

# 검색API 활용 가이드





OPEN API 활용 가이드

## Contents



### 1. 서비스 소개

#### 1.1. API 서비스 소개

#### 1.2. 이용방법

### 2. 프로그램 구현하기

#### 2.1. 송수신 포맷

#### 2.2. 에러코드

#### 2.3. 일반적인 API이용방법

#### 2.4. 웹 페이지에서 직접 API이용방법

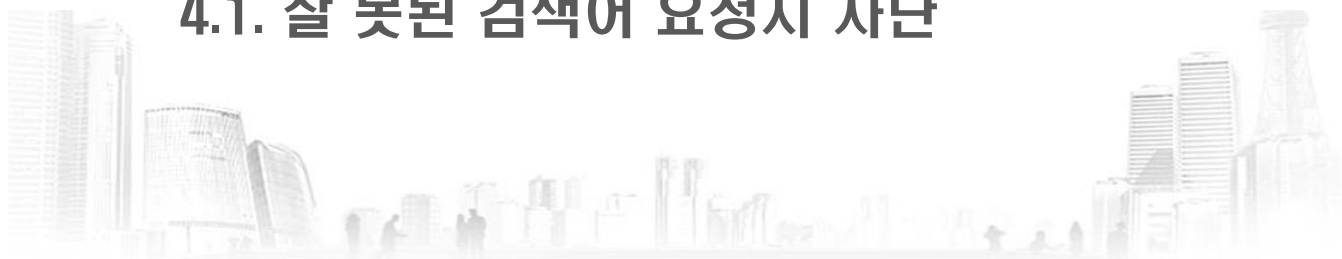
### 3. 이용 신청하기

#### 3.1. 신청하기

#### 3.2. 나의 신청내역 열람 및 변경

### 4. [참고] API 차단 사례

#### 4.1. 잘 못된 검색어 요청시 차단



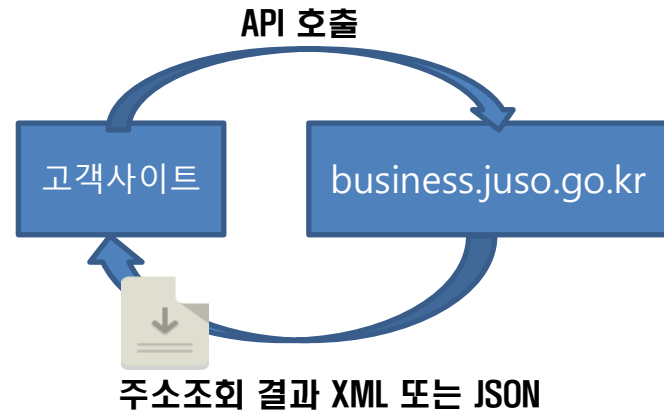


## 서비스 소개

도로명주소 DB 구축 없이 다양한 시스템 환경에서 실시간으로 주소 검색 기능을 구현할 수 있습니다.

### 1.1. API 서비스 개요

- ✓ 다양한 시스템 환경 적용 가능
- ✓ 개발언어, 플랫폼 제약없음
- ✓ DB 구축 및 현행화 부담ZERO
- ✓ 실시간 주소 검색 가능
- ✓ 신청 즉시 이용 가능
- ✓ 1 시스템 1 승인키 부여



### ◆ 검색기능 소개

<https://business.juso.go.kr/addrlink/addrLinkApi.do?currentPage=1&countPerPage=10&keyword=인천 남구 주안동 125&confmKey=승인키&hstryYn=Y>

검색결과	
22132	인천광역시 미추홀구 주안로 지하 91-1 (주안동)
22132	인천광역시 미추홀구 주안로 95-19 (주안동)

➤ '변동된 주소정보 포함' 옵션을 활용하여, **이전 주소정보**로 검색하여도 **현재 도로명주소**를 확인할 수 있습니다.

<https://business.juso.go.kr/addrlink/addrLinkApi.do?currentPage=1&countPerPage=10&keyword=인천 미추홀구 주안1동 125&confmKey=승인키>

검색결과	
22132	인천광역시 미추홀구 주안로 지하 91-1 (주안동)

➤ **행정동** 명칭으로 검색하여도 주소정보를 확인할 수 있습니다.

- ※ **법정동** : 법률로 지정된 행정구역 (주소, 공문서 등)
- ※ **행정동** : 행정운영 편의를 위하여 설정한 행정구역 (관할주민센터)



## 서비스 소개

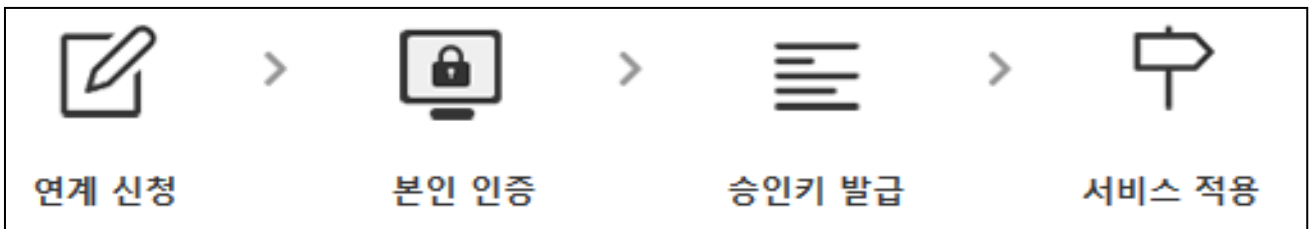
<https://business.juso.go.kr/addrlink/addrLinkApi.do?currentPage=1&countPerPage=10&keyword=세종로&confmKey=승인키&firstSort=road>

검색결과	
30125	세종특별자치시 세종로 905-22 (새롬동)
30125	세종특별자치시 세종로 911 (새롬동)
30125	세종특별자치시 세종로 933 (새롬동)
30125	세종특별자치시 세종로 941 (새롬동)
30064	세종특별자치시 세종로 1186 (종촌동)

- 다양한 정렬기능을 제공합니다.
  - **정확도순** : 검색한 키워드와 가장 일치하는 주소정보순으로 정렬
  - **도로명 포함** : 검색한 키워드와 일치하는 도로명 주소를 우선정렬
  - **지번 포함** : 검색한 키워드와 일치하는 지번 주소를 우선정렬

## 1.2. 이용방법

### ◆ 승인 키에 의한 주소 검색요청



### ◆ API 기본 정보

호출방식	요청 URL	출력결과
POST GET	<a href="https://business.juso.go.kr/addrlink/addrLinkApiJsonp.do">https://business.juso.go.kr/addrlink/addrLinkApiJsonp.do</a>	JSONP (XML또는JSON)
POST GET	<a href="https://business.juso.go.kr/addrlink/addrLinkApi.do">https://business.juso.go.kr/addrlink/addrLinkApi.do</a>	XML또는JSON

- 행정망 내 이용을 위한 URL은 도로명주소 도움센터(1588-0061)로 별도 문의 바랍니다.



## 프로그램 구현하기

JSP와 Ajax, JAVA와 Spring 프레임워크를 이용하여 일반적인 API호출방식과 웹 페이지에서 직접 API를 호출하는 방식을 안내합니다.

### 2.1. 송수신 포맷

#### ◆ 요청 변수

요청변수명	타입	필수여부	기본값	설명
confmKey	String	Y	-	신청 시 발급받은 승인키
currentPage	Number	Y	1	현재 페이지 번호 ( $n > 0$ )
countPerPage	Number	Y	10	페이지당 출력 할 결과 Row 수 ( $0 < n \leq 100$ )
keyword	String	Y	-	주소 검색어
resultType	String	N	-	검색결과형식 설정(기본 XML형식) json 입력 시 JSON형식의 결과제공
hstryYn	String	N	N	* 2020년12월8일 추가된 항목 변동된 주소정보 포함 여부
firstSort	String	N	none	* 2020년12월8일 추가된 항목 정확도 정렬 (none), 우선정렬 (road: 도로명 포함, location: 지번 포함) ※ keyword(검색어)가 우선정렬 항목에 포함된 결과 우선 표출
addInfoYn	String	N	N	* 2020년12월8일 추가된 항목 출력결과에 추가된 항목(hstryYn, relJibun, hemdNm) 제공여부 ※ 해당 옵션으로 추가제공되는 항목의 경우, 추후 특정항목이 제거되거나 추가될 수 있으니 적용 시 고려해주시기 바랍니다.



## 프로그램 구현하기

### ◆ 출력 결과(XML, JSON)

※ 2020년 12월 제공항목 추가

	출력 변수명	타입	필수여부	설명
common	totalCount	String	Y	총 검색 데이터수
	currentPage	Integer	Y	페이지 번호
	countPerPage	Integer	Y	페이지당 출력할 결과 Row수
	errorCode	String	Y	에러 코드
	errorMessage	String	Y	에러 메시지
juso	roadAddr	String	Y	전체 도로명주소
	roadAddrPart1	String	Y	도로명주소(참고항목 제외)
	roadAddrPart2	String	N	도로명주소 참고항목
	jibunAddr	String	Y	지번 정보
	engAddr	String	Y	도로명주소(영문)
	zipNo	String	Y	우편번호
	admCd	String	Y	행정구역코드
	rnMgtSn	String	Y	도로명코드
	bdMgtSn	String	Y	건물관리번호
	detBdNmList	String	N	상세건물명
	bdNm	String	N	건물명
	bdKdcdd	String	Y	공동주택여부 (1:공동주택, 0: 비공동주택)
	siNm	String	Y	시도명
	sggNm	String	N	시군구명
	emdNm	String	Y	읍면동명
	liNm	String	N	법정리명
	Rn	String	Y	도로명
	udrtYn	String	Y	지하여부 (0:지상, 1:지하)
	buldMnnm	Number	Y	건물본번
	buldSln	Number	Y	건물부번 (부번이 없는 경우 0)



## 프로그램 구현하기

### ◆ 출력 결과(XML, JSON)

※ 2020년 12월 제공항목 추가

juso	mtYn	String	Y	산여부 (0:대지, 1:산)
	InbrMnnm	Number	Y	지번본번(번지)
	InbrSln0	Number	Y	지번부번(호) (부번이 없는 경우 0)
	emdNo	String	Y	읍면동일련번호
	hstryYn	String	Y	* 2020년12월8일 추가된 항목 변동이력여부(0: 현행 주소정보, 1: 요청변수의 keyword(검색어)가 변동된 주소정보에서 검색된 정보)
	relJibun	String	Y	* 2020년12월8일 추가된 항목 관련지번
	hemdNm	String	Y	* 2020년12월8일 추가된 항목 관할주민센터 ※ 참고정보이며, 실제와 다를 수 있습니다.



## 프로그램 구현하기

### 2.2. 에러코드

#### ◆ 에러코드 정의

에러코드	에러메세지
0	정상
-999	시스템에러
E0001	승인되지 않은 KEY 입니다.
E0002	승인되지 않은 사이트 입니다.
E0003	정상적인 경로로 접속하시기 바랍니다.
E0005	검색어가 입력되지 않았습니다.
E0006	주소를 상세히 입력해 주시기 바랍니다.
E0008	검색어는 한글자 이상 입력되어야 합니다.
E0009	검색어는 문자와 숫자 같이 입력되어야 합니다. (숫자만 검색 불가)
E0010	검색어가 너무 깁니다. (한글40자, 영문,숫자 80자 이하)
E0011	검색어에 너무 긴 숫자가 포함되어 있습니다. (숫자10자 이하)
E0012	특수문자+숫자만으로는 검색이 불가능 합니다.
E0013	SQL 예약어 또는 특수문자( %, =, >, <, [ ] )는 검색이 불가능 합니다.
E0014	개발승인키 기간이 만료되어 서비스를 이용하실 수 없습니다.
E0015	검색 범위를 초과하였습니다.





## 프로그램 구현하기

### 자주 범하는 오류 사례

- ❖ E0001 에러코드가 회신된 경우에는 신청한 연계 방식을 확인하세요.
- ❖ 팝업API로 이용 신청하시고 검색API를 이용하시는 경우이거나, 검색API를 신청하시고 팝업API를 이용하시는 경우입니다.
- ❖ 팝업API와 검색API는 승인 키를 같이 사용할 수 없습니다.
- ❖ 신청하신 연계방식에 따라 구현하시거나, 구현한 방식에 따라 팝업API 또는 검색API 서비스를 신청하셔서, 승인 키를 발급받으시기 바랍니다.
- ❖ 검색어에 시도, 시군구만을 검색 할 수 없습니다.
- ❖ 과도한 페이징 요청으로 검색범위를 초과하는 검색은 불가능합니다.  
(리스트 10개 기준 900페이지)



### 이용 주의사항

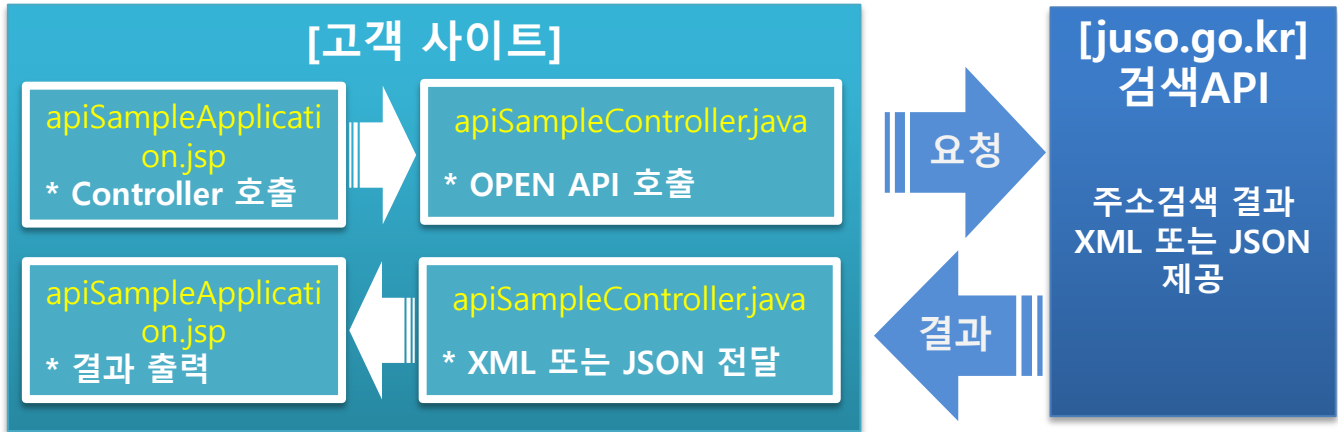
주소 검색어(keyword)에 **SQL 예약어**(select, or, union 등) 및 **특수문자**( <, >, %, = 등)이 포함된 경우에는 서비스 이용이 중단될 수 있습니다.



## 도로명 2

## 프로그램 구현하기

### ◆ 도로명주소 검색API 호출



### ◆ 샘플 데이터 보기 (xml)

<https://business.juso.go.kr/addrlink/addrLinkApi.do?currentPage=1&countPerPage=10&keyword=강남대로12길&confmKey=TESTJUSOGOKR>

### ◆ 샘플 데이터 보기 (json)

<https://business.juso.go.kr/addrlink/addrLinkApi.do?currentPage=1&countPerPage=10&keyword=강남대로12길&confmKey=TESTJUSOGOKR&resultType=json>

- 브라우저에 위 URL을 입력하시면 XML Data, JSON Data 샘플을 보실 수 있습니다.  
위 URL로 확인이 안될시 인코딩이 맞지 않아 검색결과를 받아오지 못하는 경우이므로,  
IE외 다른 브라우저를 이용하여 확인 하실 수 있습니다.
- 연계가 정상적으로 되나 검색건수가 0인 경우에는 한글 인코딩 문제 입니다.  
keyword를 `URLEncoder.encode(keyword,"UTF-8")` 해주시길 바랍니다.



## 프로그램 구현하기

### ◆ AJAX와 JAVA Controller를 이용한 XML 결과 적용하기 (1/2)

#### 샘플 파일 : apiSampleApplicationXML.jsp

```

<script language="javascript">
function getAddrLoc(){
    // 적용예 (api 호출 전에 검색어 체크)
    if (!checkSearchedWord(document.form.keyword))
    {
        return ;
    }

    $.ajax({
        url :"/sample/getAddrApi.do"
        ,type:"post"
        ,data:$("#form").serialize()
        ,dataType:"xml"
        ,success:function(xmlStr){
            $("#list").html("");
            var errCode =
            $(xmlStr).find("errorCode").text();
            var errDesc =
            $(xmlStr).find("errorMessage").text();
            if(errCode != "0"){
                alert(errCode+"="+errDesc);
            }else{
                if(xmlStr != null){
                    makeList(xmlStr);
                }
            }
        }
        ,error: function(xhr,status, error){
            alert("에러발생");
        }
    });
}

//특수문자, 특정문자열(sql예약어의 앞뒤공백포함) 제거
function checkSearchedWord(obj){
    if(obj.value.length >0){
        //특수문자 제거
        var expText = /[%=><]/ ;
        if(expText.test(obj.value) == true){
            alert("특수문자를 입력 할수 없습니다.") ;
            obj.value = obj.value.split(expText).join("");
            return false;
        }

        //특정문자열(sql예약어의 앞뒤공백포함) 제거
        var sqlArray = new Array(
            //sql 예약어
            "OR", "SELECT", "INSERT", "DELETE", "UPDATE"
            ,"CREATE", "DROP", "EXEC", "UNION"
            ,"FETCH", "DECLARE", "TRUNCATE" );

        var regex;
        for(var i=0; i<sqlArray.length; i++){
            regex = new RegExp( sqlArray[i] ,"gi" ) ;

            if (regex.test(obj.value) ) {
                alert("₩" + sqlArray[i] + "₩와(과) 같은 특
                정문자로 검색할 수 없습니다.");
                obj.value =obj.value.replace(regex, "");
                return false;
            }
        }
        return true ;
    }
}
</script>

```



## 프로그램 구현하기

### ◆ AJAX와 JAVA Controller를 이용한 XML 결과 적용하기 (2/2)

#### 샘플 파일 : apiSampleXMLController.java

```
@RequestMapping(value="/sample/getAddrApi.do")
public void getAddrApi(HttpServletRequest req,
                        HttpServletResponse response){

    // 요청변수 설정
    String currentPage = req.getParameter("currentPage");
    String countPerPage = req.getParameter("countPerPage");
    String confmKey = req.getParameter("confmKey");
    String keyword = req.getParameter("keyword");

    // API 호출 URL 정보 설정
    String apiUrl =
        "https://business.juso.go.kr/addrlink/addrLinkApi.do?
        currentPage="+currentPage+
        "&countPerPage="+countPerPage+
        "&keyword="+URLLEncoder.encode(keyword, "UTF-8")+
        "&confmKey="+confmKey;

    URL url = new URL(apiUrl);
    BufferedReader br = new BufferedReader(
        new InputStreamReader(
            url.openStream(), "UTF-8"));
    StringBuffer sb = new StringBuffer();
    String tempStr = null;
    while(true){
        tempStr = br.readLine();
        if(tempStr == null) break;
        sb.append(tempStr); // 응답결과 XML 저장
    }
    br.close();
    response.setCharacterEncoding("UTF-8");
    response.setContentType("text/xml");
    response.getWriter().write(sb.toString()); // 응답결과 반환
}
```

※ 모의 해킹 테스트 시 검색API를 호출하시면 IP가 차단 될 수 있습니다.  
주소검색API를 제외하시고 테스트 하시기 바랍니다.



## 프로그램 구현하기

### ◆ 검색결과 (XML)

```

<?xml version="1.0" encoding="UTF-8"?>
<results>
<common>
  <totalCount>1</totalCount>
  <currentPage>1</currentPage>
  <countPerPage>10</countPerPage>
  <errorCode>0</errorCode>
  <errorMessage>정상</errorMessage>
</common>
<juso>
  <roadAddr>서울특별시 마포구 성암로 301 (상암동)</roadAddr>
  <roadAddrPart1>서울특별시 마포구 성암로 301</roadAddrPart1>
  <roadAddrPart2>(상암동)</roadAddrPart2>
  <jibunAddr>서울특별시 마포구 상암동 1595 한국지역정보개발원(KLID Tower)
    </jibunAddr>
  <engAddr>301 Seongam-ro, Mapo-gu, Seoul</engAddr>
  <zipNo>03923</zipNo>
  <admCd>1144012700</admCd>
  <rnMgtSn>14403113012</rnMgtSn>
  <bdMgtSn>1144012700115950000000001</bdMgtSn>
  <detBdNmList></detBdNmList>
  <bdNm>한국지역정보개발원(KLID Tower)</bdNm>
  <bdKdcd>0</bdKdcd>
  <siNm>서울특별시</siNm>
  <sggNm>마포구</sggNm>
  <emdNm>상암동</emdNm>
  <liNm></liNm>
  <rn>성암로</rn>
  <udrtYn>0</udrtYn>
  <buldMnnm>301</buldMnnm>
  <buldSlno>0</buldSlno>
  <mtYn>0</mtYn>
  <lnbrMnnm>1595</lnbrMnnm>
  <lnbrSlno>0</lnbrSlno>
  <emdNo>03</emdNo>
</juso>
</results>

```

### ◆ AJAX와 JAVA Controller를 이용한 JSON 결과 적용하기 (1/2)

#### 샘플 파일 : apiSampleApplicationJSON.jsp

```

<script language="javascript">
function getAddrLoc(){
    // 적용예 (api 호출 전에 검색어 체크)
    if (!checkSearchedWord(document.form.keyword))
    {
        return ;
    }

    $.ajax({
        url :"/sample/getAddrApi.do"
        ,type:"post"
        ,data:$("#form").serialize()
        ,dataType: "json"
        ,success:function(jsonStr){
            $("#list").html("");
            var errCode =
            jsonStr.results.common.errorCode;
            var errDesc =
            jsonStr.results.common.errorMessage;
            if(errCode!= " 0 " ){
                alert (errCode+"="+errDesc);
            }else{
                if(xmlStr!= null){
                    makeListJson(jsonStr); //
                    JSON데이터 HTML형태로 변환
                }
            }
            },error: function(xhr,status, error){
                alert("에러발생");
            }
        });
}

//특수문자, 특정문자열(sql예약어의 앞뒤공백포함)
//제거
function checkSearchedWord(obj){
    if(obj.value.length >0){
        //특수문자 제거
        var expText = /[%=><]/;
        if(expText.test(obj.value) == true){
            alert("특수문자를 입력 할수 없습니다.");
            obj.value = obj.value.split(expText).join("");
            return false;
        }

        //특정문자열(sql예약어의 앞뒤공백포함) 제거
        var sqlArray = new Array(
            //sql 예약어
            "OR", "SELECT", "INSERT", "DELETE", "UPDATE"
            ,"CREATE", "DROP", "EXEC", "UNION"
            ,"FETCH", "DECLARE", "TRUNCATE" );

        var regex;
        for(var i=0; i<sqlArray.length; i++){
            regex = new RegExp( sqlArray[i] ,"gi" );

            if (regex.test(obj.value) ) {
                alert("W" + sqlArray[i] + "W와(과) 같은 특
                정문자로 검색할 수 없습니다.");
                obj.value =obj.value.replace(regex, "");
                return false;
            }
        }
        return true ;
    }
}
</script>

```

```

<form name="form" id="form" method="post">
    <!-- 요청 변수 설정 (검색결과형식 설정, json) -->
    <input type="hidden" name="resultType" value="json"/>
</form>

```

※ resultType을 json으로 넘겨주셔야 jsonData를 받으실 수 있습니다.



## 프로그램 구현하기

### ◆ AJAX와 JAVA Controller를 이용한 JSON 결과 적용하기 (2/2)

#### 샘플 파일 : apiSampleJSONController.java

```
@RequestMapping(value="/sample/getAddrApi.do")
public void getAddrApi(HttpServletRequest req,
                        HttpServletResponse response){

    // 요청변수 설정
    String currentPage = req.getParameter("currentPage");
    String countPerPage = req.getParameter("countPerPage");
    String resultType = req.getParameter("resultType");
    String confmKey = req.getParameter("confmKey");
    String keyword = req.getParameter("keyword");

    // API 호출 URL 정보 설정
    String apiUrl =
        "https://business.juso.go.kr/addrlink/addrLinkApi.do?
        currentPage="+currentPage+
        "&countPerPage="+countPerPage+
        "&keyword="+URLLEncoder.encode(keyword, "UTF-8")+
        "&confmKey="+confmKey+"&resultType="+resultType;

    URL url = new URL(apiUrl);
    BufferedReader br = new BufferedReader(
        new InputStreamReader(
            url.openStream(), "UTF-8"));
    StringBuffer sb = new StringBuffer();
    String tempStr = null;
    while(true){
        tempStr = br.readLine();
        if(tempStr == null) break;
        sb.append(tempStr); // 응답결과 JSON 저장
    }
    br.close();
    response.setCharacterEncoding("UTF-8");
    response.setContentType("text/xml");
    response.getWriter().write(sb.toString()); // 응답결과 반환
}
```

※ 모의 해킹 테스트 시 검색API를 호출하시면 IP가 차단 될 수 있습니다.  
주소검색API를 제외하시고 테스트 하시기 바랍니다.



## 프로그램 구현하기

### ◆ 검색 결과 (JSON)

```

{
  "results":
  { "common":
    {
      "totalCount": "1"
      , "currentPage": "1"
      , "countPerPage": "10"
      , "errorCode": "0"
      , "errorMessage": "정상"
    }
    , "juso":
    [
      {
        "roadAddr": "서울특별시 마포구 성암로 301 (상암동)"
        , "roadAddrPart1": "서울특별시 마포구 성암로 301"
        , "roadAddrPart2": "(상암동)"
        , "jibunAddr": "서울특별시 마포구 상암동 1595 한국지역정보개발원(KLID Tower)"
        , "engAddr": "301 Seongam-ro, Mapo-gu, Seoul"
        , "zipNo": "03923"
        , "admCd": "1144012700"
        , "rnMgtSn": "114403113012"
        , "bdMgtSn": "11440127001159500000000001"
        , "detBdNmList": ""
        , "bdNm": "한국지역정보개발원(KLID Tower)"
        , "bdKdcd": "0"
        , "siNm": "서울특별시"
        , "sggNm": "마포구"
        , "emdNm": "상암동"
        , "liNm": ""
        , "rn": "성암로"
        , "udrtYn": "0"
        , "buldMnnm": "301"
        , "buldSlno": "0"
        , "mtYn": "0"
        , "lnbrMnnm": "1595"
        , "lnbrSlno": "0"
        , "emdNo": "03"
      }
    ]
  }
}

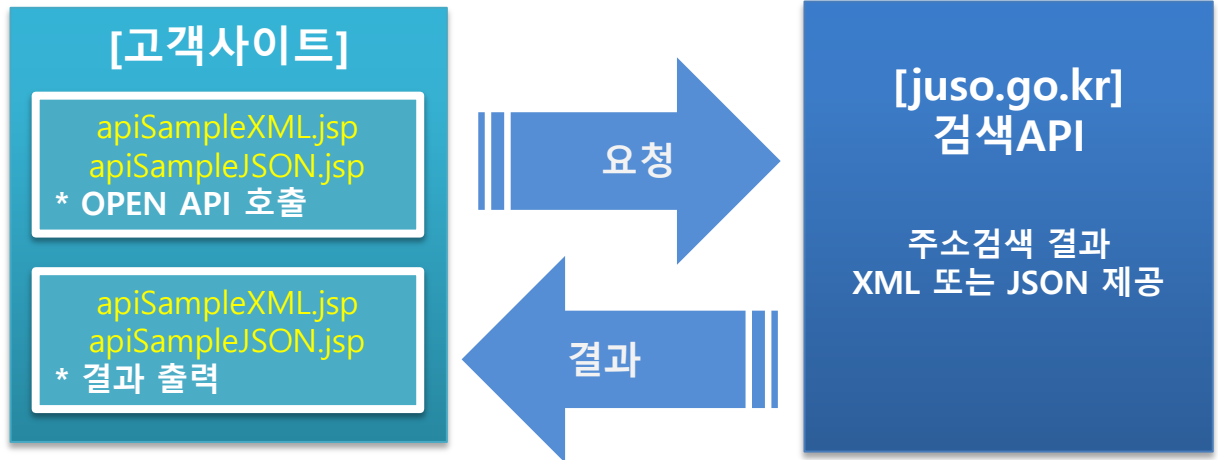
```



## 프로그램 구현하기

### 2.4. 웹 페이지에서 직접 API 이용방법

#### ◆ 도로명주소 검색API 호출



#### ◆ 도로명주소검색 API

- 웹에서 직접 주소검색 서버를 호출하여 손쉽게 주소검색기능을 구현하 실 수 있습니다.
- XML, JSON 데이터를 제공합니다.
- 크로스브라우징을 피하기 위해 JSONP 방식을 사용합니다.



## 프로그램 구현하기

### ◆ AJAX를 이용한 JSONP(XML)결과 적용하기

#### 샘플 파일 : apiSampleXML.jsp

```

<script language="javascript">
function getAddrLoc(){
    // 적용예 (api 호출 전에 검색어 체크)
    if (!checkSearchedWord(document.form.keyword))
    { return ; }

    $.ajax({
        url :
        "https://business.juso.go.kr/addrlink/addrLinkApiJsonp.do"
        ,type:"post"
        ,data:$("#form").serialize()
        ,dataType: "jsonp"
        ,crossDomain:true
        ,success:function(xmlStr){
            if (navigator.appName.indexOf ("Microsoft")
                > -1 ){
                //IE 환경에서 JSONP의 returnXml 결과데이터처리
                var xmlData =
                    newActiveXObject ("Microsoft.XMLDOM");
                xmlData.loadXML(xmlStr.returnXml)
            }else{
                //IE 이외 환경에서 처리
                var xmlData= xmlStr.returnXml;
            }
            $("#list").html("");
            var errCode=
                $(xmlData).find("errorCode").text();
            var errDesc=
                $(xmlData).find("errorMessage").text();
            if(errCode != "0"){
                alert(errCode+"="+errDesc);
            }else{
                if(xmlStr!= null){
                    makeList(xmlData);
                }
            }
            },error: function(xhr,status, error){
                alert("에러발생");
            }
        });
    }
}

```

```

//특수문자, 특정문자열(sql예약어의 앞뒤공백포함) 제거
function checkSearchedWord(obj){
    if(obj.value.length >0){
        //특수문자 제거
        var expText = /[%=><]/;
        if(expText.test(obj.value) == true){
            alert("특수문자를 입력 할수 없습니다.");
            obj.value = obj.value.split(expText).join("");
            return false;
        }

        //특정문자열(sql예약어의 앞뒤공백포함) 제거
        var sqlArray = new Array(
            //sql 예약어
            "OR", "SELECT", "INSERT", "DELETE", "UPDATE"
            ,"CREATE", "DROP", "EXEC", "UNION"
            ,"FETCH", "DECLARE", "TRUNCATE" );

        var regex;
        for(var i=0; i<sqlArray.length; i++){
            regex = new RegExp( sqlArray[i] ,"gi" );

            if (regex.test(obj.value) ) {
                alert("W" + sqlArray[i] + "W"와(과) 같은 특정문자로 검색할 수 없습니다.");
                obj.value =obj.value.replace(regex, "");
                return false;
            }
        }
        return true ;
    }
}
</script>

```

❖ 모의 해킹 테스트 시 검색API를 호출하면 IP가 차단 될 수 있습니다.  
주소검색API를 제외하시고 테스트 하시기 바랍니다.



## 프로그램 구현하기

### ◆ 검색결과 (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
<common>
  <totalCount>1</totalCount>
  <currentPage>1</currentPage>
  <countPerPage>10</countPerPage>
  <errorCode>0</errorCode>
  <errorMessage>정상</errorMessage>
</common>
<juso>
  <roadAddr>서울특별시 마포구 성암로 301 (상암동)</roadAddr>
  <roadAddrPart1>서울특별시 마포구 성암로 301</roadAddrPart1>
  <roadAddrPart2>(상암동)</roadAddrPart2>
  <jibunAddr>서울특별시 마포구 상암동 1595 한국지역정보개발원(KLID Tower)
    </jibunAddr>
  <engAddr>301 Seongam-ro, Mapo-gu, Seoul</engAddr>
  <zipNo>03923</zipNo>
  <admCd>1144012700</admCd>
  <rnMgtSn>14403113012</rnMgtSn>
  <bdMgtSn>1144012700115950000000001</bdMgtSn>
  <detBdNmList></ detBdNmList>
  <bdNm>한국지역정보개발원(KLID Tower)</bdNm>
  <bdKdcd>0</bdKdcd>
  <siNm>서울특별시</siNm>
  <sggNm>마포구</sggNm>
  <emdNm>상암동</emdNm>
  <liNm></liNm>
  <rn>성암로</rn>
  <udrtYn>0</udrtYn>
  <buldMnnm>301</buldMnnm>
  <buldSlno>0</buldSlno>
  <mtYn>0</mtYn>
  <lnbrMnnm>1595</lnbrMnnm>
  <lnbrSlno>0</lnbrSlno>
  <emdNo>03</emdNo>
</juso>
</results>
```



## 프로그램 구현하기

### ◆ AJAX를 이용한 JSONP(JSON)결과 적용하기

#### 샘플 파일 : apiSampleJSON.jsp

```

<script language="javascript">
function getAddrLoc(){
    // 적용예 (api 호출 전에 검색어 체크)
    if (!checkSearchedWord(document.form.keyword))
    { return ; }

    $.ajax({
        url :
        "https://business.juso.go.kr/addrlink/addrLinkApiJsonp.do"
        ,type:"post"
        ,data:$("#form").serialize()
        ,dataType: "jsonp"
        ,crossDomain:true
        ,success:function(jsonStr){
            $("#list").html("");
            var errCode=
                jsonStr.results.common.errorCode;
            var errDesc=
                jsonStr.results.common.errorMessage;
            if(errCode != "0"){
                alert(errCode+"="+errDesc);
            }else{
                if(jsonStr!= null){
                    makeList(jsonStr);
                }
            }
        },error: function(xhr,status, error){
            alert("에러발생");
        }
    });
}

<form name="form" id="form" method="post">
    <!-- 요청 변수 설정 (검색결과형식 설정, json) -->
    <input type="hidden" name="resultType" value="json"/>
</form>

//특수문자, 특정문자열(sql예약어의 앞뒤공백포함) 제거
function checkSearchedWord(obj){
    if(obj.value.length > 0){
        //특수문자 제거
        var expText = /[%=><]/;
        if(expText.test(obj.value) == true){
            alert("특수문자를 입력 할수 없습니다.");
            obj.value = obj.value.split(expText).join("");
            return false;
        }

        //특정문자열(sql예약어의 앞뒤공백포함) 제거
        var sqlArray = new Array(
            //sql 예약어
            "OR", "SELECT", "INSERT", "DELETE", "UPDATE"
            ,"CREATE", "DROP", "EXEC", "UNION"
            ,"FETCH", "DECLARE", "TRUNCATE" );

        var regex;
        for(var i=0; i<sqlArray.length; i++){
            regex = new RegExp( sqlArray[i], "gi" );

            if (regex.test(obj.value) ) {
                alert("W" + sqlArray[i] + "W"와(과) 같은 특정문자로 검색할 수 없습니다.");
                obj.value = obj.value.replace(regex, "");
                return false;
            }
        }
        return true ;
    }
}
</script>

```

※ resultType을 json으로 넘겨주셔야 jsonData를 받으실 수 있습니다.

❖ 모의 해킹 테스트 시 검색API를 호출하시면 IP가 차단 될 수 있습니다.  
주소검색API를 제외하시고 테스트 하시기 바랍니다.



## 프로그램 구현하기

### ◆ 검색 결과 (JSON)

```

{
  "results":
  {
    "common":
    {
      "totalCount": "1"
      , "currentPage": "1"
      , "countPerPage": "10"
      , "errorCode": "0"
      , "errorMessage": "정상"
    }
    , "juso":
    [
      {
        "roadAddr": "서울특별시 마포구 성암로 301 (상암동)"
        , "roadAddrPart1": "서울특별시 마포구 성암로 301"
        , "roadAddrPart2": "(상암동)"
        , "jibunAddr": "서울특별시 마포구 상암동 1595 한국지역정보개발원(KLID Tower)"
        , "engAddr": "301 Seongam-ro, Mapo-gu, Seoul"
        , "zipNo": "03923"
        , "admCd": "1144012700"
        , "rnMgtSn": "114403113012"
        , "bdMgtSn": "11440127001159500000000001"
        , "detBdNmList": ""
        , "bdNm": "한국지역정보개발원(KLID Tower)"
        , "bdKdcd": "0"
        , "siNm": "서울특별시"
        , "sggNm": "마포구"
        , "emdNm": "상암동"
        , "liNm": ""
        , "rn": "성암로"
        , "udrtYn": "0"
        , "buldMnnm": "301"
        , "buldSlno": "0"
        , "mtYn": "0"
        , "lnbrMnnm": "1595"
        , "lnbrSlno": "0"
        , "emdNo": "03"
      }
    ]
  }
}

```

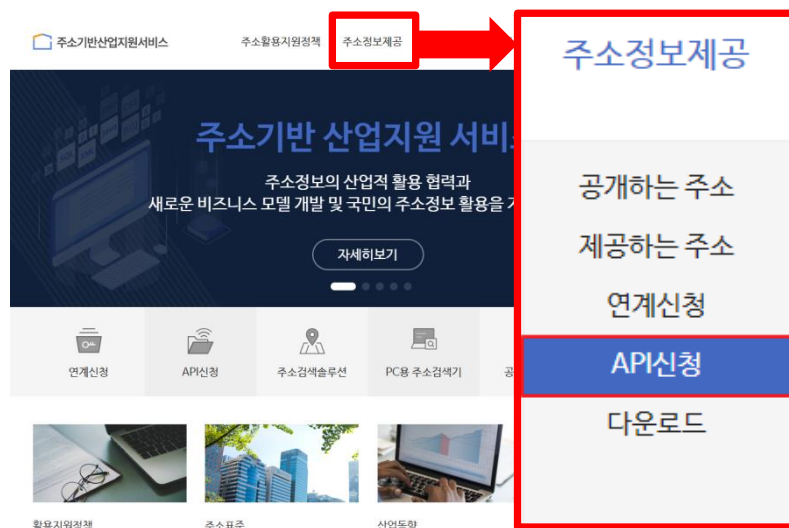


## 이용 신청하기

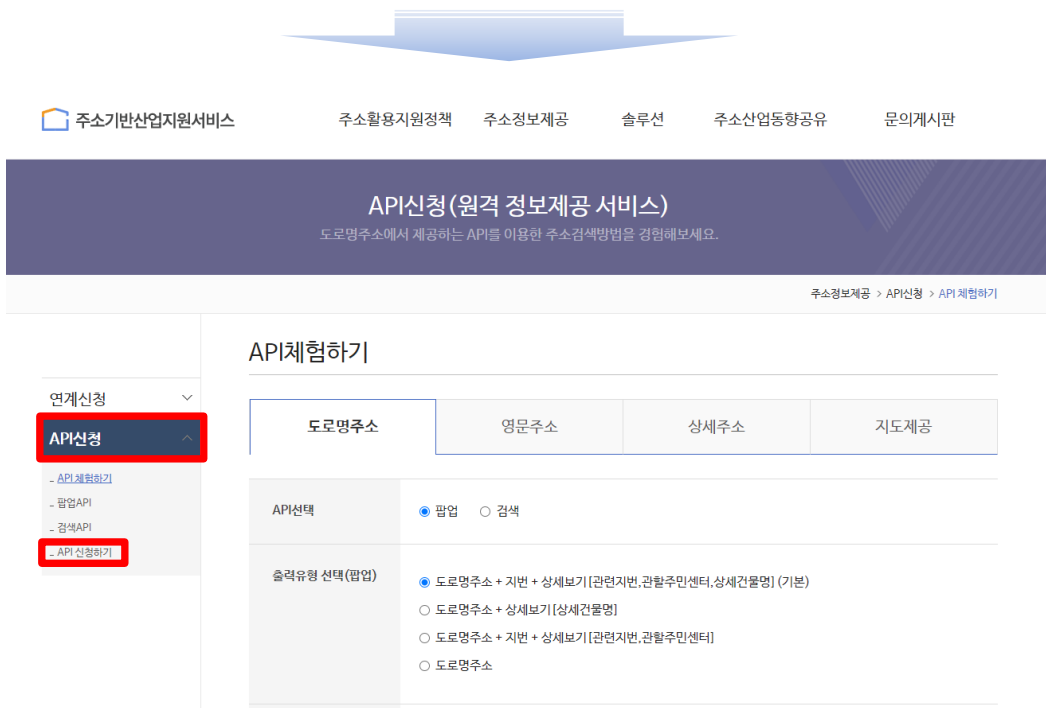
도로명주소 안내 홈페이지(business.juso.go.kr)에서 API 연계 서비스를 신청하고, 처리내역을 확인합니다.

### 3.1. 신청하기

#### ◆ 메뉴경로



✓ 주소정보제공 메뉴의 API신청을 클릭합니다.





## 이용 신청하기

## ◆ 신청등록

- ✓ 신청 시 기재사항을 모두 입력한 후 “신청”버튼을 클릭하면, 본인인증 과정을 거쳐 신청이 완료됩니다.  
( 개발승인키 발급시 본인인증 절차없이 승인키를 발급받으실 수 있습니다.)
- ✓ URL, IP가 다수인 경우 ‘활용시스템 리스트’ 양식을 다운로드 하여 다수의 URL, IP를 기재 후 첨부해주시면 각각 URL, IP 별로 승인 키를 발행해 드립니다.  
예) 서브도메인(sub.test.co.kr, new.test.co.kr) 사용 등, 복수 도메인인 경우에 해당합니다.
- ✓ 승인 키는 URL(또는 IP) 기준으로 부여되므로 다수의 사이트(또는 Application)에서 사용할 경우 각각 부여된 승인 키를 활용하셔야 하며, 그렇지 않을 경우 서비스가 중지될 수 있습니다.



## 도로명 3

## 이용 신청하기

### 3.2. 나의 신청내역 열람 및 변경

#### ◆ 메뉴경로

API신청하기 API신청 변동사항 안내

연계신청  
API신청  
- API 체험하기  
- 팝업API  
- 검색API  
- API신청하기

표시는 필수 입력

기본정보

API 종류  
☒ 공개 도로명주소 API  
☐ 제공 좌표제공 API  
☐ 제공 지도제공 API  
☐ 공개 영문주소 API  
☐ 공개 상세주소 API

API 유형  
☒ 팝업 API  
☐ 검색 API  
☐ 모바일팝업 API

신청기관 유형  
☐ 공공기관  
☐ 민간기관

시스템명  
예) 도로명주소 유통 시스템

업체(기관)명  
예) 한국지역정보개발원

URL(IP)  
예) juso.go.kr

신청내역 >

✓ “신청내역” 기능을 이용하여 본인의 신청내용이 정상적으로 처리되었음을 확인하실 수 있습니다.

✓ “신청내역” 신청내역 > 버튼 클릭 시 신청내역화면으로 이동합니다.

주소정보제공 > API신청 > API 신청하기

API신청내역

좌표정보를 제공하는 연계정보 수령 시, 주소정보 제공 **인수확인 및 보안각서** 작성이 필요합니다.

번호	기관명(시스템명)	신청일자	승인상태	연계정보
등록된 내역이 없습니다.				

<< < 1 > >>

승인키 검색어를 입력해주세요. 검색

✓ 신청자와 열람자가 다른 경우 승인 키로 신청내역을 열람하실 수 있습니다.

✓ 신청내역이 조회된 후 특정 업체명(기관명)을 클릭하시면 상세보기 화면이 나타납니다.





## 신청내역 상세

API 유형	도로명주소 검색 API		
업체기관 유형	민간기관	업체(기관)명	도로명주소 도움센터
시스템명	도로명주소 도움센터	URL (IP)	juso.go.kr
시스템 개요	도로명주소 도움센터		
승인키	TESTJUSOGOKR		
서비스망	인터넷망		
CSS 파일경로			

- 이후 절차는 '개방주소정보' API신청 > 팝업API 또는 검색API 또는 앱사 제공되는 가이드를 참고하시기 바랍니다.

## 목록

- ✓ 신청내역 상세보기에서는 URL에 부여된 '**승인키**'를 확인하실 수 있습니다.
- ✓ **승인완료**일 경우에는 예시와 같이 승인키 항목에 승인키가 조회됩니다.
- ✓ "**목록**" 버튼을 클릭하면 이전의 목록화면으로 이동합니다.
- ✓ "**수정**" 버튼은 본인인증 후 활성화 됩니다.

## 신청내역 수정

API 유형	도로명주소 검색 API		
▪ 신청기관 유형	<input type="radio"/> 공공기관 <input checked="" type="radio"/> 민간기관	▪ 업체(기관) 명	도로명주소 도움센터
▪ 시스템명	도로명주소 도움센터	▪ URL (IP)	juso.go.kr
시스템 개요	<div>도로명주소 도움센터</div>		
서비스망	인터넷망		
CSS 파일경로	http://www.juso.go.kr/css/addrlinkSample.css		

최소



## [참고] API 차단 사례

### 4.1. 잘 못된 검색어 요청시 차단

차단 보안규칙 패턴	차 단 사 유
'+or+'1'='1	Sql Injection 공격 관련 패턴 탐지에 따른 차단
' and 'f'='f	Sql Injection 공격 관련 패턴 탐지에 따른 차단
%27+and+	Sql Injection 공격 관련 패턴 탐지에 따른 차단
UNION+SELECT	Sql Injection 공격 관련 패턴 탐지에 따른 차단
%27+and+%27x%27%3D	SQL 쿼리를 삽입하여 로그인 우회 가능한 패턴으로 차단

## 조치방안

❖ API 호출 전에 검색어(keyword) 정제를 통해서 제거 및 안내메세지 출력

✓ or, select, insert, delete, update, create, drop, exec, union, fetch, declare, truncate 등 SQL 예약어

✓ < , > , = , % 등 특수문자