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.

ㆍRepublic of Korea (ROK) Government portal search API: use Open-API to provide the search results provided by ROK government integrated search service

ㆍKorea 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

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

②

③

④

Application and submission: apply for OPEN-API

Review/approval: application is reviewed and approved

Issuance of key(email): the applicant emailed the authentication key(or ID/password)

Institute user: use the issued authentication key(or user ID/password) to call Open API

-

Apply for ROK Portal search service

URL

:

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



Related sources

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 management |
| Common detail code | C OMTC C MMNDETAILC ODE | OPEN-API service classification code |

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">

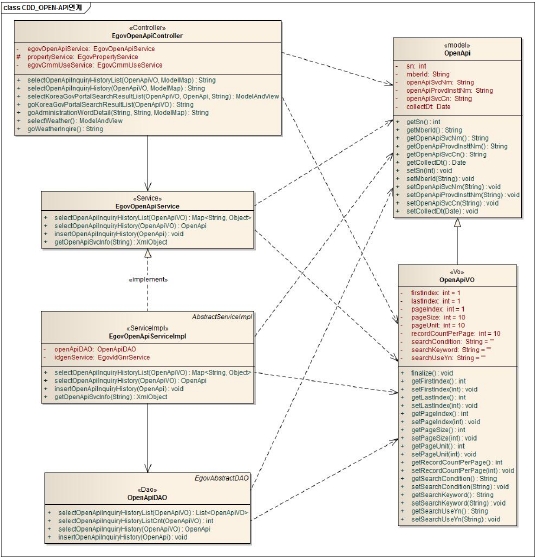
<property name="egovUserDetailsService">

<ref bean="egovUserDetailsSecurityService" />

</property>

</bean>

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



|  |  |  |
| --- | --- | --- |
| 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 Connection ServiceImpl 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 Generation Idgen XML |

<!-- 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

User

User

User

/mbl/com/oas/goAdministrationWordDetail.mdo

/mbl/com/oas/goWeatherInqire.mdo

/mbl/com/oas/selectOpenApiInquiryHistoryList.mdo

/mbl/com/oas/selectOpenApiInquiryHistory.mdo

ROK Government Portal administration word detail screen

KMA Weather Inquiry screen

OPEN-API connection information list inquiry screen

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 Mobile OK 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.

2.

key = value structure.

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.

6.

In Windows, the directory symbol should be: '\\' or '/' (do not use '\'.)

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 and kmaWeatherInquiryPassword entries.

Korean encoding setting (server.xml)

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

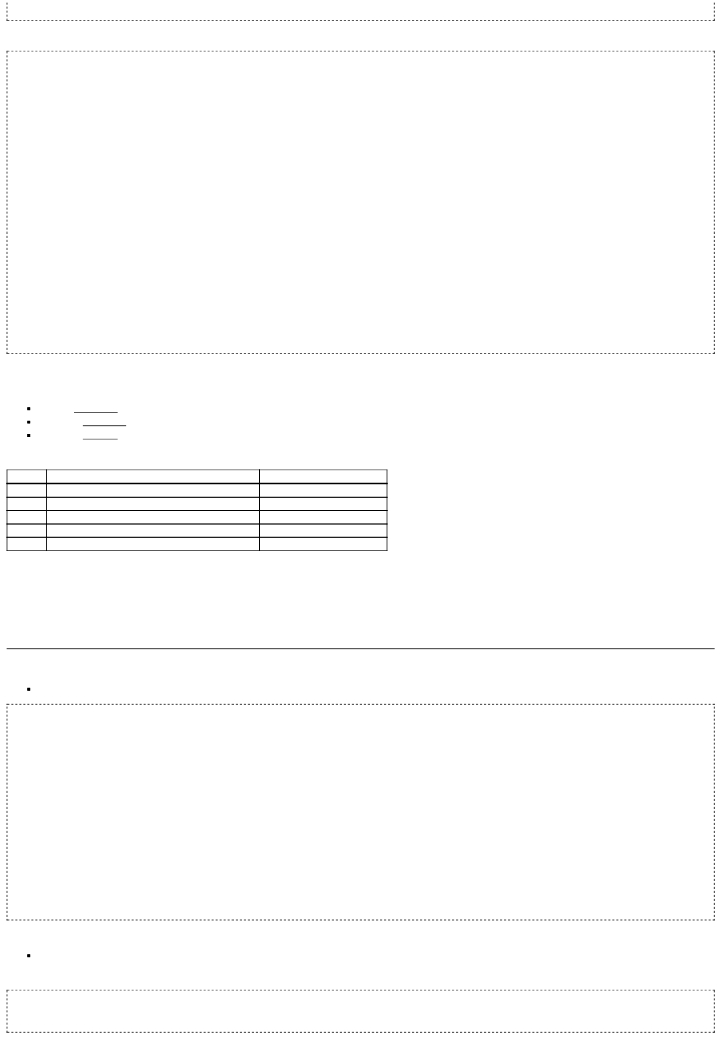
<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1" redirectPort="8443" URIEncoding="UTF-8"/>

...

...

<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" URIEncoding="UTF-8"/>

Related functions



Related functions

Mobile OPEN-API connection 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-API connenction history Detail Inquiry.

Mobile ROK Government Portal Search

Prior preparations

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

1.

2.

3.

4.

Secure OPEN-API service requiest results XML

XML Schema(XSD) based on XML

Create JAVA Class files that will map the OPEN-API service requiest results based on the written XML Schema

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>

<totalResults totalCount="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:schema xmlns: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:choice minOccurs="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-API service 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.jar file. Add this file to the project.

The currently distributed file includes KoreaGovPortalSearch.jar and doesn’t need above procedure.

\* Applying to the source

// Map the values to service-specific objects that generated by using xsd file

SearchDocument sDoc = null;

sDoc = (SearchDocument)egovOpenApiService.getOpenApiSvcInfo(requestUrl);

Search search = sDoc.getSearch();

TotalResults totalResult = 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 XXXXDocument object that inherits XmlObject. (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, use getXXXXX() method to gain the element to be accessed. (XXXX is the element name to be accessed)

4. Generate noNamespace.XXXXDocument.XXXX object 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 API user 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\_ID column 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"

<property name="table"

<property name="tableName"

</bean>

Related codes

N/A

Screen and execution manual

value="10"/>

value="COMTECOPSEQ"/>

value="OAS\_ID"/>

Action

URL

Controller method

QueryID

Search

/mbl/com/oas/selectKoreaGovPortalSearchResultList.mdo selectKoreaGovPortalSearchResultList

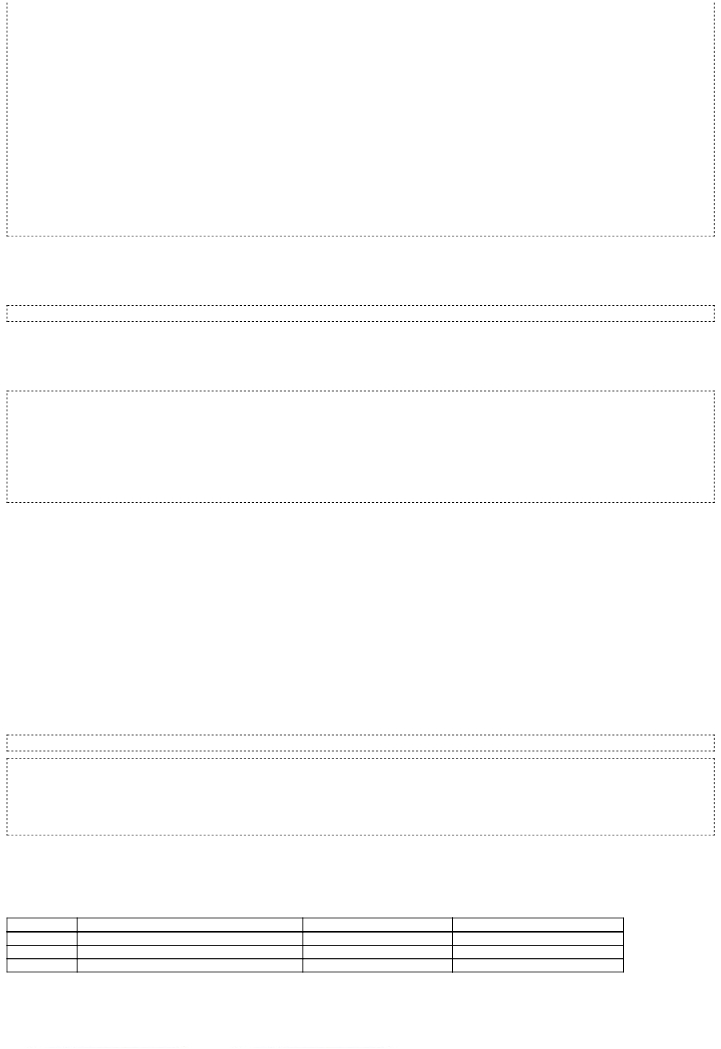
Register search history

/mbl/com/oas/selectKoreaGovPortalSearchResultList.mdo insertOpenApiInquiryHistory

“OpenApiDAO.insertOpenApiInquiryHistory”

Click the list (administration word) /mbl/com/oas/goAdministrationWordDetail.mdo

goAdministrationWordDetail



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 Search Detail Inquiry screen.(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



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 Class file.

2. Import the Jar file, a bundle of the created JAVA Class files, to the project

\* Generate Client Stub Class file

Use the WSDL of the web service, which is open on the web, to create Client Stub.

Refer to WSDL definition 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, use jant 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">

<classpath refid="jeus.libraries.classpath" />

</taskdef>

<target name="-pre-compile">

<mkdir dir="${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">

<classpath refid="jeus.libraries.classpath" />

<classpath refid="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 WSDL definition 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.jar file 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 JEUS installation 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\_ID column 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"

<property name="table"

<property name="tableName"

value="10"/>

value="COMTECOPSEQ"/>

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

C OM086 local code

C OM086 local code

C OM086 local code

C OM086 local code

101

108

112

119

131

Chuncheon

Seoul

Incheon

Suwon

Cheongju



C OM086 local code

C OM086 local code

C OM086 local code

C OM086 local code

C OM086 local code

C OM086 local code

C OM086 local code

C OM086 local code

C OM086 local code

C OM086 local code

C OM086 local code

C OM086 local code

131

133

143

146

152

155

156

159

165

168

184

189

Chongju

Daejeon

Daegu

Jeonju

Ulsan

Changwon

Gwangju

Busan

Mokpo

Yeosu

Jeju

Seoguipo

Screen and execution manual

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

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-API provider and OPEN-API service.

To change the search range per page, modify pageUnit and pageSize in context-properties.xml file.

(This setting have an influence on the entire common service functions.)



|  |  |  |  |
| --- | --- | --- | --- |
| Action | URL | Controller method | QueryID |
| Inquiry | /mbl/com/oas/selectOpenApiInquiryHistoryList.mdo | selectOpenApiInquiryHistoryList | “OpenApiDAO.selectOpenApiInquiryHistoryList” |
|  | /mbl/com/oas/selectOpenApiInquiryHistoryList.mdo | selectOpenApiInquiryHistoryListC nt | “OpenApiDAO.selectOpenApiInquiryHistoryListC nt” |
| Click list | /mbl/com/oas/selectOpenApiInquiryHistory.mdo | selectOpenApiInquiryHistory | “OpenApiDAO.selectOpenApiInquiryHistory” |

10

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

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 history Detail 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

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



|  |  |  |  |
| --- | --- | --- | --- |
| Action | URL | Controller method | QueryID |
| List | /mbl/com/oas/selectOpenApiInquiryHistoryList.mdo | selectOpenApiInquiryHistoryList | “OpenApiDAO.selectOpenApiInquiryHistoryList” |

On

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 System and distribution package plan

Related database structure and schema: related database structure and schema

e govfram e work /m com /ope n-apiconnection service.tx t · Last m odifie d: 2011/10/14 16:01 by m com

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported

[http://creativecommons.org/licenses/by-nc-sa/3.0/]

