System/Service Connection - OPEN-API Connection Service

Summary

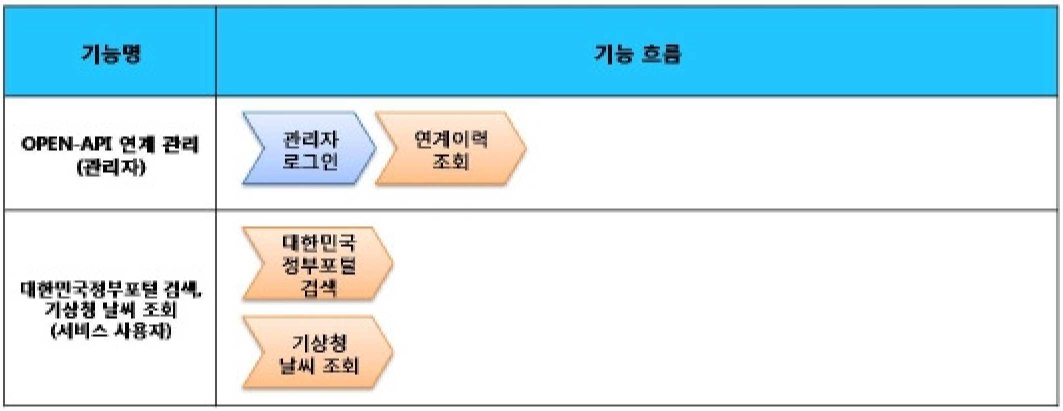
OPEN-API Connection Service is a mobile UX-based component that connects various Open-API services.
OPEN-API is a typical Web 2.0 technology that has implemented “a web as a platform” and is referred to as
a programming interface which is open to allow various functions to be used in a single web site.

As samples of this component, RESTFul type search service, SOAP type weather inquiry service and administrator OPEN-API inquiry history inquiry functions are provided.
In addition, connection with other Open-APIs is also available.

- Republic of Korea (ROK) Government portal search API: use Open-API to provide the search results provided by ROK government integrated search service
- Korea Meteorological Administration (KMA) weather inquiry API: use Open-API to provide various weather information provided by kMA

For the functions of OPEN-APIs provided by outside institutes, contact those institutes.

* Flow chart



기능명	기능 흐름
OPEN-API 연계 관리 (관리자)	관리자 로그인 → 연계이력 조회
대한민국정부포털 검색, 기상청 날씨 조회 (서비스 사용자)	대한민국 정부포털 검색 기상청 날씨 조회

This component has been developed and tested on the browsers available in the mobile devices.
Since the browser support is different depending the device (PC or mobile device), development and application should be carefully carried out.

The test devices include Galaxy S2, Galaxy S, Galaxy Tab 1, IPad2, iPhone 4.
The test mobile browsers include Android basic browser, Firefox, Safari and Opera Mobile.

The results are as follows.

Android
For the Android web browser, Firefox(6.xx), opera mobile, in android OS 2.3(gengerbread), the execution of HTML5, CSS3 and components is supported stably.
However, in Android 2.1, 2.2, the previous version of Android OS 2.3 (gingerbread), the support for HTML5 and CSS3 is not sufficient to implement the component UI properly.
Opera Mobile was excluded from the support as it caused some problems with handling colors and events.
For testing, Galaxy, Galaxy S2 and Galaxy Tab 1 were used.

iOS
iOS is one of the three supported browsers and both version (iOS 4.2, 4.3) supported HTML5 and CSS3 relatively well.
iPhone4 and iPad2 were used for testing.

Prerequisites

In order to use th mobile OPEN-API connection service, apply for OPEN-API service separately to the OPEN-API service provider.

The service application procedure is as follows.

① Application and submission: apply for OPEN-API
② Review/approval: application is reviewed and approved
③ Issuance of key(email): the applicant emailed the authentication key(or ID/password)
④ Institute user: use the issued authentication key(or user ID/password) to call Open API

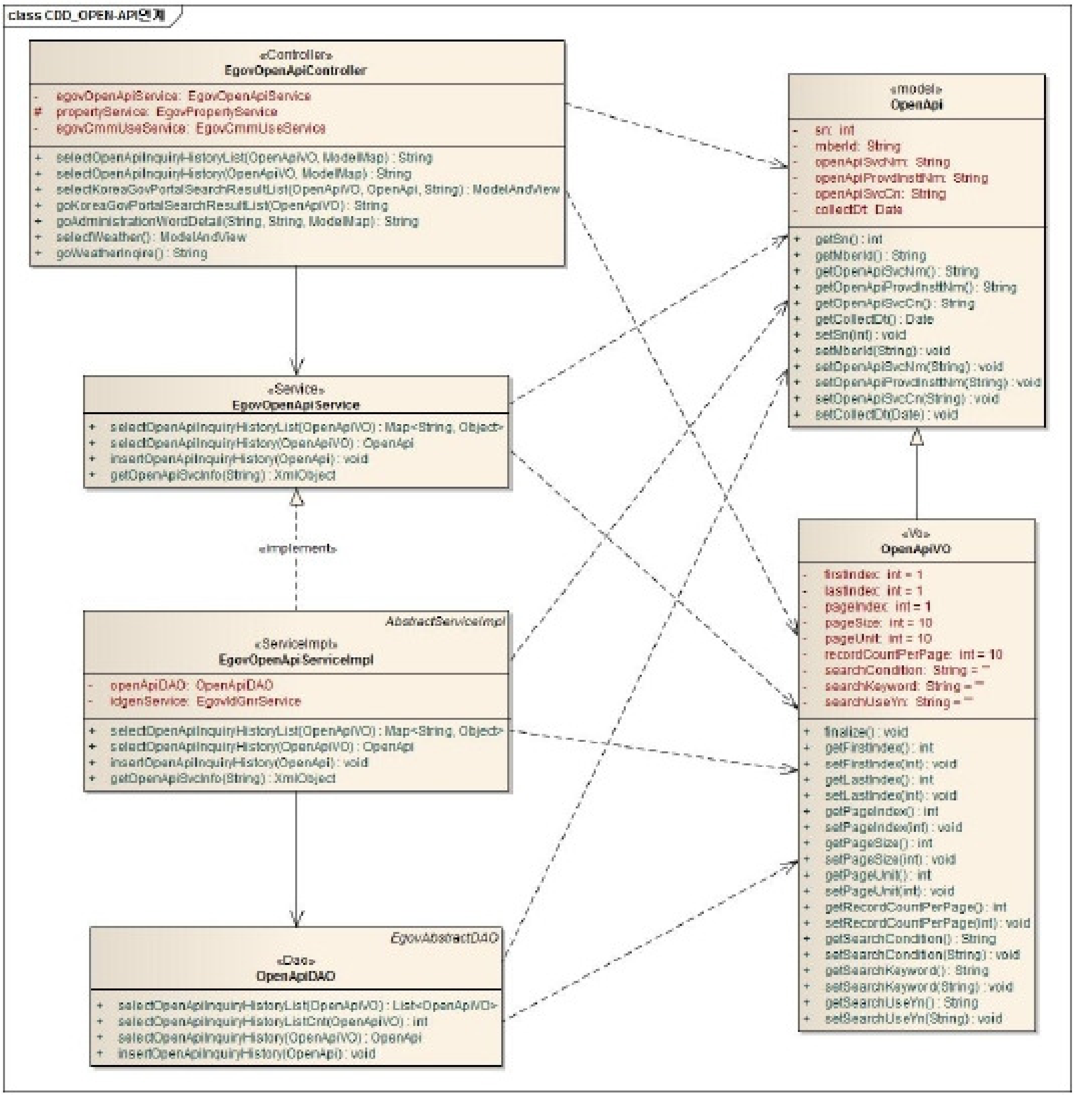
- Apply for ROKURL Portal search service : http://www.korea.go.kr/ptl/addservice/openapi/insertOpenApiView.do
[http://www.korea.go.kr/ptl/addservice/openapi/insertOpenApiView.do]
- Contact for KMA weather inquiry service: Department of Information Communication Technology (☎ 02-2181-0439), Contact E-Mail(lian1071@korea.kr)

Currently you can use OPEN-API by applying the ID/PW issued through the procedures of application-review-approval to the environmental setup file.
For details on how to apply ID/PW, refer to the environment setup.

Description

Mobile OPEN-API connection service is largely composed of OPEN-API inquiry history list inquiry and detail inquiry, ROK Government Portal Search, and KMA Weather Inquiry.(refer to the relevant fuctions)

Class diagram


class CDD_OPEN-API연계
«Controller»
EgovOpenApiController
- egovOpenApiService: EgovOpenApiService
# propertyService: EgovPropertyService
- egovCmmUseService: EgovCmmUseService
+ selectOpenApiInquiryHistoryList(OpenApiVO, ModelMap) : String
+ selectOpenApiInquiryHistory(OpenApiVO, ModelMap) : String
+ selectKoreaGovPortalSearchResultList(OpenApiVO, OpenApi, String) : ModelAndView
+ goKoreaGovPortalSearchResultList(OpenApiVO) : String
+ goAdministrationWordDetail(String, String, ModelMap) : String
+ selectWeather() : ModelAndView
+ goWeatherInqire() : String
«Service»
EgovOpenApiService
+ selectOpenApiInquiryHistoryList(OpenApiVO) : Map<String, Object>
+ selectOpenApiInquiryHistory(OpenApiVO) : OpenApi
+ insertOpenApiInquiryHistory(OpenApi) : void
+ getOpenApiSvcInfo(String) : XmlObject
«implement»
AbstractServiceImpl
«ServiceImpl»
EgovOpenApiServiceImpl
- openApiDAO: OpenApiDAO
- idgenService: EgovIdGnrService
+ selectOpenApiInquiryHistoryList(OpenApiVO) : Map<String, Object>
+ selectOpenApiInquiryHistory(OpenApiVO) : OpenApi
+ insertOpenApiInquiryHistory(OpenApi) : void
+ getOpenApiSvcInfo(String) : XmlObject
EgovAbstractDAO
«Dao»
OpenApiDAO
+ selectOpenApiInquiryHistoryList(OpenApiVO) : List<OpenApiVO>
+ selectOpenApiInquiryHistoryListCnt(OpenApiVO) : int
+ selectOpenApiInquiryHistory(OpenApiVO) : OpenApi
+ insertOpenApiInquiryHistory(OpenApi) : void
«model»
OpenApi
- sn: int
- mberId: String
- openApiSvcNm: String
- openApiProvdInsttNm: String
- openApiSvcCn: String
- collectDt: Date
+ getSn() : int
+ getMberId() : String
+ getOpenApiSvcNm() : String
+ getOpenApiProvdInsttNm() : String
+ getOpenApiSvcCn() : String
+ getCollectDt() : Date
+ setSn(int) : void
+ setMberId(String) : void
+ setOpenApiSvcNm(String) : void
+ setOpenApiProvdInsttNm(String) : void
+ setOpenApiSvcCn(String) : void
+ setCollectDt(Date) : void
«Vo»
OpenApiVO
- firstIndex: int = 1
- lastIndex: int = 1
- pageIndex: int = 1
- pageSize: int = 10
- pageUnit: int = 10
- recordCountPerPage: int = 10
- searchCondition: String = ""
- searchKeyword: String = ""
- searchUseYn: String = ""
+ finalize() : void
+ getFirstIndex() : int
+ setFirstIndex(int) : void
+ getLastIndex() : int
+ setLastIndex(int) : void
+ getPageIndex() : int
+ setPageIndex(int) : void
+ getPageSize() : int
+ setPageSize(int) : void
+ getPageUnit() : int
+ setPageUnit(int) : void
+ getRecordCountPerPage() : int
+ setRecordCountPerPage(int) : void
+ getSearchCondition() : String
+ setSearchCondition(String) : void
+ getSearchKeyword() : String
+ setSearchKeyword(String) : void
+ getSearchUseYn() : String
+ setSearchUseYn(String) : void

Related sources

Types	Source names	Notes
C ontroller	egovframework.mbl.com.oas.web.EgovOpenApiC ontroller.java	OPEN-API Connection C ontroller C lass
Service	egovframework.mbl.com.oas.service.EgovOpenApiService.java	OPEN-API Connection Service C lass
ServiceImpl	egovframework.mbl.com.oas.service.impl.EgovOpenApiServiceImpl.java	OPEN-API ConnectionServiceImpl C lass
Model	egovframework.mbl.com.oas.service.OpenApi.java	OPEN-API Connection Model C lass
VO	egovframework.mbl.com.oas.service.OpenApiVO.java	OPEN-API Connection VO C lass
DAO	egovframework.mbl.com.oas.service.impl.OpenApiDAO.java	OPEN-API Connection Dao C lass
JSP	/WEB-INF/jsp/egovframework/mbl/com/oas/EgovOpenApiInquiryHistoryList.jsp	OPEN-API Inquiry History List Inquiry Page
JSP	/WEB-INF/jsp/egovframework/mbl/com/oas/EgovOpenApiInquiryHistoryDetail.jsp	OPEN-API Inquiry History Detail Inquiry Page
JSP	/WEB-INF/jsp/egovframework/mbl/com/oas/EgovMobileKoreaGovPortalSearchResultList.jsp	Mobile ROK Government Portal Search Page
JSP	/WEB-INF/jsp/egovframework/mbl/com/oas/EgovMobileKoreaGovPortalSearchResultDetail.jsp	Mobile ROK Government Portal Search Detail Inquiry Page
JSP	/WEB-INF/jsp/egovframework/mbl/com/oas/EgovMobileWeatherInqire.jsp	Mobile KMA Weather Inquiry Page
QUERY XML	resources/egovframework/sqlmap/mbl/com/oas/EgovOpenApi_SQL_XXX.xml	OPEN-API Connection QUERY XML
Message properties	resources/egovframework/message/message-common_ko_KR.properties	OPEN-API Connection Message properties
Idgen XML	resources/egovframework/spring/com/context-idgen.xml	OPEN-API Connection History Id GenerationIdgen XML

ID Generation

- In order to use ID Generation Service, add COMTECOPSEQ에 OAS_ID (OPEN-API Inquiry History ID)entry to COMTECOPSEQ, the sequence storing table.

Related tables

Table name	Table name (English)	Notes
OPEN-API Inquiry history	C OMTNOPENAPI	OPEN-API inquiry history
Common detail code	C OMTC C MMNDETAILC ODE	OPEN-API service classification

Login/membership management/permission management

OPEN-API connection service requires the user/administer authentication in ROK Government Portal Search, KMA Weather Inquiry and OPEN-API Connection Information List Inquiry and detail Inquiry.

For authentication, you can use connect to login, membership management, and permission management component. To use Spring Security for authentication, make changes as in the following.

src/main/java/resources/egovframework/spring/com/context-common.xml

```
<!—Service class that can check the authenticated user's LoginVO, permissions and authentication-->
<bean id="egovUserDetailsHelper" class="egovframework.com.cmm.util.EgovUserDetailsHelper">
        <property name="egovUserDetailsService">
                <ref bean="egovUserDetailsSecurityService" />
        </property>
</bean>

<!—Bean that will use Security-based authentication -->
```

```
    <!-- Bean that will use Security-based authentication -->
    <bean id="egovUserDetailsSecurityService" class="egovframework.com.sec.ram.service.impl.EgovUserDetailsSecurityServiceImpl"/>
```

web.xml

```
    <!-- 2. Spring Security Setup-->
    <filter>
        <filter-name>springSecurityFilterChain</filter-name>
        <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>springSecurityFilterChain</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>

    <listener>
        <listener-class>org.springframework.security.ui.session.HttpSessionEventPublisher</listener-class>
    </listener>
    <!-- security end -->

    <!-- 4. Filter setup for Spring Security logout setup -->
    <filter>
        <filter-name>EgovSpringSecurityLogoutFilter</filter-name>
        <filter-class>egovframework.com.sec.security.filter.EgovSpringSecurityLogoutFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>EgovSpringSecurityLogoutFilter</filter-name>
        <url-pattern>/uat/uia/actionLogout.do</url-pattern>
    </filter-mapping>

    <!-- 4Filter setup for Spring Security login setup -->
    <filter>
        <filter-name>EgovSpringSecurityLoginFilter</filter-name>
        <filter-class>egovframework.com.sec.security.filter.EgovSpringSecurityLoginFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>EgovSpringSecurityLoginFilter</filter-name>
        <url-pattern>*.do</url-pattern>
    </filter-mapping>
```

As in the above, add Spring Security filter to web.xml.

For details on login, membership management, permission management, refer to the followings.

- Login: general login
- Membership management: user management
- Permission management: permission management

The following pages need the user/administrator permissions.

Permission	type Page path	Description
User	/mbl/com/oas/goKoreaGovPortalSearchResultList.mdo	ROK Government Portal search screen
User	/mbl/com/oas/goAdministrationWordDetail.mdo	ROK Government Portal administration word detail screen
User	/mbl/com/oas/goWeatherInqire.mdo	KMA Weather Inquiry screen
User	/mbl/com/oas/selectOpenApiInquiryHistoryList.mdo	OPEN-API connection information list inquiry screen
User	/mbl/com/oas/selectOpenApiInquiryHistory.mdo	OPEN-API connection information detail Inquiry screen

Settings and Notes

* The currently distributed mobile common component adopts HTML5 and CSS3 and this means that it doesn’t support the standards for mobile web site validity check of MobileOK and W3C. For details on correct validity check, please contact the competent professional institute.
* Security verification is under progress on the currently distributed mobile component. We plan to complete security verification as soon as possible to release a new security-patched version.

Environmental settings

The followings are the required items and environmental settings to use Mobile OPEN-API connection service functions.

- Property files settings (egovframework/egovProps/conf/oas.properties)

```
#-----------------------------------------------------------------------
#
#     oas.properties:Config file that shows OPEN-API connection authentication key information
#
#-----------------------------------------------------------------------
#    1.    key = value structure.
#    2.    Key values cannot include space. Value can include space
#    3.    Key values cannot include Korean,  Value can include Korean
#    4.    If a new line is needed, add '\' to the end of the line (if           you have to use '\', then use '\\'.)
#    5.    In Windows, the directory symbol should be: '\\' or '/' (do not use '\'.)
#    6.    Directory symbol in Unix: '/'
#    7.    For comments,    #
           use
#    8.    If there is space after values, an error could occur when referenced by sublets. Use trim() or set the properties values without space.
#-----------------------------------------------------------------------


# ROK Government Portal search service authentication key
koreaGovPortalSearchKey = issued authentication key

# KMA Weather Inquiry service authentication ID
kmaWeatherInquiryId = issued authentication ID

# KMA Weather Inquiry service authentication password
kmaWeatherInquiryPassword = issued authentication password
```

In the above, enter the issued authentication information to koreaGovPortalSearchKey, kmaWeatherInquiryId andkmaWeatherInquiryPasswordentries.

- Korean encoding setting (server.xml)

To prevent Korean from being broken in JSP pages, addURIEncoding=“UTF-8” to the end of Connector tage in the server.xmlfile.

```
<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1" redirectPort="8443" URIEncoding="UTF-8"/>
...
...
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" URIEncoding="UTF-8"/>
```

Related functions

Related functions

Mobile OPEN-APIconnection service is largely composed of ROK Government Portal Search for general users,Mobile ROK Government Portal SearchDetail Inquiry ,Mobile KMA Weather Inquiry, OPEN-API connection hisitry list inquiry for administrators and OPEN-APIconnenction historyDetail Inquiry.

Mobile ROK Government Portal Search

Prior preparations

In order to use ROK Government Portal search service, make the following prepareations.

1. Secure OPEN-APIservice requiest results XML
2. XML Schema(XSD) based on XML
3. Create JAVA Class files that will map the OPEN-API service requiest results based on the written XML Schema
4. Import Jar file, which is the bundle of the created JAVA class files, to the project.

* OPEN-API service request results XML

```
<?xml version="1.0" encoding="UTF-8"?>

<search>
        <conditions>
                <query>eGovernment</query>
                <collection>site</collection>
                <searchType>r</searchType>
                <searchOrder>1</searchOrder>
                <pageNum>1</pageNum>
                <pageSize>5</pageSize>
        </conditions>
        <totalResultstotalCount="20">
                <collection id="site">
                        <results count="20"></results>
                        <resource>
                                <title><![CDATA[<b>eGovernment</b>infrastructure integration service]]></title>
                                <contents><![CDATA[<b>eGovernment</b> In order to provide convenicne in carrying out the tasks,<b>eGovernment</b>an integration service for infrastructure system is
                                provided다.]]></contents
                                <organ><![CDATA[Ministry of Public Administration and Security]]></organ>
                                <linkurl><![CDATA[http://www.infra.go.kr]]></linkurl>
                        </resource>
                        <resource>
                                <title><![CDATA[<b>eGovernment</b>Stnadards Management System]]></title>
                                <contents><![CDATA[<b>eGovernment</b>Set of codes for documented stadnards, specifications, guideline and technological regulations.]]></contents>
                                <organ><![CDATA[Ministry of Public Administration and Security]]></organ>
                                <linkurl><![CDATA[http://www.guide.go.kr]]></linkurl>
                        </resource>
                        <resource>
                                <title><![CDATA[eGovernment<b> for Foreigners</b>]]></title>
                                <contents><![CDATA[<b>Electronic</b>civil complaints, visit reservation, information forum, employment support, investment promotion guide service.]]></contents>
                                <organ><![CDATA[Ministry of Justice]]></organ>
                                <linkurl><![CDATA[http://www.hikorea.go.kr/]]></linkurl>
                        </resource>
                        ...
                        ...
                        ...
                        ...
                </collection>
        </totalResults>
</search>
```

* XML Schema(XSD)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schemaxmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="search">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="conditions"/>
        <xs:element ref="totalResults"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="conditions">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="query"/>
        <xs:element ref="collection"/>
        <xs:element ref="searchType"/>
        <xs:element ref="searchOrder"/>
        <xs:element ref="pageNum"/>
        <xs:element ref="pageSize"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="query" type="xs:string"/>
  <xs:element name="searchType" type="xs:string"/>
  <xs:element name="searchOrder" type="xs:string"/>
  <xs:element name="pageNum" type="xs:string"/>
  <xs:element name="pageSize" type="xs:string"/>
  <xs:element name="totalResults">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="collection"/>
      </xs:sequence>
      <xs:attribute name="totalCount" use="required" type="xs:string"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="collection">
    <xs:complexType mixed="true">
      <xs:choiceminOccurs="0" maxOccurs="unbounded">
        <xs:element ref="resource"/>
        <xs:element ref="results"/>
      </xs:choice>
      <xs:attribute name="id" type="xs:string"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="resource">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="title"/>
        <xs:element ref="contents"/>
        <xs:element ref="organ"/>
        <xs:element ref="linkurl"/>
        <xs:element ref="platform"/>
```

```
                <xs:element ref="platform"/>
                <xs:element ref="hits"/>
                <xs:element ref="downloads"/>
                <xs:element ref="capacity"/>
                <xs:element ref="developer"/>
                <xs:element ref="registdate"/>
                <xs:element ref="modifydate"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="contents" type="xs:string"/>
    <xs:element name="organ" type="xs:string"/>
    <xs:element name="linkurl" type="xs:string"/>
    <xs:element name="platform" type="xs:string"/>
    <xs:element name="hits" type="xs:string"/>
    <xs:element name="downloads" type="xs:string"/>
    <xs:element name="capacity" type="xs:string"/>
    <xs:element name="developer" type="xs:string"/>
    <xs:element name="registdate" type="xs:string"/>
    <xs:element name="modifydate" type="xs:string"/>
    <xs:element name="results">
        <xs:complexType>
            <xs:attribute name="count" use="required" type="xs:string"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

* Create Java Class file that will map the OPEN-APIservice request results

In order to create Java Class file that will map the OPEN-API service request results, use XMLBean(http://xmlbeans.apache.org [http://xmlbeans.apache.org]) provided by Apache. Install XMLBean and carry out the following command on the console.

```
scomp -out KoreaGovPortalSearch.jar KoreaGovPortalSearch.xsd
```

Executing the above command creates KoreaGovPortalSearch.jarfile. Add this file to the project.
The currently distributed file includes KoreaGovPortalSearch.jarand doesn’t need above procedure.

*Applying to the source

```
// Map the values to service-specific objects that generated by using xsdfile
SearchDocumentsDoc = null;
sDoc = (SearchDocument)egovOpenApiService.getOpenApiSvcInfo(requestUrl);
Search search = sDoc.getSearch();
TotalResultstotalResult = search.getTotalResults();
noNamespace.ResourceDocument.Resource[] resourceList = totalResult.getCollection().getResourceArray();

for (noNamespace.ResourceDocument.Resource resource:resourceList) {
      System.out.println("Title = " + resource.getTitle());
      System.out.println("Contents = " + resource.getContents());
      System.out.println("Linkurl = " + resource.getLinkurl());
}
```

1. Create XXXXDocumentobject that inheritsXmlObject. (XXXX is XML’s root name. ROK Government Portal search service results XML root is Search.)
requestUrl is the URL of the provided OPEN-API service.
2.In the generated object, use getXXXX()method to gain the root. (similarly, XXXX is the root name of XML)
3.In the root object, usegetXXXXX()method to gain the element to be accessed. (XXXX is the element name to be accessed)
4. Generate noNamespace.XXXXDocument.XXXXobject and use getXXXXArray() method to store data in array.
(XXXX is higher element of repeated data. ROK Government Portal search OPEN-API is a lower element of resource. It includes title, contents,linkurl, etc.)

*Others

In order to use other functions and options in addition to the function provided as samples, refer to ROK Government Portal Open APIuser manual'.
(ROK Government Portal Open API User Manual : http://www.korea.go.kr/ptl/addservice/openapi/selectOpenApiInfo.do [http://www.korea.go.kr/ptl/addservice/openapi/selectOpenApiInfo.do])

※In order to other OPEN-APIs in addition to ROK Government Portal search OPEN-API, go through the above procedure.

Business rules

You can use the search conditions to search for the information provided by ROK Government Portal. If the search condition is an administration word, Detail Inquiry is available. For the other cases, you will be moved to the relevant URLS.

Every time search is carried out, the search information is automatically registered in OPEN-API Inquiry History table.
When registered, OAS_IDcolumn uses “egovframework.rte.fdl.idgnr.impl.EgovTableIdGnrService”.

```
Primary Key => OAS_ID:serial number is automatically generated.
```

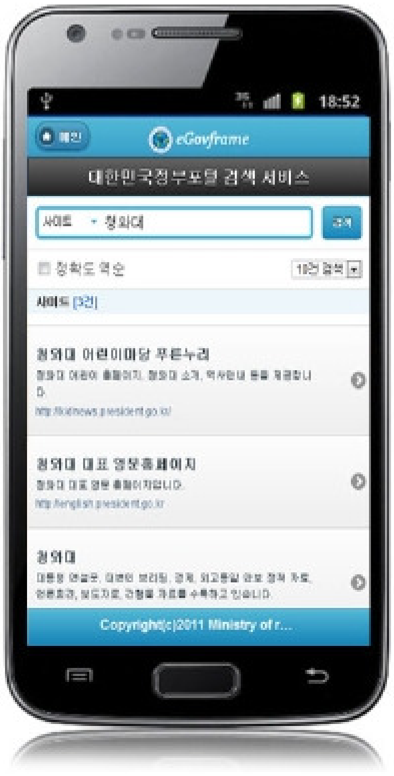
```
        <bean name="egovOpenApiIdGnrService"
                class="egovframework.rte.fdl.idgnr.impl.EgovTableIdGnrService"
                destroy-method="destroy">
                <property name="dataSource" ref="dataSource" />
                <property name="blockSize"          value="10"/>
                <property name="table"              value="COMTECOPSEQ"/>
                <property name="tableName"          value="OAS_ID"/>
        </bean>
```

Related codes

N/A

Screen and execution manual

Action	URL	Controller method	QueryID
Search	/mbl/com/oas/selectKoreaGovPortalSearchResultList.mdo	selectKoreaGovPortalSearchResultList	
Register search history	/mbl/com/oas/selectKoreaGovPortalSearchResultList.mdo	insertOpenApiInquiryHistory	“OpenApiDAO.insertOpenApiInquiryHistory”
Click the list(administration word)	/mbl/com/oas/goAdministrationWordDetail.mdo	goAdministrationWordDetail	



대한민국정부포털 검색 서비스

사이트 ▾ 청와대 검색

□ 정확도 역순 10건 검색 ▾

사이트 [3건]

청와대 어린이마당 푸른누리
청와대 어린이 홈페이지. 청와대 소개, 역사안내 등을 제공합니다.
http://kidnews.president.go.kr/

청와대 대표 영문홈페이지
청와대 대표 영문 홈페이지입니다.
http://english.president.go.kr

청와대
대통령 연설문, 대변인 브리핑, 경제, 외교통일 안보 정책 자료, 언론보도, 보도자료, 간행물 자료를 수록하고 있습니다.

Copyright(c)2011 Ministry of r...

,

Search: In order to search for the information provided by ROK Government Portal, select the search condition at the top, enter a key word and click the search button.
Click list:moves to Mobile ROK Government Portal SearchDetail Inquiryscreen.(only when search condition is an administration word)

Mobile ROK Government Portal Search Detail Inquiry

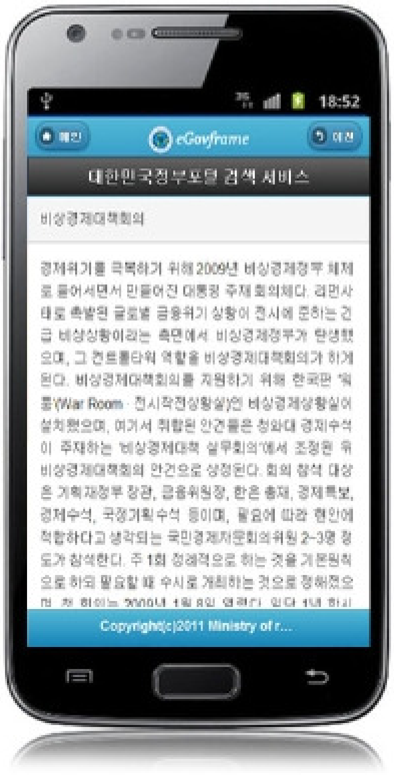
Business rules

By clicking an administration word on the Mobile ROK Government Portal search screen, you can move to this screen. It shows detailed information on the administration search information.

Related code

N/A

Screen and execution manual



대한민국정부포털 검색 서비스

비상경제대책회의

경제위기를 극복하기 위해 2009년 비상경제정부 체제로 들어서면서 만들어진 대통령 주재 회의체다. 리먼사태로 촉발된 글로벌 금융위기 상황이 전시에 준하는 긴급 비상상황이라는 측면에서 비상경제정부가 탄생했으며, 그 컨트롤타워 역할을 비상경제대책회의가 하게 된다. 비상경제대책회의를 지원하기 위해 한국판 '워룸(War Room · 전시작전상황실)'인 비상경제상황실이 설치됐으며, 여기서 취합된 안건들은 청와대 경제수석이 주재하는 '비상경제대책 실무회의'에서 조정된 뒤 비상경제대책회의 안건으로 상정된다. 회의 참석 대상은 기획재정부 장관, 금융위원장, 한은 총재, 경제특보, 경제수석, 국정기획수석 등이며, 필요에 따라 현안에 적합하다고 생각되는 국민경제자문회의위원 2~3명 정도가 참석한다. 주 1회 정례적으로 하는 것을 기본원칙으로 하되 필요할 때 수시로 개최하는 것으로 정해졌으며, 첫 회의는 2009년 1월 8일 열렸다.

Copyright(c)2011 Ministry of r...

Mobile KMA Weather Inquiry

Prior preparations

To use Mobile KMA Weather Inquiry service, the following preparations should be made.

To use Mobile KMA Weather Inquiry service, the following preparations should be made.
※The following process was set up based on the guide provided by KMA.

1.Use the wsdl of the web service, which is open on the web, to generate Client Stub Classfile.
2.Import the Jar file, a bundle of the created JAVA Class files, to the project

* Generate Client Stub Classfile

Use the WSDL of the web service, which is open on the web, to create Client Stub.
Refer to WSDLdefinition page (http://websky.kma.go.kr [http://websky.kma.go.kr]) provided by KMA.
The service used in the component example a weather web servcie.

In order to generate Client Stub, usejant provided by JEUS(http://technet.tmax.co.kr [http://technet.tmax.co.kr]).

* Generate build.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<project name="SurfaceService" default="build" basedir=".">

    <property name="is.app-client.module" value="true" />

    <import file="common-build.xml" />

    <taskdef name="wsdl2java"
             classname="jeus.util.ant.webservices.Wsdl2JavaTask">
        <classpathrefid="jeus.libraries.classpath" />
    </taskdef>

    <target name="-pre-compile">
        <mkdirdir="${build.classes.dir}" />
        <wsdl2java destDir="${build.classes.dir}"
                   verbose="true"
                   mode="gen:client"
                   doCompile="true"
                   noDataBinding="false"
                   package="kma.websky.client.stub.forecast"
                   wsdl="http://websky.kma.go.kr/services/SurfaceService?wsdl">
            <classpathrefid="jeus.libraries.classpath" />
            <classpathrefid="classpath" />
        </wsdl2java>
    </target>

</project>
```

In the above code,modify wsdl="http://websky.kma.go.kr/services/SurfaceService?wsdl"to suit the service to be provided.
(For the service provided by KMA, go to WSDLdefinition page (http://websky.kma.go.kr [http://websky.kma.go.kr]) to get the address.)

Generate build.xml and place it [JEUS installation folder\samples\common] and execute jant in the path where build.xml is located.

```
 C:\build.xml location path>jant
```

If you run jant, build and dist folders are generated in the folder where build.xml is located.
In the dist folder, SurfaceService.jarfile is generated.

To you Client Stub Class(SurfaceService.jar), the following libraries are needed.

```
bootstrap.jar
FastInfoset.jar
javax.jms.jar
jaxrpc-api.jar
jaxrpc-impl.jar
jaxrpc-spi.jar
jaxws-rt.jar
jeus.jar
jeusapi.jar
jeusutil.jar
jeus-ws.jar
saaj-api.jar
saaj-impl.jar
```

The following libraries should be added manually as they are not distributed with the above libraries.
(the libraries are located in JEUSinstallation folder\lib\system)

```
bootstrap.jar
jeus.jar
jeusapi.jar
jeusutil.jar
jeus-ws.jar
```

Business rules

You can retrieve the weather information provided by KMA. Every time the inquiry is made, the inquiry information is automatically registered in OPEN-API Inquiry History table.
When registered, OAS_IDcolumn uses “egovframework.rte.fdl.idgnr.impl.EgovTableIdGnrService”.

```
Primary Key => OAS_ID:serial number automatically generated.
```

```
<bean name="egovOpenApiIdGnrService"
        class="egovframework.rte.fdl.idgnr.impl.EgovTableIdGnrService"
        destroy-method="destroy">
        <property name="dataSource" ref="dataSource" />
        <property name="blockSize"          value="10"/>
        <property name="table"              value="COMTECOPSEQ"/>
        <property name="tableName"          value="OAS_ID"/>
</bean>
```

Related codes

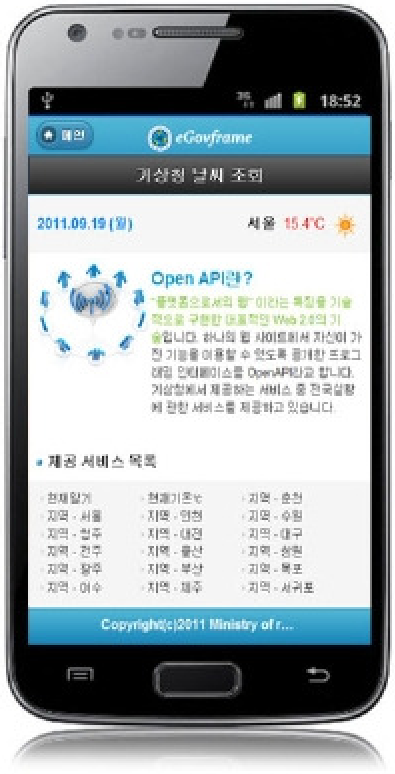
The codes used in Mobile KMA Weather Inquiry and settings are reflected as in the following.

Code classification	code classification name	code ID	code name
C OM086	local code	101	Chuncheon
C OM086	local code	108	Seoul
C OM086	local code	112	Incheon
C OM086	local code	119	Suwon
C OM086	local code	131	Cheongju

C OM086	local code	131	Chongju
C OM086	local code	133	Daejeon
C OM086	local code	143	Daegu
C OM086	local code	146	Jeonju
C OM086	local code	152	Ulsan
C OM086	local code	155	Changwon
C OM086	local code	156	Gwangju
C OM086	local code	159	Busan
C OM086	local code	165	Mokpo
C OM086	local code	168	Yeosu
C OM086	local code	184	Jeju
C OM086	local code	189	Seoguipo

Screen and execution manual

Action	URL	Controller method	QueryID
Weather inquiry	/mbl/com/oas/selectWeather.mdo	selectWeather	
Register search history	/mbl/com/oas/selectKoreaGovPortalSearchResultList.mdo	insertOpenApiInquiryHistory	“OpenApiDAO.insertOpenApiInquiryHistory”



“플랫폼으로서의 웹”이라는 특징을 기술적으로 구현한 대표적인 Web 2.0의 기술입니다. 하나의 웹 사이트에서 자신이 가진 기능을 이용할 수 있도록 공개한 프로그래밍 인터페이스를 OpenAPI라고 합니다. 기상청에서 제공하는 서비스 중 전국실황에 관한 서비스를 제공하고 있습니다.

OPEN-API Inquiry History List Inquiry

Business rules

Use inquiry condition to carry out List Inquiry.

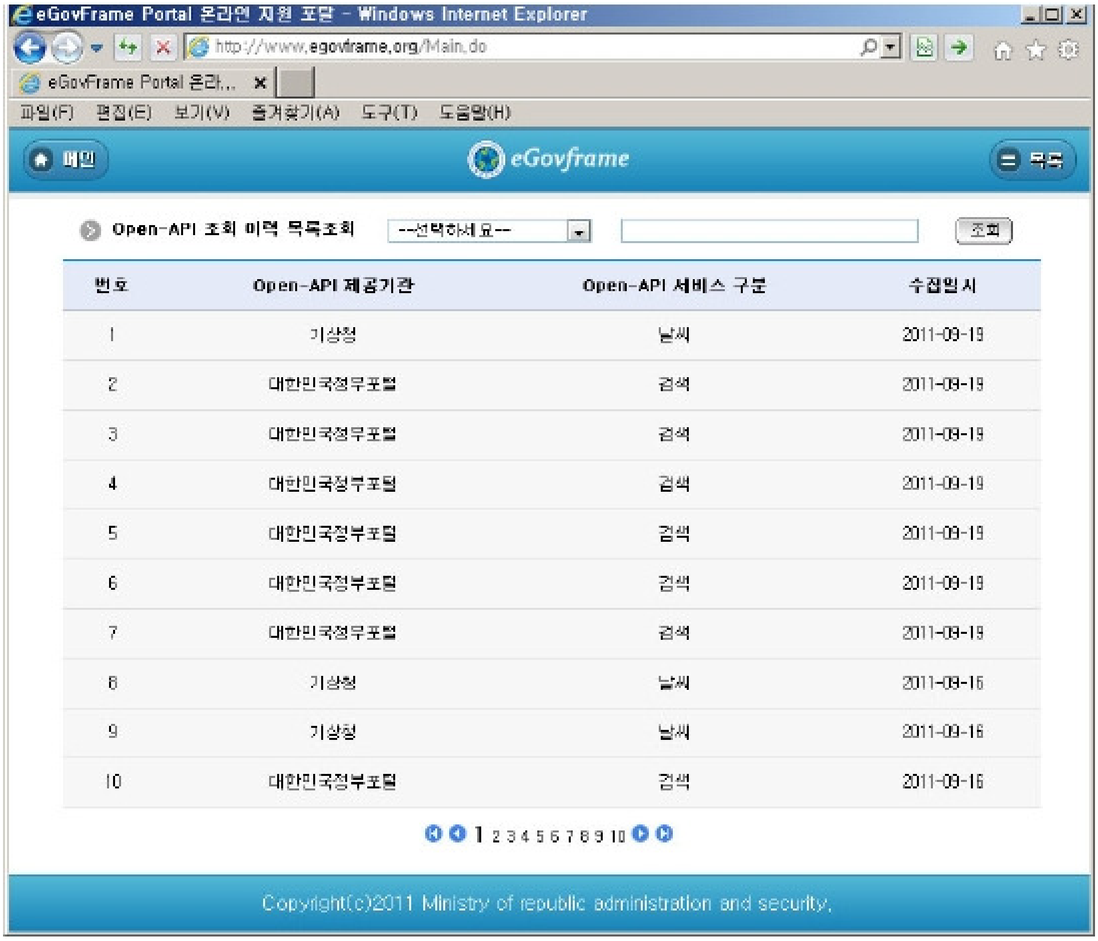
Related codes

Code classification	Code classification name	code ID	Code name
C OM081	OPEN-API provider	APIS001	Weather
C OM081	OPEN-API provider	APIS002	Search
C OM082	OPEN-API service	APIO001	KMA
C OM082	OPEN-API service	APIO002	ROK Government Portal

Screen and execution manual

Action	URL	Controller method	QueryID
Inquiry	/mbl/com/oas/selectOpenApiInquiryHistoryList.mdo	selectOpenApiInquiryHistoryList	“OpenApiDAO.selectOpenApiInquiryHistoryList”
	/mbl/com/oas/selectOpenApiInquiryHistoryList.mdo	selectOpenApiInquiryHistoryListCnt	“OpenApiDAO.selectOpenApiInquiryHistoryListCnt”
Click list	/mbl/com/oas/selectOpenApiInquiryHistory.mdo	selectOpenApiInquiryHistory	“OpenApiDAO.selectOpenApiInquiryHistory”

10 OPEN-API Inquiry History lists are retrieved per page and paging is carried out on a 10 page basis. The search conditions are OPEN-APIprovider and OPEN-APIservice.
To change the search range per page, modify pageUnit and pageSize in context-properties.xmlfile.
(This setting have an influence on the entire common service functions.)


eGovFrame Portal 온라인 지원 포탈 - Windows Internet Explorer
http://www.egovframe.org/Main.do
eGovFrame Portal 온라...
파일(F)
편집(E)
보기(V)
즐겨찾기(A)
도구(T)
도움말(H)
메인
eGovframe
목록

Open-API 조회 이력 목록조회 --선택하세요-- 조회

번호	Open-API 제공기관	Open-API 서비스 구분	수집일시
1	기상청	날씨	2011-09-19
2	대한민국정부포털	검색	2011-09-19
3	대한민국정부포털	검색	2011-09-19
4	대한민국정부포털	검색	2011-09-19
5	대한민국정부포털	검색	2011-09-19
6	대한민국정부포털	검색	2011-09-19
7	대한민국정부포털	검색	2011-09-19
8	기상청	날씨	2011-09-16
9	기상청	날씨	2011-09-16
10	대한민국정부포털	검색	2011-09-16

1 2 3 4 5 6 7 8 9 10

Copyright(c)2011 Ministry of republic administration and security.

Inquiry: in order to OPEN-API Inquiry History, select the search conditions at the top, enter a key word and click the Inquiry button.
Click list: moves to OPEN-API inquiry historyDetail Inquiry.

OPEN-API Inquiry History Detail Inquiry

Business rules

By clicking a list in OPEN-API Inquiry History List Inquiry, you can move to this screen. It shows detailed information on OPEN-API Inquiry History.

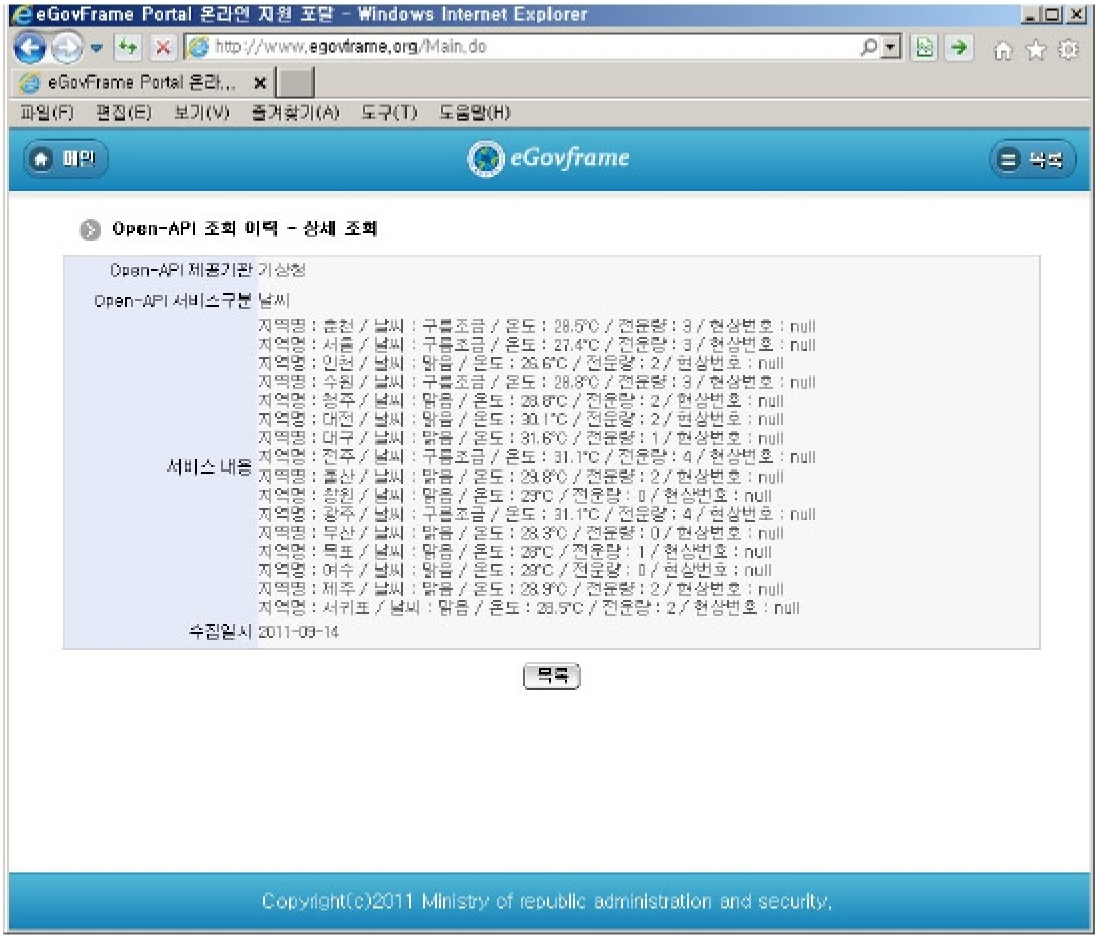
Related codes

N/A

Screen and execution manual

Action	URL	Controller method	QueryID
List	/mbl/com/oas/selectOpenApiInquiryHistoryList.mdo	selectOpenApiInquiryHistoryList	"OpenApiDAO.selectOpenApiInquiryHistoryList"

On the OPEN-API Inquiry History Detail Inquiry, you can carry out OPEN-API Inquiry History List Inquiry.


eGovFrame Portal 온라인 지원 포탈 - Windows Internet Explorer
http://www.egovframe.org/Main.do
eGovFrame Portal 온라... ×
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
메인
eGovframe
목록

Open-API 조회 이력 - 상세 조회

Open-API 제공기관	기상청
Open-API 서비스구분	날씨
서비스 내용	지역명 : 춘천 / 날씨 : 구름조금 / 온도 : 28.5°C / 전운량 : 3 / 현상번호 : null 지역명 : 서울 / 날씨 : 구름조금 / 온도 : 27.4°C / 전운량 : 3 / 현상번호 : null 지역명 : 인천 / 날씨 : 맑음 / 온도 : 26.6°C / 전운량 : 2 / 현상번호 : null 지역명 : 수원 / 날씨 : 구름조금 / 온도 : 28.8°C / 전운량 : 3 / 현상번호 : null 지역명 : 청주 / 날씨 : 맑음 / 온도 : 28.6°C / 전운량 : 2 / 현상번호 : null 지역명 : 대전 / 날씨 : 맑음 / 온도 : 30.1°C / 전운량 : 2 / 현상번호 : null 지역명 : 대구 / 날씨 : 맑음 / 온도 : 31.6°C / 전운량 : 1 / 현상번호 : null 지역명 : 전주 / 날씨 : 구름조금 / 온도 : 31.1°C / 전운량 : 4 / 현상번호 : null 지역명 : 울산 / 날씨 : 맑음 / 온도 : 29.8°C / 전운량 : 2 / 현상번호 : null 지역명 : 창원 / 날씨 : 맑음 / 온도 : 29°C / 전운량 : 0 / 현상번호 : null 지역명 : 광주 / 날씨 : 구름조금 / 온도 : 31.1°C / 전운량 : 4 / 현상번호 : null 지역명 : 부산 / 날씨 : 맑음 / 온도 : 28.3°C / 전운량 : 0 / 현상번호 : null 지역명 : 목포 / 날씨 : 맑음 / 온도 : 28°C / 전운량 : 1 / 현상번호 : null 지역명 : 여수 / 날씨 : 맑음 / 온도 : 29°C / 전운량 : 0 / 현상번호 : null 지역명 : 제주 / 날씨 : 맑음 / 온도 : 28.9°C / 전운량 : 2 / 현상번호 : null 지역명 : 서귀포 / 날씨 : 맑음 / 온도 : 28.5°C / 전운량 : 2 / 현상번호 : null
수집일시	2011-09-14

List: move to OPEN-API inquiry history list inquiry screen.

References

- Refer to the execution environment: ID Generation Service
- Distribution and test: Mobile Common Component Systemanddistribution package plan
- Related database structure and schema:related database structure and schema