

Anyframe Simpleweb Plugin



Version 1.0.1

저작권 © 2007-2011 삼성SDS

본 문서의 저작권은 삼성SDS에 있으며 Anyframe 오픈소스 커뮤니티 활동의 목적하에서 자유로운 이용이 가능합니다. 본 문서를 복제, 배포할 경우에는 저작권자를 명시하여 주시기 바라며 본 문서를 변경하실 경우에는 원문과 변경된 내용을 표시하여 주시기 바랍니다. 원문과 변경된 문서에 대한 상업적 용도의 활용은 허용되지 않습니다. 본 문서에 오류가 있다고 판단될 경우 이슈로 등록해 주시면 적절한 조치를 취하도록 하겠습니다.

I. Introduction	1
II. Simplification of web	2
1. Configuration	3
1.1. Tiles View 설정	3
1.2. Tiles Configurer 설정	3
1.3. JSON View 설정	4
1.4. TilesInterceptor 설정	4
2. Controller	6
2.1. AbstractServiceController	6
2.2. SimpleMapServiceController	7
2.3. SimpleJSONController	7
2.4. SimpleJSONTreeController	8
2.5. FileUploadController	9
3. Tag Library	11
3.1. Spring JS 연계	11
3.2. submit tag	12
3.3. link tag	13
3.4. setProperty tag	13
3.5. doublesubmit tag	14
3.6. init tag	15
3.7. validator tag	15
3.8. model tag	16
3.9. tiles tag	16
3.10. validate tag	16

I.Introduction

Simpleweb Plugin은 Spring MVC 기반의 웹 어플리케이션을 개발할 때 개발자가 웹 개발을 보다 쉽고 간단하게 할 수 있게 하는데 필요한 공통 설정 파일 및 라이브러리들(공통 Controller 클래스 및 태그 라이브러리 포함)을 제공한다. 실제 사용 예를 보여주는 샘플 코드는 Simpleweb VO/Map/jQuery Plugin 등에서 제공되므로 참고하도록 한다.

Installation

Command 창에서 다음과 같이 명령어를 입력하여 simpleweb-plugin을 설치한다.

```
mvn anyframe:install -Dname=simpleweb
```

installed(mvn anyframe:installed) 혹은 jetty:run(mvn clean jetty:run) command를 이용하여 설치 결과를 확인해볼 수 있다.

Dependent Plugins

Plugin Name	Version Range
Idgen [http://dev.anyframejava.org/docs/anyframe/plugin/optional/idgen/1.0.3/reference/htmlsingle/idgen.html]	2.0.0 > *
Tiles [http://dev.anyframejava.org/docs/anyframe/plugin/optional/tiles/1.0.1/reference/htmlsingle/tiles.html]	2.0.0 > *

II.Simplification of web

Spring MVC 기반의 웹 어플리케이션을 개발할 때 개발자는 요청을 처리할 Controller 클래스, 각 요청과 Controller 클래스를 매핑해줄 *-servlet.xml을 작성해야 한다. 이러한 과정은 매 요청을 처리할 때마다 발생하게 되며 때로는 많은 시간을 필요로 하게 된다. 이에 Anyframe에서는 Controller 클래스 작성 없이도 요청을 처리해줄 공통 Controller 와 Tag Library를 제공한다. 또한, 보다 간결하고 역동적인 웹 개발을 위해 Spring JS, jQuery와의 연계 방안을 제시하고 있다. Anyframe에서 웹 개발을 간소화하기 위해 제공하는 기능 목록은 아래와 같다.

- Controller 클래스 개발 없이도 요청 처리가 가능한 공통 Controller 제공
 - 비즈니스 서비스 호출
 - Command 객체 바인딩(VO, Map 타입 지원)
 - 중복 폼 서브밋 방지
 - Custom Validator 클래스 적용
 - Model Validation 기능 사용을 위한 JSR-303 Validator 적용
 - 화면 초기 데이터 셋팅
 - Tiles Attribute 셋팅
 - View 이름 셋팅
 - 공통 Controller 클래스에서 사용될 데이터를 JSP 페이지에서 설정할 수 있도록 Anyframe Tag Library 제공
 - VO 객체를 사용하는 Application 개발을 위한 VO용 공통 Controller 제공
 - Map 객체를 사용하는 Application 개발을 위한 Map용 공통 Controller 제공
 - jQuery와의 연계를 위한 공통 Controller 제공
 - Tiles 화면 레이아웃 사용 시 Partial Rendering 및 Popup 기능 제공
 - 공통 Controller의 넘겨줄 데이터를 셋팅할 수 있는 Anyframe Tag Library 제공
 - 간편하게 Client단 Validation을 구현할 수 있는 Anyframe Tag Library 제공
-

1. Configuration

위에 나열된 Anyframe에서 제공하는 기능들을 이용한다면, Spring MVC 기반의 웹 어플리케이션 개발을 간소화시켜줄 수 있다. 다만, Partial Rendering과 Popup 등의 Spring JS(Javascript) 기반으로 동작하는 기능을 사용하기 위해서는 해당 웹 어플리케이션의 화면 레이아웃이 Tiles로 구성되어야 한다.

1.1. Tiles View 설정

Spring MVC에서는 Tiles 기반의 Spring JS(Ajax 지원)를 사용하기 위해 AjaxUrlBasedViewResolver와 AjaxTilesView 클래스를 제공한다. Anyframe의 보다 많은 기능을 제공하기 위해 각각의 클래스를 확장한 AjaxUrlBasedContentTypeViewResolver와 AjaxParamTilesView를 제공하며 아래와 같이 설정해 주도록 한다. 다음은 simpleweb-plugin 설치로 추가된 Spring 속성 정의 파일 simpleweb-servlet.xml 파일의 일부이다.

```
<bean id="ajaxViewResolver"
  class="org.anyframe.simpleweb.js.ajax.AjaxUrlBasedContentTypeViewResolver">
  <property name="viewClass"
    value="org.anyframe.simpleweb.js.ajax.tiles2.AjaxParamTilesView"/>
  <property name="order" value="1"/>
  <property name="contentType" value="text/html; charset=utf-8"/>
</bean>
```

또한, 이러한 Tiles를 사용할 경우 Tiles Definition 정보가 작성되어 있는 xml 파일의 위치를 TilesConfigurer를 통해 정의해야 한다. 이는 Anyframe Tiles Plugin 설치시 자동으로 등록되며, 자세한 사항은 Tiles Plugin >> Tiles Configure [http://dev.anyframejava.org/docs/anyframe/plugin/optional/tiles/1.0.1/reference/htmlsingle/tiles.html#tiles_tilesintegration_tilesconfigurer]를 참조하도록 한다.

1.2. Tiles Configurer 설정

Tiles를 사용할 경우 Tiles Definition 정보가 작성되어 있는 xml 파일의 위치를 TilesConfigurer를 통해 정의하도록 한다. 이 때, Anyframe에서는 Spring 3.0의 EL문을 사용할 수 있도록 tilesInitializer를 통해 ELTilesInitializer 아래와 같이 정의해 주도록 한다. 다음은 simpleweb-plugin 설치로 추가된 Spring 속성 정의 파일 moviefinder-servlet.xml 파일의 일부이다.

```
<bean id="tilesConfigurer"
  class="org.springframework.web.servlet.view.tiles2.TilesConfigurer">
  <property name="tilesInitializer">
    <bean class="org.anyframe.tiles.ELTilesInitializer">
      <property name="definitions">
        <list>
          <value>/WEB-INF/tilesviews.xml</value>
        </list>
      </property>
    </bean>
  </property>
</bean>
```



Servlet 2.4 spec의 Tiles EL 지원

위에서 설명한 Tiles에서 제공하는 EL은 기본적으로 Servlet 2.5 spec이상에서만 지원된다. 이에 Anyframe Simpleweb에서는 Servlet 2.4 spec에서도 Tiles EL을 지원해 주기 위해 tilesInitializer 클래스를 제공한다. 설정 방법은 아래와 같다.

```
<bean id="tilesConfigurer"
  class="org.springframework.web.servlet.view.tiles2.TilesConfigurer">
```

```
<property name="tilesInitializer">
  <bean class="org.anyframe.tiles.Servlet24ELTilesInitializer">
    <property name="definitions">
      <list>
        <value>/WEB-INF/tilesviews.xml</value>
      </list>
    </property>
  </bean>
</property>
</bean>
```

1.3.JSON View 설정

jQuery를 사용하여 Application을 개발할 때 기본적으로 JSON 객체 타입의 데이터를 주고 받게 되며 Spring 3.0에서는 이러한 JSON View를 지원하기 위해 MappingJacksonJsonView 클래스를 제공한다. Application에서는 JSON View가 아닌 일반적인 JSP View도 함께 사용하므로 이미 정의되어 있는 AjaxUrlBasedContentTypeViewResolver와 별도로 XmlViewResolver를 정의하고 JSON 요청을 처리할 MappingJacksonJsonView를 정의하도록 한다. 다음은 simpleweb-plugin 설치로 추가된 Spring 속성 정의 파일 simpleweb-vo-servlet.xml의 일부이다. (simpleweb-map-servlet.xml of Simpleweb-Map plugin, simpleweb-json-servlet.xml of Simpleweb-JSON plugin)

```
<bean name="jsonViewResolver" class="org.springframework.web.servlet.view.XmlViewResolver">
  <property name="order" value="1"/>
  <property name="location" value="/WEB-INF/simpleweb-json-views.xml"/>
</bean>
```



다중 View Resolver의 우선 순위

UrlBasedViewResolver를 상속받고 있는 View Resolver들은 View 이름을 찾지 못하면 404 에러를 발생 시키므로 View Resolver 중 우선 순위를 제일 낮게 설정해 줘야 한다. 이에 XmlViewResolver의 우선 순위를 위에서 정의한 AjaxUrlBasedContentTypeViewResolver의 우선 순위보다 높게 정의해줘야 한다.

위에서 정의한 XmlViewResolver에서 View 정보를 정의할 simpleweb-vo-views.xml에 MappingJacksonJsonView를 정의해 준다. 다음은 simpleweb-plugin 설치로 추가된 Spring 속성 정의 파일 simpleweb-vo-views.xml 파일의 일부이다.

```
<bean name="jsonView"
  class="org.springframework.web.servlet.view.json.MappingJacksonJsonView"/>
```

위의 코드에서 "jsonView"라는 view 이름을 MappingJacksonJsonView에서 처리하게 되므로 JSON 요청을 처리하는 Controller 클래스의 리턴되는 view 이름은 "jsonView"가 되도록 한다.

1.4.TilesInterceptor 설정

Tiles Layout 기반의 개발을 하기 위해서는 기본적으로 리턴되는 view 이름을 Tiles Definition으로 정의하고 있다. 이 때, Anyframe submit tag, link tag의 layout 속성을 사용하여 view 이름을 정의해 줄 수 있다. 이 때, 이 속성을 정의하지 않을 시 즉, Tiles Definition을 정의하지 않을 시에 오류가 발생하게 된다. 이에 layout을 정의하지 않아도 기본적으로 "standardLayout"이라는 view를 출력해주기 위해 아래와 같이 Simpleweb을 사용하는 요청에 대해 Interceptor를 정의해 줘야한다. 아래는 Inertceptor를 정의한 simpleweb-servlet.xml의 일부이다.

```
<mvc:interceptors>
  <mvc:interceptor>
    <mvc:mapping path="/simple*" />
    <bean class="org.anyframe.simpleweb.interceptor.TilesInterceptor" />
  </mvc:interceptor>
</mvc:interceptors>
```

```
</mvc:interceptor>  
</mvc:interceptors>
```

2.Controller

Spring MVC 기반의 Web Application을 개발할 때는 크게 JSP파일, Controller 클래스, Spring 속성 정의 파일(*-servlet.xml) 세가지 파일을 작성해야 하며 Tiles를 사용한 화면 갯수에 따라 Tiles Definition을 정의하게 된다. 매번 비슷한 작업을 위해 Controller 클래스와 Spring 속성 정의 파일, Tiles Definition을 작성하게 되면 때때로 많은 시간이 걸리거나 관리되어야 할 파일의 갯수가 많아지게 된다. 이에 Anyframe에서는 공통 Controller 클래스와 Tag Library를 제공함으로써 발자가 매번 Controller 클래스를 작성해야 하는 개발 공수를 줄여주고 Spring 속성 정의 또한 필요하지 않게 만들어 준다. 이번 장에서는 Anyframe에서 제공하는 공통 Controller 클래스에 대해 알아보고 각각의 Controller 클래스를 확장하여 사용할 수 있는 확장 포인트에 대해서도 알아보도록 한다.

2.1.AbstractServiceController

AbstractServiceController는 Spring에서 제공하는 MultiActionController를 상속받고 있으며 실제로 `handleRequest(request, response)`메소드를 오버라이드하여 요청을 처리 하고 있다. 다음은 `handleRequest()` 메소드의 일부이다.

```
public ModelAndView handleRequest(HttpServletRequest request,
    HttpServletResponse response) throws Exception {

    // 1. preventing duplicated submission.
    checkDuplicatedSubmission(request);

    // 2. execute service method and get view name
    Map viewInfo = executeServiceAndReturnView(request);

    String viewName = (String)viewInfo.get("VIEW_NAME");

    if (viewInfo.get("INPUT_PAGE") == null)
        // set tiles attribute names and values
        setTilesAttributes(request, null);

    // 3. set attributes for empty command object
    setCommandAttributes(request);

    // 4. set attributes for initial data
    setInitdataAttributes(request);

    ModelAndView mav = new ModelAndView(viewName);

    return mav;
}
```

일반 Controller 클래스에서 해주는 작업들을 공통화하여 제공하고 있으며 Transfer Object의 종류에 따라 데이터 바인딩 로직을 확장하여 구현하여야 한다. Anyframe에서는 VO, Map, JSON 객체에 대한 데이터 바인딩 로직을 구현한 SimpleServiceController, SimpleMapServiceController, SimpleJSONController를 확장하여 제공한다. AbstractServiceController에서 제공하는 Controller 클래스의 기본 기능은 아래의 같다.

- Form Double Submission 방지 기능
- RequestParameter 값을 이용하여 해당 Tiles attribute의 값 변경 기능
- 화면에서 필요한 초기 데이터 셋팅
- form에서 사용할ModelAttribute 셋팅

- 데이터 바인딩 (VO, Map, JSON 객체 타입에 따른 Controller 별도 제공)
- Date 타입의 데이터에 대해 포매팅 적용
- Server단 Validation (Validator 실행)
- 비즈니스 서비스 호출
- 출력해줄 View 리턴 (메소드의 리턴 타입에 따른 처리 로직 ex> org.anyframe.pagination.Page 타입 일 경우 페이지 객체 셋팅)



BeanMethodRepository 설정

AbstractServiceController는 사용자가 입력한 Service Bean의 id 혹은 name과 메소드 정보를 가지고 해당 서비스의 메소드를 찾아 실행 시켜주는 기능을 제공한다. 이 때, Bean으로 정의된 Service의 정보와 메소드 정보는 BeanMethodRepository에서 찾게 된다. 이에 BeanMethodRepository를 ApplicationContext의 Bean으로 정의 하고 AbstractServiceController를 상속받은 모든 클래스에서는 beanMethodsRepo 속성을 설정 하여 BeanMethodRepository를 인젝션 할 수 있게 해줘야 한다. 다음은 simpleweb-plugin 설치로 추가된 context-simpleweb.xml 파일의 일부이다.

```
<bean id="beanMethodsRepo"
      class="org.anyframe.simpleweb.beans.support.BeanMethodRepository"/>
```

그러므로, AbstractServiceController를 확장한, SimpleServiceController, SimpleMapController, SimpleJSONController등의 Controller 클래스를 사용할 경우 Controller Bean 설정에 beanMethodsRepo와의 Dependency를 정의해줘야 한다.

```
<!-- simple direct service controller -->
<bean name="/simple.do"
      class="org.anyframe.simpleweb.controller.SimpleServiceController">
  <property name="beanMethodsRepo" ref="beanMethodsRepo" />
</bean>
```

2.2.SimpleMapServiceController

SimpleMapServiceController는 Map 타입의 객체를 Transfer Object를 사용할 때의 데이터 바인딩을 지원해줄 AbstractServiceController를 확장한 Controller 클래스이다. HttpServletRequest의 Parameter로 넘어온 값을 메소드 정보의 argument 타입의 객체(여기선 Map객체가 됨)로 바인딩 해주며 Map 타입의 객체를 Transfer Object로 사용하는 경우 데이터 바인딩 방법을 바꾸거나 특정 작업을 추가 할 때, 또는 AbstractServiceController의 기능을 변경하고 싶을 때는 SimpleMapServiceController를 확장하여 구현 하면 된다. 다음은 simpleweb-plugin 설치로 추가된 Spring 속성 정의 파일 simpleweb-map-servlet.xml 파일의 일부이다.

```
<!-- simple direct service controller -->
<bean name="/simplemap.do"
      class="org.anyframe.simpleweb.controller.SimpleMapServiceController">
  <property name="beanMethodsRepo" ref="mapBeanMethodsRepo" />
</bean>
```

위와 같이 정의하면 "/simplemap.do" 요청을 SimpleMapServiceController가 처리하게 된다.

2.3.SimpleJSONController

SimpleJSONController는 JSON 객체를 Datatype으로 사용하는 jQuery의 Component를 지원하기 위해 개발되었다. 현재 SimpleServiceController를 상속받아 구현되어 있으므로 Service Layer에서의 Transfer

Object로는 VO 객체를 사용하도록 한다. 물론, jQuery를 연계하기 위해 SimpleJSONController를 사용할 때 Service Layer의 변경이나 추가 코딩없이 기존의 비즈니스 서비스를 호출하여 사용할 수 있다. 현재 Anyframe simpleweb-plugin에서는 jQuery의 Grid, Autocomplete, UI, Tree, Combobox 컴포넌트 예제를 제공하고 있으며 SimpleJSONController는 주로 Grid(jqgrid plugin), Autocomplete(autocomplete plugin) 기능에 대한 데이터를 셋팅해 주는 역할을 하고 있다. jQuery의 Grid 컴포넌트(jqgrid plugin)에서 Grid를 그릴 때 비즈니스 서비스 호출후 리턴되는 Page 객체를 바로 받을 수 있는 것이 아닌 Grid에서 인식할 수 있는 Key와 Value 쌍의 Map 형태로 Model 객체에 셋팅해줘야 한다. 다음은 SimpleJSONController 클래스의 일부이다.

```
if (methodInfo.getOutputParam()[0].getName().equals(
    ANYFRAME_COMMON_PAGE)) {
    //....
    jsonModel.put("page", ((Page) result).getCurrentPage() + "");
    jsonModel.put("total", ((Page) result).getMaxPage() + "");
    jsonModel.put("records", ((Page) result).getTotalCount());
    jsonModel.put("rows", ((Page) result).getList());
}
```

위의 코드에서 볼 수 있듯이 비즈니스 서비스 수행후 Return 값이 org.anyframe.pagination.Page 타입일 경우 jQuery의 Grid에서 인식 할 수 있는 key값으로 jsonModel 객체를 셋팅해 주고있다. Autocomplete 컴포넌트를 사용할 때도 마찬가지로 출력해줄 데이터를 Autocomplete 컴포넌트에서 인식할 수 있는 key값의 데이터로 셋팅해 주는 부분이 필요하다.

```
jsonModel.put("autoData", resultValue);
```

이렇게 생성된 jsonModel 객체는 Controller의 Model 객체로 셋팅되어 View로 전달되게 된다. 위의 코드는 이미 SimpleJSONController에 구현되어 있으므로 개발자가 jQuery의 Grid, AutoComplete의 기능을 사용할 때에는 추가 코딩없이 사용할 수 있다. 이 외의 특정 Key 값의 데이터를 필요로 하는 또 다른 컴포넌트를 도입하거나 추가 기능이 필요하게 되면 SimpleJSONController를 확장하여 사용하면 된다. 다음은 simpleweb-plugin 설치로 추가된 Spring 속성 정의 파일 simpleweb-json-servlet.xml 파일의 일부이다.

```
<bean name="/simplejson.do" class="org.anyframe.simpleweb.controller.SimpleJSONController">
  <property name="beanMethodsRepo" ref="jsonBeanMethodsRepo" />
</bean>
```

위와 같이 정의하면 "/simplejson.do" 요청을 SimpleJSONController가 처리하게 된다.

2.4.SimpleJSONTreeController

SimpleJSONTreeController는 jQuery의 Tree 컴포넌트(jstree)를 지원하기 위한 공통 Controller이다. SimpleJSONTreeController는 SimpleJSONController를 상속받아 구현되어 있다. 기본적으로 Tree에서 필요한 노드 정보를 셋팅하기 위해 Tree 컴포넌트에서 인식 할 수 있는 형태의 도메인 객체를 사용해야 한다. 이는 Anyframe에서 제공하는 Attributes, JSTreeNode 도메인 클래스를 사용하도록 하고 추가적인 Attribute의 설정이 필요할 경우 해당 클래스를 확장하여 추가하도록 한다. SimpleJSONTreeController는 기본적으로 사용자가 클릭하는 노드의 ID를 가지고 비즈니스 서비스를 호출하고 그 ID에 따라 알맞은 Tree에 출력할 데이터를 셋팅해 주게 된다. 이 때, Tree의 데이터를 셋팅하는 것은 화면 시나리오에 따라 달라질 수 있으므로 개발자는 반드시 SimpleJSONTreeController 클래스를 상속 받고 setTreeData() 메소드를 오버라이드 하여 구현해야한다. 다음은 simpleweb-plugin 설치로 추가된 SimpleJSONTreeController를 확장하여 구현한 MovieTreeController.java 파일의 일부이다.

```
public class MovieTreeController extends SimpleJSONTreeController {

    @Override
    protected void setTreeData(ArrayList<JSTreeNode> listNode, List jsTreeList,
        String id) throws Exception {
        JSTreeNode node;
        Attributes attr;
```



```
}
```

위의 코드에서 볼수 있듯이 "fileData"를 MultipartFile 타입의 객체로 받고 Request Parameter의 "fileDir"를 받아 파일 업로드 로직을 수행한 후 "fileNm", "fileSize", "fileDir"의 데이터를 셋팅한 후 화면에 그 결과를 리턴한다.

3.Tag Library

Anyframe에서는 개발자가 보다 쉽고 간결한 웹 개발을 할 수 있도록 유용한 Tag Library를 제공한다. 이는 주로 Simpleweb 기반의 개발을 지원할 수 있는 Tag와 Spring JS 호출을 위한 Tag로 구성되어 있다. Anyframe에서 제공하는 Tag Library를 사용하기 위해서는 아래와 같이 taglib이 정의되어 있어야 한다.

```
<%@ taglib uri="http://www.anyframejava.org/tags/simpleweb" prefix="simpleweb" %>
```

* 해당 tld 파일은 anyframe-simpleweb 프로젝트 내에 존재한다.

Anyframe Simpleweb에서 제공하는 Tag Library의 주요 기능은 아래와 같다.

- Get 방식의 Link 지원
- POST 방식의 Submit 지원
- SimpleWeb 기반의 공통 Controller 호출
 - 호출할 비즈니스 서비스 정의
 - 비즈니스 서비스 수행 후 리턴할 뷰 정의
- Submit 시 중복 폼 서브밋 방지 기능
- JSP 페이지에서 필요한 Model 객체 정의
- Request Parameter 정의
- Submit이나 Link 사용시 javascript function 호출
- 서버단 Validation 적용을 위한 Validator 정의
- Spring JS를 이용한 Client단 Validation
- Spring JS를 이용한 Tiles 기반의 Partial Rendering
- Spring JS를 이용한 Popup(Dialog) 기능 제공

3.1.Spring JS 연계

위의 기능에서 알 수 있듯이 Partial Rendering, Popup, Validation 기능을 위해 Spring JS를 사용한다. 이번 절에서는 Spring JS와의 연계 방법에 대해 알아보도록 한다. Spring JS를 사용하기 위해서는 spring-js-x.x.jar 라이브러리가 필요하다. 이는 simpleweb-plugin 설치시 자동 설치 되므로 개발자가 신경쓰지 않아도 된다. Spring JS는 dojo기반으로 동작하고 있으며 Spring JavaScript 형태로 dojo를 호출해 준다. 필요한 js, image, css 파일의 접근을 위해 ResourceServlet을 제공하고 있으며 web.xml에 해당 servlet을 정의해줘야 한다.

```
<servlet>  
  <servlet-name>Resource Servlet</servlet-name>  
  <servlet-class>org.springframework.js.resource.ResourceServlet</servlet-class>  
</servlet>  
<servlet-mapping>  
  <servlet-name>Resource Servlet</servlet-name>  
  <url-pattern>/resources/*</url-pattern>  
</servlet-mapping>
```

servlet 정의가 끝나면 해당 js파일과 css파일의 경로를 jsp의 상단에 정의하도록 한다. 다음은 simpleweb-plugin 설치로 추가된 화면 레이아웃이 정의되어 있는 standard.jsp 파일의 일부이다.

```

<script type="text/javascript" src="<c:url value="/resources/dojo/dojo.js" />"></script>
<script type="text/javascript" src="<c:url value="/resources/dojo/io/iframe.js" />"></script>
<script type="text/javascript" src="<c:url value="/resources/anyframe/spring/Anyframe-Spring.js" />"></script>
<script type="text/javascript" src="<c:url value="/resources/anyframe/spring/Anyframe-Spring-Dojo.js" />"></script>
<link type="text/css" rel="stylesheet" href="<c:url value="/resources/dijit/themes/tundra/tundra.css" />" /> />
<!-- 약태의 css는 Anyframe에서 tundra.css를 customizing 한 css 파일이며 필요시 참조하도록 한다. (필수 정의 아님)-->
<link rel="stylesheet" href="<c:url value="/sample/css/tundra-customized.css"/>" type="text/css">
    
```

기본적으로 Spring JS와의 연계를 위해서는 Spring에서 제공하는 JavaScript인 Spring.js와 Spring-Dojo.js를 import하게 된다. 하지만 여기서는 Simpleweb 기능을 지원하기 위해 Anyframe에서 확장한 Anyframe-Spring.js와 Anyframe-Spring-Dojo.js를 사용하도록 한다.

3.2.submit tag

submit tag는 기본적으로 Form submit을 처리하기 위한 tag로 Anyframe의 공통 Controller를 통해 별도의 Controller 클래스 작성없이 바로 비즈니스 서비스를 호출 할 수 있다. submit tag에서 제공하는 Attribute는 아래의 표와 같다.

Attribute	Required	Description
id	true	submit tag를 식별할 유일한 ID
form	true	Submit 대상이 되는 form의 이름
action	false	servlet에서 처리하게될 요청 명. default 값은 "/simple.do"임
service	false	호출해야하는 비즈니스 서비스와 메소드명 (ex> movieService.get(movieId)). 단, service 명은 bean의 id로 명시
forward	false	비즈니스 서비스 호출 후 또다른 요청을 하는 경우. forward 방식으로 전달되기 때문에 Request 유지
redirect	false	비즈니스 서비스 호출 후 또다른 요청을 하는 경우. redirect 방식으로 전달되기 때문에 Request가 유지되지 않음
render	false	요청 처리 후 출력되는 View가 Partial Rendering 될 지 결정. render의 값이 "partial"일 경우 Partial Rendering 됨 단, Tiles 기반의 Layout에서만 가능
layout	false	요청 처리 후 출력되는 View가 사용하게 될 Tile Definition의 이름. default 값은 "standardLayout"임
tiles	false	Tiles Definition의 put-attribute에 들어갈 name, value 쌍을 name:value의 형태로 지정 (ex> tiles="body:list.jsp")
view	false	요청 처리 후 출력할 view 지정
jsValidate	false	<simpleweb:validate> tag를 사용하여 적용된 Spring JS Validation을 Submit 시 체크 할 것인지 설정

submit tag의 정의 예는 아래와 같다.

```

<simpleweb:submit id="addlink" form="movieForm" action="/simplejsoncommon.do"
  layout="jsonLayout" service="simplewebJsonMovieService.create(movie)"
  forward="simplewebJsonMovieFinder.getPagingList(search)"
    
```

```

tiles="body:/WEB-INF/jsp/simpleweb-json/moviefinder/movie/list.jsp"
render="partial" jsvalidate="true">
submit
</simpleweb:submit>
    
```

3.3.link tag

link tag는 Get 방식의 요청을 처리하게 될 tag이다. submit tag를 상속받아 구현되어 있으며 tag attribute 또한 submit tag와 같다. 단, Link Tag에서는 Spring JS를 사용한 Popup 기능을 attribute 설정을 통해 사용할 수 있다. Popup 기능을 사용하기 위해 Link Tag에 추가된 Attribute는 아래의 표와 같다.

Attribute	Required	Description
popup	false	Partial Rendering 대상이 되는 Tile Definition에 해당하는 화면을 Popup으로 제공
popupTitle	false	Popup창의 Title bar에 나오게될 제목

link tag의 정의 예는 아래와 같다.

```

<simpleweb:link action="/simplejsoncommon.do" id="addviewlink"
layout="jsonLayout" render="partial" popup="true">
<simpleweb:setProperty name="title" value="add"/>
</simpleweb:link>
    
```

3.4.setProperty tag

setProperty tag는 submit tag나 link tag의 Inner tag로 사용되며 단순히 name, value 쌍을 가지는 property를 셋팅해주기 위한 tag이다. setProperty tag에서 제공하는 Attribute는 아래와 같다.

Attribute	Required	Description
name	true	property로 정의하게 되는 key 값. Anyframe에서 제공하는 특정 요소를 이름으로 가지며 name으로 셋팅해 줄 수 있는 값과 각각의 기능은 아래 리스트 참고. <ul style="list-style-type: none"> title link나 submit을 실행시킬 대상이 되는 text 또는 이미지 요소 hiddenDiv submit tag, link tag로 인해 생성되는 Parameter들을 가지게 되는 div 요소의 id 값. default는 "hiddenDiv" upload jQuery의 uploadify 플러그인을 이용한 파일 업로드 여부 javascript link나 submit을 실행시키기 전에 수행시킬 javascript 함수 request:\${key}

Attribute	Required	Description
		Controller 클래스에 넘겨줄 Request Parameter
value	false	name에 셋팅해 주게될 실제 값

setProperty의 설정 예는 아래와 같다.

```
<simpleweb:link id="jsonMovie" action="/simplejson.do" layout="jsonLayout" tiles="body:/WEB-INF/jsp/simpleweb-json/moviefinder/movie/list.jsp">
  <simpleweb:setProperty name="title">Simpleweb-JSON[1]</simpleweb:setProperty>
  <simpleweb:setProperty name="javascript" value="fncAlert();"></simpleweb:setProperty>
  <simpleweb:setProperty name="request:pageSize" value="100"></simpleweb:setProperty>
  <simpleweb:setProperty name="request:pageUnit" value="3"></simpleweb:setProperty>
</simpleweb:link>
```

```
<simpleweb:submit id="updateLink" form="movieForm"
  service="simplewebVoMovieService.update(movie)"
  forward="simplewebVoMovieFinder.getPagingList(search)"
  layout="voLayout" tiles="body:/WEB-INF/jsp/simpleweb-vo/moviefinder/movie/list.jsp"
  render="partial" jsValidate="true">
  <simpleweb:setProperty name="title">
    
  </simpleweb:setProperty>
  <simpleweb:setProperty name="hiddenDiv" value="hiddenDivPopup" />
  <simpleweb:setProperty name="upload" value="true" />
</simpleweb:submit>
```

3.5.doublesubmit tag

doublesubmit tag는 link tag나 submit tag의 inner tag로 사용되며 중복 폼 서브밋 방지 기능 구현을 위해 사용한다. doublesubmit tag에서 제공하는 Attribute는 아래와 같다.

Attribute	Required	Description
formName	true	중복 form submit 기능을 적용하게 되는 form의 이름
isShowNewForm	false	중복 form submit 방지 기능을 적용하는 form을 출력하는 요청에 "true"로 설정
isSessionForm	false	form submit에 "true"로 설정

중복 form submit 방지 기능의 구현 예는 아래와 같다.

```
<simpleweb:link id="addviewlink" layout="voLayout" tiles="body:/WEB-INF/jsp/simpleweb-vo/moviefinder/movie/form.jsp">
  <simpleweb:setProperty name="title"></simpleweb:setProperty>
  <simpleweb:model name="movie" type="{package}.simpleweb.vo.moviefinder.domain.Movie"/>
  <simpleweb:init valueList="genreList:simplewebVoGenreService.getList()" />
  <simpleweb:doublesubmit formName="movieForm" isShowNewForm="true"/>
</simpleweb:link>
```

1. 중복 form submit 방지 기능의 대상이 되는 form 출력 요청

formName을 지정해 주고 isShowNewForm 속성을 "true"로 설정한다.

2. form submit 요청

formName을 지정해 주고(위에서 설정한 formName과 동일해야함) isSessionForm 속성을 "true"로 설정한다.

```
<simpleweb:submit id="addlink" form="movieForm"
  service="simplewebVoMovieService.create(movie)"
  forward="simplewebVoMovieFinder.getPagingList(search)"
  layout="voLayout" tiles="body:/WEB-INF/jsp/simpleweb-vo/moviefinder/movie/list.jsp"
  render="partial" jsValidate="true">
  <simpleweb:setProperty name="title"></simpleweb:setProperty>
  <simpleweb:setProperty name="hiddenDiv" value="hiddenDivPopup" />
  <simpleweb:setProperty name="upload" value="true" />
  <simpleweb:doublesubmit formName="movieForm" isSessionForm="true"/>
</simpleweb:submit>
```

3.6.init tag

init tag는 link tag나 submit tag의 inner tag로 사용되며 JSP에서 출력해줄 초기 데이터를 셋팅할 수 있도록 해준다. ", "를 사용하여 여러개의 데이터 셋팅이 가능하다. init tag에서 제공하는 Attribute는 아래와 같다.

Attribute	Required	Description
valueList	true	초기데이터를 셋팅해 줄 값의 리스트. "\${key}: \${service.method};\${key}:\${service.method}..." 형태로 정의

init tag의 정의 예는 아래와 같다.

```
<simpleweb:link id="jsonTree" action="/simplejson.do" layout="jsonLayout" tiles="body:/WEB-INF/jsp/simpleweb-json/moviefinder/movie/tree.jsp">
  <simpleweb:model name="movie" type="${package}.simpleweb.json.moviefinder.domain.Movie" />
  <simpleweb:init valueList="genreList:simplewebJsonGenreService.getDropDownGenreList()"></simpleweb:init>
  <simpleweb:setProperty name="title">Simpleweb-JSON[2]</simpleweb:setProperty>
</simpleweb:link>
```

3.7.validator tag

validator tag는 link tag나 submit tag의 inner tag로 사용되며 Server 단 Validation 또는, Custom Javascript Validation을 지원한다. validator tag에서 제공하는 Attribute는 아래와 같다.

Attribute	Required	Description
validator	false	Validation할 validator를 지정. Server단 Validation일 경우 Validator 클래스의 bean id로 정의 하고 javascript 함수일 경우 "javascript:{function name};"으로 정의
inputpage	false	Server단 Validation일 경우 에러발생시 출력할 View를 지정. Tiles Definition 파일에 정의된 put-attribute의 \${name}: \${value}의 형태로 정의

다음은 validator tag의 정의 예이다.

```
<simpleweb:validator validator="movieValidator" inputpage="body:/WEB-INF/jsp/simpleweb-vo/moviefinder/movie/form.jsp"/>
```

3.8.model tag

model tag는 link tag나 submit tag의 inner tag로 사용되며 JSP 페이지의 Model Attribute를 셋팅해준다. model tag에서 제공하는 Attribute는 아래와 같다.

Attribute	Required	Description
name	true	model attribute의 이름. Spring Form tag의 modelAttribute 속성과 같아야 함.
type	true	model attribute의 타입. 풀 패키지 명으로 정의. ex>org.anyframe.datatype.SearchVO

다음은 model tag의 정의 예이다.

```
<simpleweb:link id="jsonMovie" action="/simplejson.do" layout="jsonLayout" tiles="body:/WEB-INF/jsp/simpleweb-json/moviefinder/movie/list.jsp">
<simpleweb:model name="search"
type="pack.simpleweb.json.moviefinder.service.MovieSearchVO" />
<simpleweb:setProperty name="title">Simpeweb-JSON[1]</simpleweb:setProperty>
</simpleweb:link>
```

3.9.tiles tag

tiles tag는 submit tag와 link tag의 inner tag로 Tiles Definition의 요소를 정의해주는 역할을 한다. 위에서 언급했듯이 Tiles 요소는 submit tag와 link tag의 tiles attribute로 정의할 수 있다. 개발자가 여러개의 Tiles Definition을 정의해야할 때 submit tag와 link tag의 정의가 길어지는 것을 원하지 않는다면 tiles tag를 사용하도록 한다. tiles tag의 Attribute는 아래와 같다.

Attribute	Required	Description
name	true	Tiles Definition의 put-attribute에 들어갈 name
value	true	Tiles Definition의 put-attribute에 들어갈 value

다음은 tiles tag의 정의 예이다.

```
<simpleweb:tiles name="body" value="/WEB-INF/jsp/simpleweb-vo/moviefinder/movie/form.jsp"/>
```

3.10.validate tag

Spring JS에서는 dojo 기반의 Client단 Validation을 지원한다. Anyframe에서는 JavaScript 문을 사용하는 Spring JS를 간편하게 제공하고자 자주 사용하게 되는 요소에 대해서 Anyframe validation tag를 제공한다. Validation에 위배되는 필드에 대해서 Validation Error를 UI로 출력해 주며 Submit시 해당사항을 체크해준다. validate tag에서 제공하는 Attribute는 아래와 같다.

Attribute	Required	Description
id	true	validation 체크 대상이 되는 필드의 유일한 id. 반드시 input 요소의 id와 같아야 함.
type	false	필드에 적용할 widget type. default로 설정되는 widget type은 dijit.form.ValidationTextBox <ul style="list-style-type: none"> Number : dijit.form.NumberTextBox Date : dijit.form.DateTextBox

Attribute	Required	Description
		<ul style="list-style-type: none"> • CheckBox : dijit.form.CheckBox • RadioButton : dijit.form.RadioButton • Currency : dijit.form.CurrencyTextBox
required	false	필수 입력 값인지 체크
promptMessage	false	form에 입력 값을 넣을 때 보여줄 메시지
promptMessageKey	false	form에 입력 값을 넣을 때 보여줄 메시지의 key 값 (messageSource bean에 정의된 리소스에서 메시지 추출)
promptMessageArgs	false	promptMessageKey를 사용할 때 넘겨줄 argument
invalidMessage	false	validation에 위배되었을 때 보여줄 메시지
invalidMessageKey	false	validation에 위배되었을 때 보여줄 메시지 key 값 (messageSource bean에 정의된 리소스에서 메시지 추출)
invalidMessageArgs	false	invalidMessageKey를 사용할 때 넘겨줄 argument
constraints	false	validator에 넘겨줄 사용자가 정의한 parameter
regExp	false	필드 값을 체크하게될 Regular Expression
datePattern	false	type이 "Date"(dijit.form.DateTextBox)일 경우 format 정의. default는 yyyy-MM-dd
checked	false	type이 "CheckBox"(dijit.form.CheckBox)일 경우 check/uncheck 여부 결정
trim	false	입력 받은 문자열을 trim
currency	false	type이 "Currency"(dijit.form.CurrencyTextBox)일 경우 currency 설정. default 값은 "KRW"
value	false	필드에 value 지정

다음은 simpleweb-plugin 설치로 추가된 form이 정의된 form.jsp 파일의 일부이다.

```

<form:form modelAttribute="movie" method="post" action="/simple.do" id="movieForm"
name="movieForm" enctype="multipart/form-data">
<form:input path="title" cssClass="ct_input_g" cssErrorClass="text medium error" size="40"
maxLength="50" />
<simpleweb:validate id="title" required="true" promptMessage="Enter movie Title." />
...중략...
<form:input path="director" cssClass="ct_input_g" cssErrorClass="text medium error"
size="40" maxLength="50" />
<simpleweb:validate id="director" required="true" />
...중략...
<form:input path="actors" cssClass="ct_input_g" cssErrorClass="text medium error" size="60"
maxLength="50" /></td>
<simpleweb:validate id="actors" required="true" promptMessage="Enter Actors." />
...중략...
<form:input path="runtime" cssClass="ct_input_g" cssErrorClass="text medium error"
size="10" maxLength="3" /> min.
<simpleweb:validate id="runtime" type="Number" />
...중략...
<form:input path="releaseDate" cssClass="ct_input_g" cssErrorClass="text medium error"
maxLength="10" />
<c:if test="${empty movie.movieId}">
<simpleweb:validate id="releaseDate" type="Date" value="currentDate"/>
</c:if>
<c:if test="${not empty movie.movieId}">

```

```

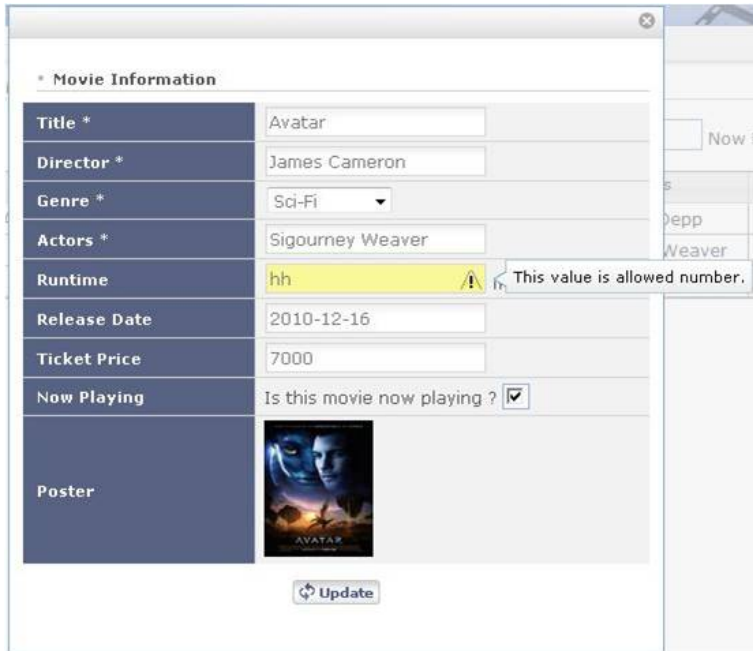
<simpleweb:validate id="releaseDate" type="Date"/>
</c:if>
... 중략...
<form:input path="ticketPrice" cssClass="ct_input_g" cssErrorClass="text medium error"
maxLength="7" />
<simpleweb:validate id="ticketPrice" type="Number" />
... 중략...
Is this movie now playing ? <form:checkbox path="nowPlaying" value="Y"/>

<simpleweb:submit id="addlink" form="movieForm" ... 중략... jsvalidate="true">
<simpleweb:setProperty name="title">
  " width="64" height="18"
border="0" />
</simpleweb:setProperty>
<simpleweb:setProperty name="hiddenDiv" value="hiddenDivPopup" />
<simpleweb:setProperty name="upload" value="true" />
<simpleweb:doublesubmit formName="movieForm" isSessionForm="true"/>
</simpleweb:submit>
.....

```

위의 코드에서 볼 수 있듯이 <simpleweb:validate> tag만 사용하면 단지 UI로 validation 위배 사항을 알려 줄 뿐이다. submit시 validation 오류를 체크하기 위해서는 반드시 submit tag나 link tag의 jsValidate 속성을 "true"로 설정해줘야 한다.

다음은 validate tag를 적용한 샘플 화면이다.



Anyframe에서 제공하는 유용한 value

Anyframe은 많이 쓰이는 타입의 필드에 대해 몇가지 유용한 value를 제공한다. 해당 요소는 아래와 같다.

- type이 "Date"인 필드의 value 값을 "currentDate"로 지정할 시 필드에 오늘 날짜로 초기 데이터 설정
- type을 정의 하지 않고(default로 "dijit.form.ValidationTextBox"로 widget type 설정) value 값을 "zipcode"로 지정할 시 NNN-NNN 형식의 우편번호 Validation 체크
- type을 정의 하지 않고(default로 "dijit.form.ValidationTextBox"로 widget type 설정) value 값을 "email"로 지정할 시 이메일 형태의 Validation 체크

※ Spring JS에서는 dojo 기반의 다양한 JavaScript를 지원한다. 자세한 사항은 아래의 Link를 참조하도록 한다.

- Spring JS Documentation : 바로가기 [<http://static.springsource.org/spring-webflow/docs/2.1.x/reference/htmlsingle/spring-webflow-reference.html>]
- dojo homepage : 바로가기 [<http://www.dojotoolkit.org/>]