

Table of Contents

- 시스템/서비스 연계 - OPEN-API연계서비스
 - 개요
 - 전제조건
 - 설명
 - 설정 및 주의사항
 - 환경설정
 - 관련기능
 - 모바일 대한민국정부포털 검색
 - 모바일 대한민국정부포털 검색 상세조회
 - 모바일 기상청 날씨 조회
 - OPEN-API조회이력 목록조회
 - OPEN-API조회이력 상세조회
 - 참고자료

개요

OPEN-API연계서비스는 다양한 Open-API 서비스를 연계하는 모바일 UX 기반의 컴포넌트이다.
OPEN-API는 “플랫폼으로서의 웹”이라는 특징을 기술적으로 구현한 대표적인 Web 2.0의 기술로써 하나의 웹 사이트에서 자신이 가진 기능을 이용할 수 있도록 공개한 프로그래밍 인터페이스를 말한다.

본 컴포넌트의 예제로 RESTful 방식의 대한민국정부포털 검색서비스와 SOAP 방식의 날씨 조회서비스, 관리자의 OPEN-API 조회 이력 조회 기능을 제공하고 있다.
예제로 제공되는 서비스 외에 다른 **Open-API도 연계가 가능하다.**

- 대한민국정부포털 검색 API : 대한민국정부포털의 통합검색 서비스에서 제공하고 있는 검색결과를 Open-API를 통해 제공
- 기상청 날씨 조회 API : 기상청에서 제공하고 있는 다양한 날씨 관련 정보를 Open-API를 통해 제공

본 컴포넌트에서 연계하고 있는 외부 기관에서 제공하는 OPEN-API의 기능에 대한 질의는 해당 기관에 문의한다.

* 기능 흐름



본 컴포넌트의 모바일 화면은 모바일 디바이스에 탑재된 브라우저를 대상으로 개발 및 테스트 검증이 완료되었다.
또한 동일 제조사의 동일한 브랜드라도 PC용 브라우저와 mobile용 브라우저에 대한 지원이 상이함으로 개발시 유의가 필요함을 밝혀둔다.

검증 대상 디바이스와 브라우저로는 각각 갤럭시S2, 갤럭시S, iPad2, iPhone4, 갤럭시탭1 등의 디바이스와 android용 웹 브라우저, Firefox, Safari, opera mobile 등의 모바일 브라우저를 사용하였다.

검증결과는 아래와 같다.

Android
android용 웹 브라우저, Firefox(6.xx), opera mobile은 android OS 2.3(gingerbread) 에서 HTML5, CSS3 및 컴포넌트의 실행이 안정적으로 실행되는 것을 확인 하였다.
반면 android OS 2.3(gingerbread) 하위 브라우저인 android 2.1, 2.2에서는 컴포넌트 실행에 있어 매끄럽지 못한 HTML5, CSS3지원으로 인하여 컴포넌트의 UI 구현이 원활하지 못한 모습을 보였다.

opera mobile의 경우 UI처리시 색상 및 이벤트 처리에 문제를 야기하였으며
그로인해 opera mobile은 3종 지원브라우저 대상에서 제외 되었다.
테스트 모바일 디바이스로는 갤럭시, 갤럭시S2, 갤럭시탭1을 채택하였다.

iOS
iOS는 3종 지원브라우저 대상중 하나로 Safari 4.2, 4.3 버전에서 검증을 하였으며 비교적 원활한 HTML5, CSS3를 지원을 하였다.
테스트 모바일 디바이스로는 iPhone4, iPad2를 채택하였다.

전제조건

모바일OPEN-API연계서비스를 사용하기 위해서는 OPEN-API 서비스 제공기관에 별도로 OPEN-API 서비스 신청을 해야한다.

서비스 신청 절차는 다음과 같다.

- ① 신청 및 접수 : 신청자가 OPEN-API 사용신청
- ② 검토/승인 : 접수 된 신청서를 관리자가 검토하여 승인처리
- ③ 키발급(이메일) : 신청자에게 이메일을 통하여 인증키(or 사용 아이디/패스워드) 통보
- ④ 기관사용 : 발급된 인증키(or 사용 아이디/패스워드)를 이용하여 실제 Open API를 호출하여 사용

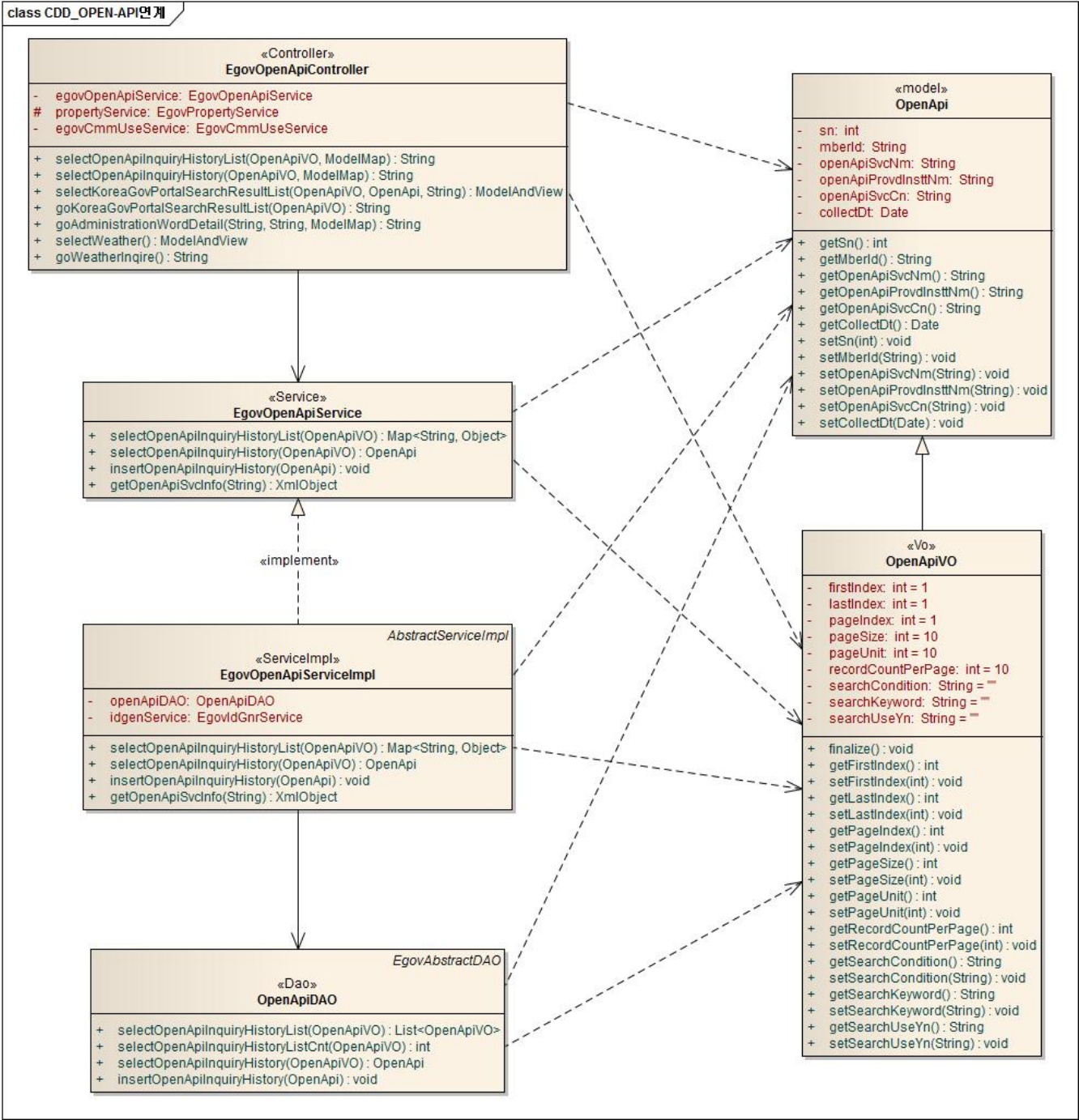
- 대한민국정부포털 검색서비스 신청 URL : <http://www.korea.go.kr/ptl/addservice/openapi/insertOpenApiView.do>
- 기상청 날씨 조회서비스 문의처 : 기상청 정보통신기술과(☎ 02-2181-0439), 담당자 E-Mail(lian1071@korea.kr)

현재 연계된 OPEN-API의 활용은 해당 프로세스를 통해 요청/심의/승인 과정을 통해 발급된 ID/PW를 환경설정 파일에 반영하여 사용 가능하다. 자세한 반영방법은 환경설정 부분을 참고한다.

설명

모바일OPEN-API연계서비스는 크게 OPEN-API조회이력의 관리기능인 **OPEN-API조회이력 목록조회, 상세조회**로 구성되어 있으며, **대한민국정부포털 검색, 기상청 날씨 조회** 기능으로 구성되어 있다.(관련기능 부분참조)

클래스 다이어그램



관련소스

유형	대상소스명	비고
Controller	egovframework.mbl.com.oas.web.EgovOpenApiController.java	OPEN-API연계 Controller Class

Service	egovframework.mbl.com.oas.service.EgovOpenApiService.java	OPEN-API연계 Service Class
ServiceImpl	egovframework.mbl.com.oas.service.impl.EgovOpenApiServiceImpl.java	OPEN-API연계 ServiceImpl Class
Model	egovframework.mbl.com.oas.service.OpenApi.java	OPEN-API연계 Model Class
VO	egovframework.mbl.com.oas.service.OpenApiVO.java	OPEN-API연계 VO Class
DAO	egovframework.mbl.com.oas.service.impl.OpenApiDAO.java	OPEN-API연계 Dao Class
JSP	/WEB-INF/jsp/egovframework/mbl/com/oas/EgovOpenApiInquiryHistoryList.jsp	OPEN-API조회이력 목록조회 페이지
JSP	/WEB-INF/jsp/egovframework/mbl/com/oas/EgovOpenApiInquiryHistoryDetail.jsp	OPEN-API조회이력 상세조회 페이지
JSP	/WEB-INF/jsp/egovframework/mbl/com/oas/EgovMobileKoreaGovPortalSearchResultList.jsp	모바일 대한민국정부포털 검색 페이지
JSP	/WEB-INF/jsp/egovframework/mbl/com/oas/EgovMobileKoreaGovPortalSearchResultDetail.jsp	모바일 대한민국정부포털 검색 상세조회 페이지
JSP	/WEB-INF/jsp/egovframework/mbl/com/oas/EgovMobileWeatherInquire.jsp	모바일 기상청 날씨 조회 페이지
QUERY XML	resources/egovframework/sqlmap/mbl/com/oas/EgovOpenApi_SQL_XXX.xml	OPEN-API연계 QUERY XML
Message properties	resources/egovframework/message/message-common_ko_KR.properties	OPEN-API연계 Message properties
Idgen XML	resources/egovframework/spring/com/context-idgen.xml	OPEN-API연계이력 Id생성 Idgen XML

ID Generation

- ID Generation Service를 활용하기 위해서 Sequence 저장테이블인 COMTECOPSEQ에 OAS_ID (OPEN-API조회이력 아이디) 항목을 추가해야 한다.

관련테이블

테이블명	테이블명(영문)	비고
OPEN-API조회이력	COMTNOPENAPI	OPEN-API조회 이력 관리
공통상세코드	COMTCCMMNDETAILCODE	OPEN-API서비스구분코드

로그인/회원관리/권한관리

OPEN-API 연계 서비스는 **대한민국정부포털 검색, 기상청 날씨 조회, OPEN-API 연계정보 목록조회, 상세조회**에서 사용자/관리자 인증이 필요하다. 인증을 위해 로그인/회원관리/권한관리 컴포넌트와 연계하여 사용할 수 있다. Spring Security를 통해 인증을 할 경우 다음과 같이 변경하면 된다.

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

```

<!-- 인증된 유저의 LoginVO, 권한, 인증 여부를 확인 할 수있는 서비스 클래스-->
<bean id="egovUserDetailsHelper" class="egovframework.com.cmm.util.EgovUserDetailsHelper">
  <property name="egovUserDetailsService">
    <ref bean="egovUserDetailsService" />
  </property>
</bean>

<!-- 시큐리티를 이용한 인증을 사용할 빈 -->
<bean id="egovUserDetailsService" class="egovframework.com.sec.ram.service.impl.EgovUserDetailsServiceImpl" />

```

web.xml

```

<!-- 2. 스프링 시큐리티 설정-->
<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 끝 -->

<!-- 4. Spring Security 로그아웃 설정을 위한 필터 설정 -->
<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>

<!-- 4. Spring Security 로그인 설정을 위한 필터 설정 -->
<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>

```

web.xml에 위 내용과 같은 Spring Security 관련 필터를 추가 하여야 한다.

로그인/회원관리/권한관리에 대한 자세한 내용은 다음을 참조한다.

- 로그인 : [일반로그인](#)
- 회원관리 : [사용자관리](#)
- 권한관리 : [권한관리](#)

사용자/관리자 권한이 필요한 페이지는 다음과 같다.

권한 종류	페이지 경로	설명
사용자	/mbl/com/oas/goKoreaGovPortalSearchResultList.mdo	대한민국정부포털 검색 화면
사용자	/mbl/com/oas/goAdministrationWordDetail.mdo	대한민국정부포털 행정용어 상세화면
사용자	/mbl/com/oas/goWeatherInquire.mdo	기상청 날씨 조회 화면
관리자	/mbl/com/oas/selectOpenApiInquiryHistoryList.mdo	OPEN-API 연계 정보 목록조회 화면
관리자	/mbl/com/oas/selectOpenApiInquiryHistory.mdo	OPEN-API 연계 정보 상세조회 화면

설정 및 주의사항

- * 현재 배포된 모바일 공통컴포넌트는 HTML5, CSS3를 적용하고 있다.
- * 그에 따라, 현재 모바일OK, W3C의 모바일 웹 사이트 유효성 검사의 표준에 맞지 않으며, 정확한 유효성 검사에 대한 문의는 해당 전문기관에 문의가 필요하다.
- * 현재 배포된 모바일 공통컴포넌트 V 2.0은 소스코드에 대한 보안성 검증이 완료 되었다.

환경설정

모바일OPEN-API연계서비스 기능을 활용하기 위하여 필요한 항목 및 그 환경 설정은 다음과 같다.

- 프로퍼티 파일 설정(egovframework/egovProps/conf/oas.properties)

```
#-----
#
# oas.properties : OPEN-API 연계 관련 인증키 정보를 나타내는 Config 파일
#
#-----
#
# 1. key = value 구조
# 2. key 값은 공백문자를 포함불가, value 값은 공백문자를 가함
# 3. key 값으로 한글을 사용불가, value 값은 한글사용이 가능
# 4. 줄을 바꿈 필요가 있으면 '\n'를 라인의 끝에 추가 (만약 '\n'문자를 사용해야 하는 경우는 '\\n'를 사용)
# 5. Windows에서의 디렉토리 표시 : '\\\n' 또는 '/' ('\\n' 사용하면 안됨)
# 6. Unix에서의 디렉토리 표시 : '/'
# 7. 주석문 처리는 # 사용
# 8. value 값 뒤에 스페이스가 존재하는 경우 서버릿에서 참조할때는 에러발생할 수 있으므로 trim()하거나 마지막 공백없이 properties 값을 설정할것
#
# 대한민국정부포털 검색서비스 인증키
koreaGovPortalSearchKey = 발급받은인증키
# 기상청 날씨조회서비스 인증 ID
kmaWeatherInquiryId = 발급받은인증ID
# 기상청 날씨조회서비스 인증 패스워드
kmaWeatherInquiryPassword = 발급받은인증패스워드
```

위의 정보에서 **koreaGovPortalSearchKey**, **kmaWeatherInquiryId**, **kmaWeatherInquiryPassword** 항목에 각각 발급받은 인증정보를 입력한다.

- 한글 인코딩 설정(server.xml)

JSP페이지에서 한글이 깨지는 것을 방지하기 위해 server.xml 파일 내의 Connector 태그의 제일 마지막에 **URIEncoding="UTF-8"** 를 추가한다.

```
<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" />
```

관련기능

모바일OPEN-API연계서비스는 크게 일반사용자가 사용하는 **모바일 대한민국정부포털 검색**, **모바일 대한민국정부포털 검색 상세조회**, **모바일 기상청 날씨 조회** 기능 및 관리자가 사용하는 **OPEN-API연계이력 목록조회**, **OPEN-API연계이력 상세조회** 기능으로 구성되어있다.

모바일 대한민국정부포털 검색

사전 준비

모바일 대한민국정부포털 검색 서비스를 이용하기 위해서는 아래와 같은 준비가 필요하다.

1. OPEN-API 서비스 요청 결과 XML 확보
2. XML을 바탕으로 XML Schema(XSD) 작성
3. 작성된 XML Schema를 바탕으로 OPEN-API 서비스 요청 결과를 매핑할 JAVA Class파일 생성
4. 생성된 JAVA Class파일 묶음인 Jar파일을 Project에 Import

* OPEN-API 서비스 요청 결과 XML

```
<?xml version="1.0" encoding="UTF-8"?>
<search>
  <conditions>
    <query>전자정부</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>전자정부</b> 인프라 통합서비스]]></title>
        <contents><![CDATA[<b>전자정부</b> 관련 업무 수행의 편의성을 제공하고자 <b>전자정부</b> 인프라 시스템에 대한 통합서비스를 제공
합니다.]]></contents>
        <organ><![CDATA[ 행정안전부]]></organ>
        <linkurl><![CDATA[http://www.infra.go.kr]]></linkurl>
      </resource>
      <resource>
        <title><![CDATA[<b>전자정부</b> 표준관리시스템]]></title>
        <contents><![CDATA[<b>전자정부</b> 추진관련 문서화된 기준, 규격, 지침 및 기술규정에 대한 규약의 집합 시스템입니
다.]]></contents>
        <organ><![CDATA[ 행정안전부]]></organ>
        <linkurl><![CDATA[http://www.guide.go.kr]]></linkurl>
      </resource>
      <resource>
        <title><![CDATA[ 외국인을 위한 <b>전자정부</b>]]></title>
        <contents><![CDATA[<b>전자</b> 민원, 방문예약, 정보마당, 고용지원, 투자촉진 안내 서비스입니다.]]></contents>
        <organ><![CDATA[ 법무부]]></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="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>

* OPEN-API 서비스 요청 결과 매핑 JAVA Class파일 생성

OPEN-API 서비스 요청 결과를 매핑할 JAVA Class파일 생성하기 위해 Apache에서 제공하는 **XMLBean**(<http://xmlbeans.apache.org>)을 이용한다. XMLBean을 설치한 후, 콘솔상에서 아래의 명령을 실행한다.

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

위의 명령을 실행하면 **KoreaGovPortalSearch.jar** 파일이 생성되며, 이 파일을 프로젝트에 추가한다. 현재 배포파일에는 **KoreaGovPortalSearch.jar** 파일이 포함되어 있으므로 위의 절차가 불필요하다.

* 소스에 적용하기

```
// xsd 파일을 통해 만들어진 서비스 특화된 객체에 값을 매핑
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. XmlObject를 상속하는 **XXXXDocument** 객체를 생성한다. (XXXX는 XML의 루트명. 대한민국정부포털 검색서비스의 결과 XML 루트는 Search이다.) **requestUrl**은 제공하는 OPEN-API의 서비스의 URL이다.
2. 생성한 객체에서 **getXXXX()** 메소드를 이용하여 루트를 얻는다. (마찬가지로 XXXX는 XML의 루트명)
3. 루트객체에서 **getXXXX()** 메소드를 이용하여 접근하고자 하는 엘리먼트를 얻는다. (XXXX는 접근하고자 하는 엘리먼트명)
4. **noNamespace.XXXXXDocument.XXXX** 객체를 생성하여 얻고자하는 데이터를 **getXXXXArray()** 메소드를 통해 배열로 저장한다. (XXXX는 반복되는 데이터의 상위 엘리먼트명. 대한민국정부포털 검색 OPEN-API는 resource의 하위로 title, contents, linkurl 등이 존재한다.)

* 기타사항

예제로 제공되는 기능 외의 추가 기능 및 옵션을 사용하고자 한다면 '**대한민국정부포털 Open API 사용자 매뉴얼**'을 참조한다. (대한민국정부포털 Open API사용자 매뉴얼 : <http://www.korea.go.kr/ptl/addservice/openapi/selectOpenApiInfo.do>)

※ 예제로 제공하는 대한민국정부포털 검색 OPEN-API 외의 다른 OPEN-API 사용하고자할 때 위와 같은 절차를 거치면 연계하여 사용이 가능하다.

비즈니스 규칙

검색조건에 따라 대한민국정부포털에서 제공하는 정보를 검색할 수 있다. 검색조건이 행정용어인 경우 상세조회가 가능하며, 나머지의 경우 해당 URL로 이동한다.

검색을 수행할 때마다 검색 정보가 OPEN-API조회이력 테이블에 자동으로 등록된다.

등록처리 시 OAS_ID 컬럼은 "egovframework.rte.fdl.idgnr.impl.EgovTableIdGnrService"를 통하여

Primary Key => OAS_ID : 일련번호로 자동생성 부여된다.

```
<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>
```

관련코드

N/A

관련화면 및 수행매뉴얼

Action	URL	Controller method	QueryID
검색	/mbl/com/oas/selectKoreaGovPortalSearchResultList.mdo	selectKoreaGovPortalSearchResultList	
검색이력 등록	/mbl/com/oas/selectKoreaGovPortalSearchResultList.mdo	insertOpenApiInquiryHistory	"OpenApiDAO.insertOpenApiInquiryHistory"
목록클릭(행정용어)	/mbl/com/oas/goAdministrationWordDetail.mdo	goAdministrationWordDetail	



검색: 대한민국정부포털에서 제공하는 정보를 검색하기 위해서 상단의 검색조건을 선택 후 해당하는 검색문자를 입력 후 검색 버튼을 클릭한다.
 목록클릭: 모바일 대한민국정부포털 검색 상세조회 화면으로 이동한다.(검색조건이 행정용어일 경우)

모바일 대한민국정부포털 검색 상세조회

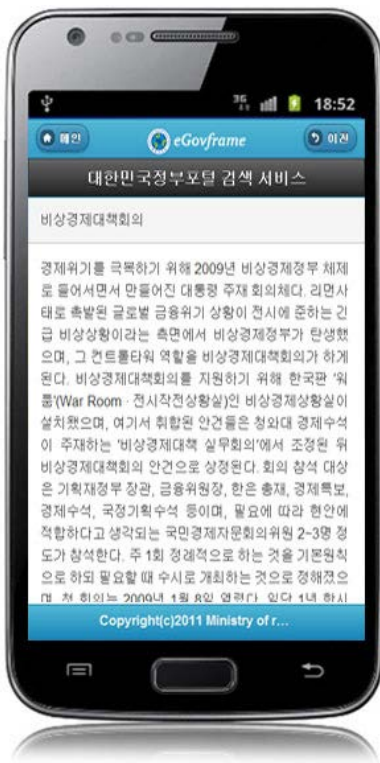
비즈니스 규칙

모바일 대한민국정부포털 검색 화면에서 행정용어 검색 후 목록 클릭 시 이동되는 화면으로 행정용어 검색 정보에 대한 상세정보를 보여준다.

관련코드

N/A

관련화면 및 수행매뉴얼



모바일 기상청 날씨 조회

사전 준비

모바일 기상청 날씨 조회 서비스를 이용하기 위해서는 아래와 같은 준비가 필요하다.
※ 아래의 프로세스는 기상청에서 제공하는 가이드에 준하여 작성하였다.

1. 웹상에 공개된 웹서비스의 wsdl을 참조하여 Client Stub Class 파일 생성
2. 생성된 JAVA Class파일 묶음인 Jar파일을 Project에 Import

* Client Stub Class 파일 생성

웹상에 공개된 웹서비스의 WSDL을 참조하여 **Client Stub**을 생성한다.
기상청에서 제공하는 WSDL 정의페이지(<http://websky.kma.go.kr>)를 참조한다.
본 컴포넌트 예제에서 사용하는 서비스는 **지상 웹서비스** 이다.

Client Stub을 생성하기 위해서는 JEUS(<http://technet.tmax.co.kr>)에서 제공하는 **jant**를 이용한다.

* build.xml 생성

```
<?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>
```

위의 코드에서 wsdl="http://websky.kma.go.kr/services/SurfaceService?wsdl" 부분을 제공하고자 하는 서비스에 맞게 수정한다.
(기상청에서 제공하는 서비스는 WSDL 정의페이지(<http://websky.kma.go.kr>)에서 해당 주소값을 얻을 수 있다.)

build.xml 생성 후 [JEUS 설치 폴더\Wsamples\Wcommon]에 **build.xml**을 위치시키고, **build.xml**이 위치한 경로에서 **jant**를 실행한다.

```
C:\build.xml 위치경로>jant
```

jant를 실행하고나면 build.xml이 위치한 폴더에 build 폴더와 dist 폴더가 생성되며,
dist 폴더에 **SurfaceService.jar** 파일이 생성된다.

Client Stub Class(SurfaceService.jar)를 사용하기 위해서는 다음과 같은 라이브러리가 필요하다.

```
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
saa-api.jar
saa-impl.jar
```

위의 라이브러리 중 아래의 파일은 함께 배포되지 않기 때문에 별도로 프로젝트에 추가하여야 한다.
(해당 라이브러리 파일은 JEUS 설치폴더\Wlib\Wsystem 내에 존재)

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

비즈니스 규칙

기상청에서 제공하는 날씨정보를 조회할 수 있다.
조회를 수행할 때마다 조회 정보가 OPEN-API조회이력 테이블에 자동으로 등록되며,
등록처리 시 OAS_ID 컬럼은 "egovframework.rte.fdl.idgnr.impl.EgovTableIdGnrService"를 통하여

```
Primary Key => OAS_ID : 일련번호로 자동생성 부여된다.
```

```
<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>
```


관련코드

모바일 기상청 날씨 조회에서 사용되는 코드 및 그에 따른 설정 값의 반영사항은 다음과 같다.

코드분류	코드분류명	코드ID	코드명
COM086	지역코드	101	춘천
COM086	지역코드	108	서울
COM086	지역코드	112	인천
COM086	지역코드	119	수원
COM086	지역코드	131	청주
COM086	지역코드	133	대전
COM086	지역코드	143	대구
COM086	지역코드	146	전주
COM086	지역코드	152	울산
COM086	지역코드	155	창원
COM086	지역코드	156	광주
COM086	지역코드	159	부산
COM086	지역코드	165	목포
COM086	지역코드	168	여수
COM086	지역코드	184	제주
COM086	지역코드	189	서귀포

관련화면 및 수행매뉴얼

Action	URL	Controller method	QueryID
날씨조회	/mbil/com/oas/selectWeather.mdo	selectWeather	
검색이력 등록	/mbil/com/oas/selectKoreaGovPortalSearchResultList.mdo	insertOpenApiInquiryHistory	"OpenApiDAO.insertOpenApiInquiryHistory"



OPEN-API조회이력 목록조회

비즈니스 규칙

조회조건으로 목록조회를 할 수 있다.

관련코드

코드분류	코드분류명	코드ID	코드명
COM081	OPEN-API 제공기관	APIS001	날씨
COM081	OPEN-API 제공기관	APIS002	검색
COM082	OPEN-API 서비스구분	APIO001	기상청
COM082	OPEN-API 서비스구분	APIO002	대한민국정부포털

관련화면 및 수행매뉴얼

Action	URL	Controller method	QueryID
조회	/mbl/com/oas/selectOpenApiInquiryHistoryList.mdo	selectOpenApiInquiryHistoryList	"OpenApiDAO.selectOpenApiInquiryHistoryList"
	/mbl/com/oas/selectOpenApiInquiryHistoryList.mdo	selectOpenApiInquiryHistoryListCnt	"OpenApiDAO.selectOpenApiInquiryHistoryListCnt"
목록클릭	/mbl/com/oas/selectOpenApiInquiryHistory.mdo	selectOpenApiInquiryHistory	"OpenApiDAO.selectOpenApiInquiryHistory"

OPEN-API조회이력 목록은 페이지 당 10건씩 조회되며 페이지는 10페이지씩 이루어진다. 검색조건은 OPEN-API 제공기관, OPEN-API 서비스 구분에 대해서 수행된다.

페이지 당 검색 범위를 변경하고자 하는 경우 context-properties.xml 파일의 pageUnit, pageSize를 변경한다.

(단 해당 설정은 전체 공통서비스 기능에 영향을 미친다.)

The screenshot shows the eGovFrame Portal interface. At the top, there's a navigation bar with '메인' (Main) and '목록' (List) buttons. Below it, a search bar is visible with the text 'Open-API 조회 이력 목록조회' and a dropdown menu. The main content area displays a table with the following data:

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

At the bottom of the table, there are pagination controls showing '1' as the current page, with links for previous and next pages. The footer contains the text 'Copyright(c)2011 Ministry of republic administration and security.'

조회: OPEN-API조회이력을 조회하기 위해서는 상단의 검색조건을 선택 후 해당하는 검색문자를 입력 후 조회 버튼을 클릭한다.
 목록클릭: OPEN-API조회이력 상세조회 화면으로 이동한다.

OPEN-API조회이력 상세조회

비즈니스 규칙

OPEN-API조회이력 목록조회에서 목록 클릭 시 이동되는 화면으로 OPEN-API조회이력에 대한 상세정보를 보여준다.

관련코드

N/A

관련화면 및 수행매뉴얼

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

OPEN-API조회이력 상세조회화면은 OPEN-API조회이력 목록조회를 할 수 있다.



Open-API 조회 이력 - 상세 조회

Open-API 제공기관 기상청

Open-API 서비스구분 날씨

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 : 28.8°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 : 28°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

목록

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

목록: OPEN-API조회이력 목록조회 화면으로 이동한다.

참고자료

- 실행환경 참조 : [ID Generation Service](#)
- 배포 및 테스트 관련 : [모바일 공통컴포넌트란 및 배포 패키지 구성안](#)
- 관련 데이터베이스 구조 및 스키마 : [관련 데이터베이스 구조 및 스키마](#)