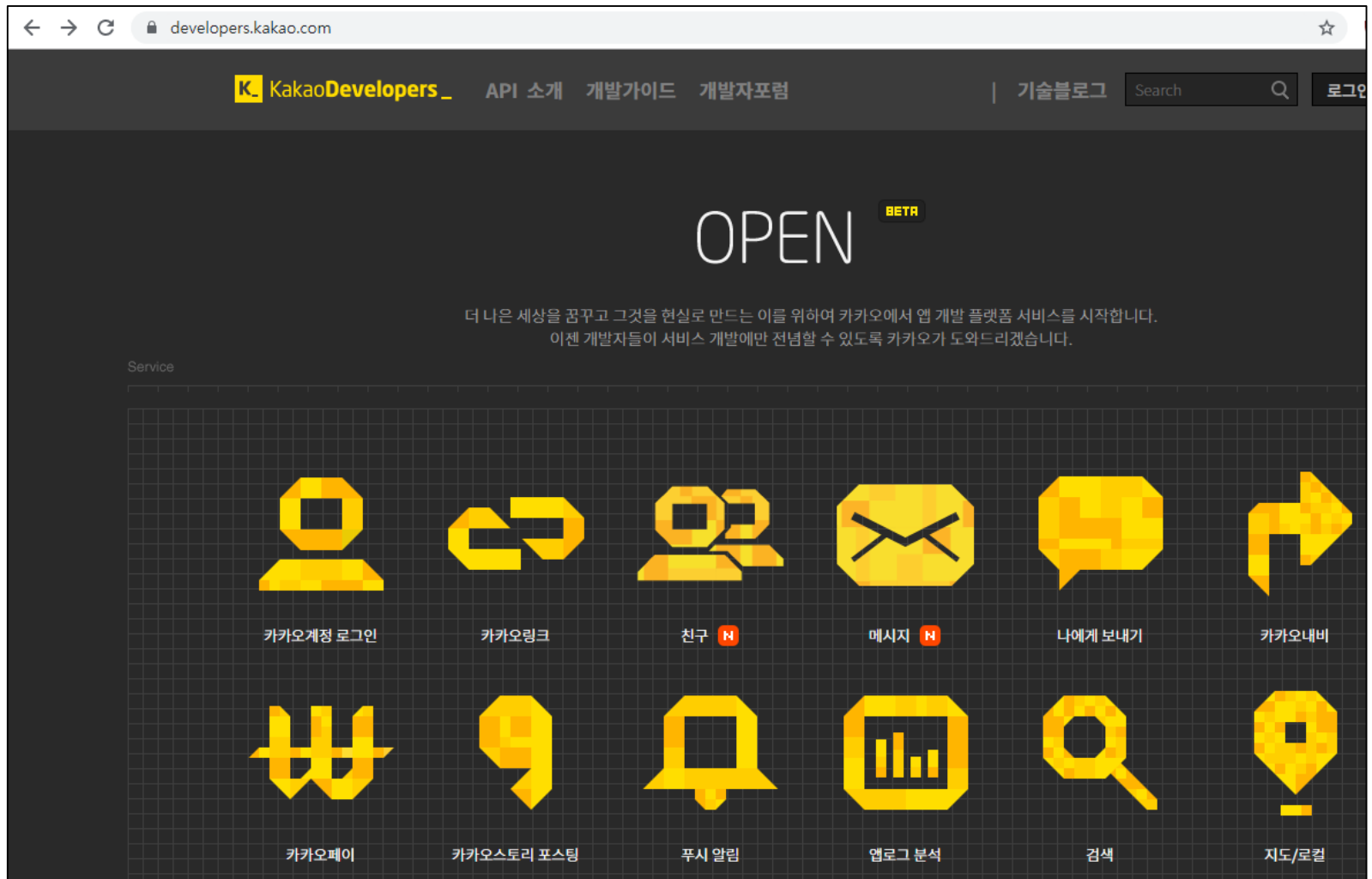
```
}
}
```

```
경기도 광주시 도척면 도척윗로 278, 가을 화담숲 단풍 축제 2019, 20191012 ~ 20191103
강원도 고성군 죽왕면 삼포리 1-1, 가족여행과 함께하는 고성 오션투유 삼포해수욕장 연날리기 축제 2019, 20190101 ~ 20191231
서울특별시 강동구 상암로 168, 강동특페스티벌 2019, 20191102 ~ 20191102
강원도 강릉시 울곡로3139번길 24, 강릉 관노가면극 2019, 20190413 ~ 20191103
강원도 강릉시 주문진읍 해안로 1730, 강릉 하트볼꽃크루즈 2019, 20190101 ~ 20191231
강원도 강릉시 종합운동장길 84, 강릉국제영화제(GIFF) 2019, 20191108 ~ 20191114
전라남도 강진군 강진읍 남당로 97-111, 강진만 춤추는 갈대축제 2019, 20191026 ~ 20191103
경상남도 거제시 거제면 거제남서로 3577, 거제섬꽃축제 2019, 20191026 ~ 20191103
경상남도 거창군 신원면 신차로 2924, 거창 국화전시 2019, 20191025 ~ 20191110
서울특별시 종로구 사직로 161, 경복궁 야간 특별관람 2019, 20190427 ~ 20191106
```

■ Kakao 주소 – 위도/경도 변환 API 활용

● Kakao Developers 관리 페이지 접속

- <https://developers.kakao.com/>



■ Kakao 주소 – 위도/경도 변환 API 활용

● 앱 만들기 (프로젝트 생성)

The screenshot displays the Kakao Developers portal interface. At the top, there's a navigation bar with the Kakao Developers logo, links for API 소개, 개발가이드, 개발자포럼, and 기술블로그, a search bar, and a user profile for 김남현. The main content area is titled '내애플리케이션' (My Applications) and shows a list of 8 applications under the heading '전체 애플리케이션 (8)'. The applications are arranged in a grid, each with an icon, a name, and a role (Owner or Editor). Below the list, there's a section for '사용량' (Usage) showing '총 API 요청' (Total API Requests) as 1 request and '총 쿼터 사용량' (Total Quota Usage) as 1 request, with a maximum quota of 3,000,000 requests. A progress bar visualizes the quota usage.

KakaoDevelopers API 소개 개발가이드 개발자포럼 | 기술블로그 Search 김남현

내애플리케이션

전체 애플리케이션 ▼

앱 만들기

전체 애플리케이션 (8)

APP	이름	역할
HTML	HTML	Owner
강동원	강동원	Owner
꼭지네민박	꼭지네민박	Owner
맛있는부산	맛있는부산	Editor
연습	연습	Owner
전자정부	전자정부	Owner
홍길동	홍길동	Owner

사용량

총 API 요청 ?
- 1건

총 쿼터 사용량 ?
- 1건

최대 3,000,000건

0 250k 500k 750k 1 000k 1 250k 1 500k 1 750k 2 000k 2 250k 2 500k 2 750k 3 0...

■ Kakao 주소 - 위도/경도 변환 API 활용

● REST API 키 확인

개요

카카오 로그인

[사용자 관리](#)

사용 불가 - 카카오 로그인을 사용하기 위해서는 사용자 관리를 설정해야 합니다.

앱 정보

[설정](#)

앱 키

?

[앱 키 표시](#)

네이티브 앱 키	?	d1f6ec7c7bd294f8ebbf38bc9ba4a68c
REST API 키	?	d4be7b479f4b4cbd99bd19ae87f88b4b
JavaScript 키	?	38d34c29c097c5f5c8c02416cb4aefff
Admin 키	?	d0b2b37135baa669a4b42e62361da3b1

■ Kakao 주소 – 위도/경도 변환 API 활용

● API 사용법 확인

주소검색

주소를 지도 위에 정확하게 표시하기 위해 해당 주소의 좌표 정보를 제공합니다. 주소 유형에 구분없이 변환할 수 있으며, 주소에 해당하는 지번 주소, 도로명 주소, 좌표, 우편번호, 빌딩명 등의 다양한 정보를 함께 제공하여 다양하게 활용할 수 있습니다.

[Request]

```
GET /v2/local/search/address.{format} HTTP/1.1
Host: dapi.kakao.com
Authorization: KakaoAK {app_key}
```

앱키를 헤더에 담아 GET으로 요청합니다. 원하는 검색어와 함께 결과 형식 파라미터의 값을 선택적으로 추가할 수 있습니다.

키	설명	필수	타입
query	검색을 원하는 질의어	O	String
page	결과 페이지 번호	X(기본 1)	Integer
size	한 페이지에 보여질 문서의 개수	X(기본 10)	1-30 사이 Integer

예를 들면,

```
curl -v -X GET "https://dapi.kakao.com/v2/local/search/address.json" \
--data-urlencode "query=전북 삼성동 100" \
-H "Authorization: KakaoAK kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk"
```

<https://dapi.kakao.com/v2/local/search/address.json?query=부산 연제구 연산동 1000>

■ Kakao 주소 – 위도/경도 변환 API 활용

● 응답 메시지

```
{
  "meta": {
    "total_count": 4,
    "pageable_count": 4,
    "is_end": true
  },
  "documents": [
    {
      "address_name": "전북 익산시 부송동 100",
      "y": "35.97664845766847",
      "x": "126.99597295767953",
      "address_type": "REGION_ADDR",
      "address": {
        "address_name": "전북 익산시 부송동 100",
        "region_1depth_name": "전북",
        "region_2depth_name": "익산시",
        "region_3depth_name": "부송동",
        "region_3depth_h_name": "삼성동",
        "h_code": "4514069000",
        "b_code": "4514013400",
        "mountain_yn": "N",
        "main_address_no": "100",
        "sub_address_no": "",
        "zip_code": "570972",
        "x": "126.99597295767953",
        "y": "35.97664845766847"
      },
      "road_address": {
        "address_name": "전북 익산시 망산길 11-17",
        "region_1depth_name": "전북",
        "region_2depth_name": "익산시",
        "region_3depth_name": "부송동",
        "road_name": "망산길",
        "underground_yn": "N",
        "main_building_no": "11",
        "sub_building_no": "17",
        "building_name": "",
        "zone_no": "54547",
        "v": "35.976749396987046"
      }
    }
  ]
}
```

■ Kakao 주소 – 위도/경도 변환 API 활용

● NetworkUtil.java 메소드 추가 (Request Header 정보 추가용)

```
public String getKakao(String address, String appKey) {  
    StringBuffer result = new StringBuffer();  
    try {  
        URL url = new URL(address);  
        URLConnection con = url.openConnection();  
        con.addRequestProperty("Authorization", "KakaoAK " + appKey);  
  
        InputStream is = con.getInputStream();  
        InputStreamReader isr = new InputStreamReader(is, "utf-8");  
        BufferedReader reader = new BufferedReader(isr);  
  
        String separator = "";
```

■ Kakao 주소 – 위도/경도 변환 API 활용

● NetworkUtil.java 메소드 추가 (Request Header 정보 추가용)

```
while (true) {  
    String data = reader.readLine();  
    if (data == null) {  
        break;  
    }  
    result.append(separator + data);  
    separator = "\n";  
}  
} catch (IOException e) {  
    e.printStackTrace();  
}  
return result.toString();  
}
```

■ Kakao 주소 – 위도/경도 변환 API 활용

● 전체 소스코드 (1 / 2)

```
package http;

import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;

import org.json.JSONArray;
import org.json.JSONObject;

public class Http7 {
    public static void main(String[] args)
        throws UnsupportedEncodingException {
        NetworkUtil nu = new NetworkUtil();
        String url = "https://dapi.kakao.com/v2/local/search/address.json";
        String param = "?query=" +
            URLEncoder.encode("부산 연제구 연산동 1000", "utf-8");
        String appKey = "d4be7b479f4b4cbd99bd19ae87f88b4b";
        String result = nu.getKakao(url + param, appKey);
    }
}
```

■ Kakao 주소 – 위도/경도 변환 API 활용

● 전체 소스코드 (2 / 2)

```
JSONObject obj = new JSONObject(result);
JSONArray documents = obj.getJSONArray("documents");
for(int i = 0; i < documents.length(); i++) {
    JSONObject document = documents.getJSONObject(i);
    String addressName = document.getString("address_name");
    String latitude = document.getString("y");
    String longitude = document.getString("x");
    System.out.printf("%s, %s / %s\n",
                      addressName, latitude, longitude);
}
}
```

부산 연제구 연산동 1000, 35.17991044339109 / 129.0746507518113

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● Naver Developers 관리 페이지 접속

- <https://developers.naver.com/>

The screenshot shows the Naver Developers homepage. At the top, there's a navigation bar with links to Products, Documents, Application, NAVER D2, Support, and Forum. A search bar is also present. Below the navigation bar, there are five main service tiles: Clova, 네이버 아이디로 로그인 (Naver ID Login), papago, 서비스 API (Service API), and 네이버 클라우드 플랫폼 (Naver Cloud Platform). Each tile has a logo and a brief description. Below these tiles, there are four announcement cards. The first card is for NAVER OpenAPI, mentioning Clova 음성 관련 API. The second card is for 네이버 아이디로 로그인, mentioning a new login page. The third card is for NAVER Developers, mentioning Clova 챗봇 빌더 서비스. The fourth card is for NAVER OpenAPI, mentioning Papago SMT 번역 서비스. At the bottom, there's a footer with links to various resources like API 이용 안내, Clova, 네이버 아이디로 로그인, 파파고, 서비스 API, API 공통 가이드, SDK & Tools, Clova, 네이버 아이디로 로그인, 파파고, 서비스 API, 내 애플리케이션, 애플리케이션 등록, Clova Platform Console β, API 제휴 신청, 계정 설정, NAVER D2 PROGRAM, D2.NAVER.COM, 네이버 오픈소스, Forum, 공지사항, API Status, and 라이선스 준수.

NAVER Developers

Products Documents Application NAVER D2 Support Forum

API 상태 Search Here

Clova 네이버 아이디로 로그인 papago 서비스 API 네이버 클라우드 플랫폼

NAVER OpenAPI

리 보기

NAVER Developers

NAVER OpenAPI

Clova 음성 관련 API는 이제 네이버 클라우드 플랫폼에서 만나 보세요!
2018.04.24 | 989

[네이버 아이디로 로그인] 사전 검수 신청 페이지 개선 및 가이드 오픈 공지
2019.09.24 | 2400016

Clova 챗봇 빌더 서비스 중지 및 NCP 챗봇 빌더 이용 안내
2019.09.18 | 505433

Papago SMT 번역 서비스 종료 안내
2019.08.28 | 350890

NAVER Developers

Products Documents Application NAVER D2 Support

API 이용 안내 Clova 네이버 아이디로 로그인 파파고 서비스 API

API 공통 가이드 SDK & Tools Clova 네이버 아이디로 로그인 파파고 서비스 API



내 애플리케이션 애플리케이션 등록 Clova Platform Console β API 제휴 신청 계정 설정

NAVER D2 PROGRAM D2.NAVER.COM 네이버 오픈소스 Forum

공지사항 API Status 라이선스 준수

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● Application 등록 (프로젝트 생성)

Application 목록		Application 등록
Client ID	Application 명	Action
QvG8m_WuBBMv3VZzHiwh	음성변환	
bxyQtmlZmYkfgpSChNBj	맛있는부산	
n76OOSmkSve7Q5K2VsRy	맛있는부산로컬	
ilMWEFLmpb62_00YWW5f	단축url	
OpcnSsAln37qlu6lyad6	검색테스트	
Whd1tzKCIU5K6E9dwXsq	네이버 검색	

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● API 설정

번역테스트					
개요	API 설정	멤버관리	로그인 통계	API 통계	Playground(Beta)
애플리케이션 이름 ↗	<div>번역테스트</div> <ul style="list-style-type: none">네이버 아이디로 로그인할 때 사용자에게 표시되는 이름이므로 가급적 10자 이내의 간결한 이름을 사용해주세요.40자 이내의 영문, 한글, 숫자, 공백문자, "-", "_" 만 입력 가능합니다.				
카테고리	<div>기타 ▼</div>				
사용 API ↗	<div>선택하세요. ▼</div> <div>Papago NMT 번역 ×</div>				

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● Client ID, Client Secret 확인

번역테스트

개요	API 설정	멤버관리	로그인 통계	API 통계	Playground(Beta)
----	--------	------	--------	--------	------------------

애플리케이션 정보

Client ID	<div>OpcnSsAln37qlu6lyad6</div>
Client Secret	<div>p7qtbsYx8N</div> <div>재발급</div>

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● API 사용법 확인

Documents

네이버 오픈 API를 이용해 창의적인 애플리케이션을 제작해 보세요.

파파고

파파고는 다국어 언어 처리에 대한 네이버의 기술과 경험을 제공하는 RESTful 형태의 API를 사용하면 서비스에 번역 기능을 추가할 수 있습니다.

파파고가 제공하는 API는 다음과 같습니다.

- **Papago NMT 번역**: 파파고의 인공 신경망 기반 기계 번역 서비스
- **Papago SMT 번역**: 파파고의 통계 기반 기계 번역 서비스
- **언어 감지**: 입력된 텍스트의 언어를 감지해 주는 RESTful API
- **한글 인명-로마자 변환**: 한글로 된 이름을 로마자 표기로 변환하는 서비스

하는 핵심 번역 기술을 강화하기 위해 품질 높은 학습 모델을 사용합니다.

손쉬운 사용

웹 기반의 콘솔을 통해 쉽고 편리한 사용이 가능합니다. RESTful 형태로 지원되는 API를 통해 고객의 서비스에 번역 기능을 간단하게 적용할 수 있습니다.

* 한국어(ko)-영어(en), 한국어(ko)-일본어(ja), 한국어(ko)-중국어 간체(zh-CN), 한국어(ko)-중국어 번체(zh-TW), 한국어(ko)-스페인어(es), 한국어(ko)-프랑스어(fr), 한국어(ko)-러시아어(ru), 한국어(ko)-베트남어(vi), 한국어(ko)-태국어(th), 한국어(ko)-인도네시아어(id), 한국어(ko)-독일어(de), 한국어(ko)-이탈리아어(it), 중국어 간체(zh-CN) - 중국어 번체(zh-TW), 중국어 간체(zh-CN) - 일본어(ja), 중국어 번체(zh-TW) - 일본어(ja), 영어(en)-일본어(ja), 영어(en)-중국어 간체(zh-CN), 영어(en)-중국어 번체(zh-TW), 영어(en)-프랑스어(fr)를 지원합니다.

* 처리한도 : 10,000글자/일

[오픈 API 이용 신청](#)[개발 가이드 보기](#)

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● API 사용법 확인

2. API 기본 정보 [↗](#)

API명	메서드	요청 URL	출력 포맷
N2MT	POST	https://openapi.naver.com/v1/papago/n2mt	JSON

요청예시 [↗](#)

```
POST /v1/papago/n2mt HTTP/1.1
HOST: openapi.naver.com
User-Agent: curl/7.49.1
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Naver-Client-Id: {애플리케이션 등록 시 발급받은 client id 값}
X-Naver-Client-Secret: {애플리케이션 등록 시 발급받은 client secret 값}
Content-Length: 51
```

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● 요청 메시지

3. 요청 변수

요청 변수명	타입	필수 여부	설명
source	string	Y	번역 대상 언어 코드. 1.ko : 한국어 2.en : 영어 3.zh-CN : 중국어 간체 4.zh-TW : 중국어 번체 5.es : 스페인어 6.fr : 프랑스어 7.vi : 베트남어 8.th : 태국어 9.id : 인도네시아어 ko(-)en, ko(-)zh-CN, ko(-)zh-TW, ko(-)es, ko(-)fr, ko(-)vi, ko(-)th, ko(-)id, en(-)ja, en(-)fr 조합만 가능
target	string	Y	번역 결과 언어 코드. 1.ko : 한국어 2.en : 영어 3.zh-CN : 중국어 간체 4.zh-TW : 중국어 번체 5.es : 스페인어 6.fr : 프랑스어 7.vi : 베트남어 8.th : 태국어 9.id : 인도네시아어 ko(-)en, ko(-)zh-CN, ko(-)zh-TW, ko(-)es, ko(-)fr, ko(-)vi, ko(-)th, ko(-)id, en(-)ja, en(-)fr 조합만 가능
text	string	Y	번역할 text (1회 호출시 최대 5,000자까지 가능)

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● 응답 메시지

응답예시 [↗](#)

```
{
  "message": {
    "@type": "response",
    "@service": "naverservice.labs.api",
    "@version": "1.0.0",
    "result": {
      "translatedText": "tea"
    }
  }
}
```

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● NetworkUtil.java 메소드 추가 (Request Header 정보 추가, POST 방식)

```
public String postNaver(String address, String param,
                        String clientId, String clientSecret) {

    StringBuffer result = new StringBuffer();

    try {
        URL url = new URL(address);
        URLConnection urlConnection = url.openConnection();
        HttpURLConnection con = (HttpURLConnection) urlConnection;

        con.setRequestProperty("X-Naver-Client-Id", clientId);
        con.setRequestProperty("X-Naver-Client-Secret", clientSecret);

        con.setRequestMethod("POST");
        con.setDoOutput(true);

        OutputStream out = con.getOutputStream();
        OutputStreamWriter osr = new OutputStreamWriter(out, "utf-8");
        BufferedWriter writer = new BufferedWriter(osr);
```

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● NetworkUtil.java 메소드 추가 (Request Header 정보 추가, POST 방식)

```
writer.write(param);
```

```
writer.close();
```

```
osr.close();
```

```
out.close();
```

```
InputStream is = con.getInputStream();
```

```
InputStreamReader isr = new InputStreamReader(is, "utf-8");
```

```
BufferedReader reader = new BufferedReader(isr);
```

```
String separator = "";
```

```
while (true) {
```

```
    String data = reader.readLine();
```

```
    if (data == null) {
```

```
        break;
```

```
    }
```


■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● NetworkUtil.java 메소드 추가 (Request Header 정보 추가, POST 방식)

```
        result.append(separator + data);
        separator = "\n";
    }
} catch (IOException e) {
    e.printStackTrace();
}

return result.toString();
}
```

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● 전체 소스코드 (1 / 2)

```
package http;

import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;

import org.json.JSONObject;

public class Http8 {
    public static void main(String[] args)
        throws UnsupportedEncodingException {
        NetworkUtil nu = new NetworkUtil();
        String url = "https://openapi.naver.com/v1/papago/n2mt";
        String param = "source=ko&target=fr&text=";
        param += URLEncoder.encode(
            "안녕하세요. 저는 자바 개발자입니다.", "utf-8");
        String clientId = "OpcnSsAIn37qIu6Iyad6";
        String clientSecret = "p7qtbsYx8N";
        String result = nu.postNaver(url, param, clientId, clientSecret);
    }
}
```

■ Naver Papago NMT (인공 신경망 기반 기계 번역) API 활용

● 전체 소스코드 (2 / 2)

```
JSONObject obj = new JSONObject(result);
JSONObject message = obj.getJSONObject("message");
JSONObject res = message.getJSONObject("result");

String translatedText = res.getString("translatedText");
System.out.printf("%s", translatedText);
}
}
```

```
Bonjour. Je suis le développeur de Java.
```