



WEBDeK 구성

Supply Chain Management Innovation Company Introduction

1. 기반기술

Client Side



jQuery



jQuery UI



jQuery Plugins

HTML



JS



Server Side



MVC

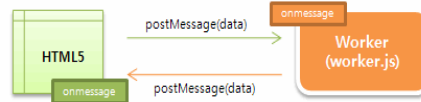
AOP

TASK

Transaction



2. 특징



Search: Q Calendar: 2020.01.26

SingleRange: SingleSearch: Q

일반 상세

번호	라벨 그룹 ID	라벨 ID	언어 코드	라벨	라벨 타입
1	BOTTOM	FROM	[KO]Korea	시작	
2	BOTTOM	LOGICAL	[KO]Korea	연결자	
3	BOTTOM	NUM	[KO]Korea	순번	
4	BOTTOM	OPER	[KO]Korea	연산자	
5	BOTTOM	SINGLE	[KO]Korea	값	
6	BOTTOM	TO	[KO]Korea	끝	
7	STD	ADDITION_CO...	[KO]Korea	추가 요건	
8	STD	ADDR	[KO]Korea	기본 주소	
9	STD	ADDRESC	[KO]Korea	상세 주소	
10	STD	ADDRESS	[KO]Korea	주소	
11	STD	ADDRID	[KO]Korea	주소록 ID	
12	STD	ADDRKEY1	[KO]Korea	출발 타임일	
13	STD	ADDRKEY2	[KO]Korea	도착 타임일	
14	STD	AKEY	[KO]Korea	문자열(자리)	
15	STD	ALLOCATION...	[KO]Korea	배차시간	
16	STD	ALLTID	[KO]Korea	약도설명	
17	STD	APIKEY	[KO]Korea	API연동 key	

```
$(document).ready(function(){
    gridList.setGrid({
        id: "gridList",
        module: "System",
        command: "SYSLABEL"
    });
});

function searchList(){
    if(validate.check("searchArea")){
        var param = inputList.setRangeParam("searchArea");

        gridList.gridList({
            id: "gridList",
            param: param
        });
    }
}

function saveData(){
    if(gridList.validationCheck("gridList", "all")){
        gridList.gridSave({
            id: "gridList",
            modifyType: "A"
        });
    }
}
```

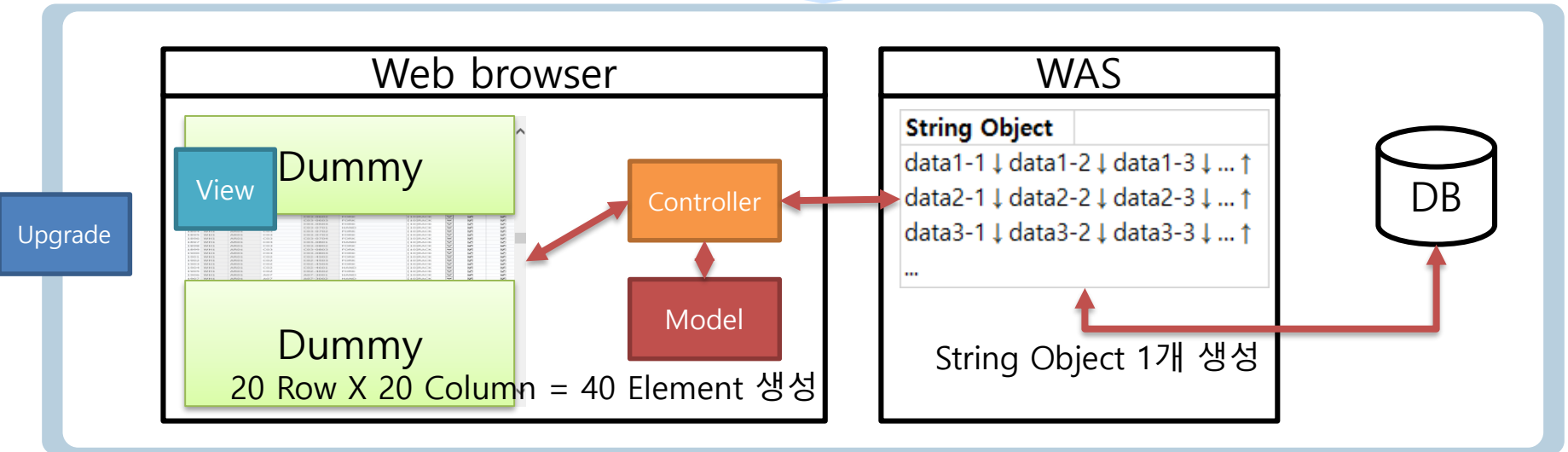
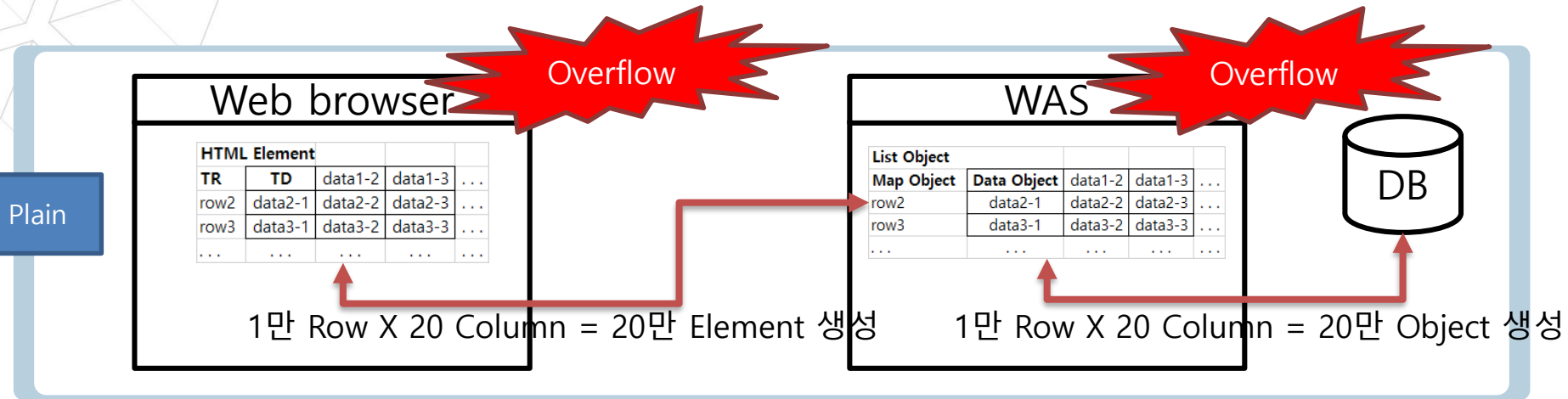
□ 웹표준 오픈소스 프레임워크 구성

- Client side : jQuery, HTML5
- Server side : Spring Framework
- WebSocket, WebWorker등의 최신 기술 사용

□ 프레임워크 주요 기능

- 대용량(몇 만건) 데이터 그리드 기능
- 경량의 강력한 에디트 그리드의 지원
- 공통 팝업, 콤보 등의 기능제공
- 멀티 input, 레인지 input 기능 지원
- CSS활용을 통한 다양한 디자인 적용가능
- jQuery UI방식의 간편한 코딩
- 개발 편의성 제공을 위한 중복 코딩 방지
- 다양한 Drag & Drop 기능 제공
- 개인화 기능을 통한 다양한 요구사항 충족

2. 특징



3. 개발 생산성

- 직관적인 개발 표현

✓ 개발자가 이해하기 쉬운 HTML코드 상 Attribute 기반의 코딩 지원

```
<input type="button" CB="Search SEARCH BTN_DISPLAY" />
<input type="button" CB="Save SAVE BTN_SAVE" />
<input type="button" CB="New ADD BTN_NEW" />
<input type="button" CB="Delete DELETE BTN_DELETE" />
<input type="button" CB="Reset RESET BTN_RESET" />
```

```
<dl>
  <dt CL="Search"></dt>
  <dd>
    <input type="text" class="input" name="Search" UIInput="S,SHAREMA" IName="Search" UIFormat="U"/>
  </dd>
</dl>
<dl>
  <dt CL="Calendar"></dt>
  <dd>
    <input type="text" class="input" name="Calendar" UIFormat="C Y" />
  </dd>
</dl>
<dl>
  <dt CL="SingleRange"></dt>
  <dd>
    <input type="text" class="input" name="SingleRange" UIInput="SR"/>
  </dd>
</dl>
```

```
<tbody id="gridList">
  <tr CGRow="true">
    <td GH="40" GCol="rownum">1</td>
    <td GH="40" GCol="rowCheck"></td>
    <td GH="80" GCol="text,LABELGID"></td>
    <td GH="150" GCol="text,LABELID"></td>
    <td GH="80" GCol="select,LANGCODE">
      <select class="input" CommonCombo="LANGKY">
      </select>
    </td>
    <td GH="150" GCol="text,LABEL"></td>
    <td GH="80" GCol="text,LABELTYPE"></td>
  </tr>
</tbody>
```

```
eSearch"></dt>
  <dd>
    <input type="text" class="input" name="SingleSearch" UIInput="SR,SHAREMA"/>
  </dd>
</dl>
```

3. 개발 생산성

- jQuery UI 와 동일한 방식의 Component 설정 방식 지원

```
gridList.setGrid({  
    id : "gridList",  
    module : "System",  
    command : "SYSLABEL"  
});
```

- Grid event에 대한 세부적인 call back function 제공

```
function gridListEventDataBindEnd(gridId, dataLength, excelLoadType){  
    //commonUtil.debugMsg("gridListEventDataBindEnd : ", arguments);  
}  
  
function gridListEventDataSortViewEnd(gridId) {  
    //commonUtil.debugMsg("gridListEventDataSortViewEnd : ", arguments);  
}  
  
function gridListEventDataViewSetEnd(gridId) {  
    //commonUtil.debugMsg("gridListEventDataSortViewEnd : ", arguments);  
}  
  
function gridListEventRowDbldclick(gridId, rowNum, colName, colValue){  
    //commonUtil.debugMsg("gridListEventRowDbldclick : ", arguments);  
}  
  
function gridListEventRowRemoveAfter(gridId, rowNum){  
    //commonUtil.debugMsg("gridListEventRowRemoveAfter : ", arguments);  
}
```

3. 개발 생산성

코드 최소화

✓ Query 기반의 간단한 코딩으로 데이터 표현 가능

- module : Mybatis namespace
- command : Mybatis bean name rule

```
gridList.setGrid({  
    id : "gridList",  
    module : "System",  
    command : "SYSLABEL"  
});
```

```
gridList.gridList({  
    id : "gridList",  
    param : param  
});
```

```
gridList.gridSave({  
    id : "gridList"  
});
```

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="System">  
  
    <select id="SYSLABEL_LIST" parameterType="hashmap" resultMap="DataMap">  
        ...  
    </select>  
  
    <insert id="SYSLABEL_INSERT" parameterType="hashmap">  
        ...  
    </insert>  
  
    <update id="SYSLABEL_UPDATE" parameterType="hashmap">  
        ...  
    </update>  
  
    <delete id="SYSLABEL_DELETE" parameterType="hashmap">  
        ...  
    </delete>  
</mapper>
```

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="Common">  
  
    <select id="COMCODE_LIST" parameterType="hashmap" resultMap="DataMap">  
        ...  
    </select>  
</mapper>
```

```
<select name="Combo" class="input" Combo="Common,COMCODE">  
    ...  
</select>
```