

I. DataCollection 컬럼 추가 삭제 가이드

본 가이드는 DataList의 컬럼 추가 및 삭제, DataMap의 컬럼 추가 가이드입니다.

1. GridView 컴포넌트와 연동한 다건 데이터 처리 예제 (multiRow_Data.xml)

- 본 샘플에서는 DataList에 컬럼 추가 및 삭제를 구현하는 예제입니다.
- 아래의 코드에서 사용하는 com.insertColumn, com.removeColumn은 본 문서 하단에 정의했습니다.

```
scwin.onpageload = function () {  
    // 테스트를 위해서 JSON 데이터를 직접 DataList에서 세팅합니다.  
    // DataList(dlt_memberList)에 Array 형태로 데이터를 세팅해야 하기 때문에 배열로 데이터로 정의해야 합니다.  
    var jsonData = [  
        { "id": "P0001", "name": "홍길동", "tel": "010-2323-4323", "region": "서울" },  
        { "id": "P0002", "name": "김성현", "tel": "010-5343-4323", "region": "인천" },  
        { "id": "P0003", "name": "박만수", "tel": "010-5322-4323", "region": "김포" },  
        { "id": "P0004", "name": "김영호", "tel": "010-1231-7343", "region": "부천" },  
        { "id": "P0005", "name": "박기영", "tel": "010-5343-6534", "region": "서울" }  
    ];  
    dlt_memberList.setJSON(jsonData);  
};  
  
scwin.btn_insertColumn_onclick = function (e) {  
    // DataList에 컬럼을 추가합니다.  
    com.insertColumn(dlt_memberList, "email", { name: "이메일", dataType: "text" });  
    dlt_memberList.setCellData(0, "email", "gdhong@test.com");  
    console.log(dlt_memberList.getAllJSON());  
};  
  
scwin.btn_removeColumn_onclick = function (e) {  
    // DataList에 컬럼을 삭제합니다.  
    com.removeColumn(dlt_memberList, "email");  
    console.log(dlt_memberList.getAllJSON());  
};
```

아래의 화면은 multiRow_Data.xml 파일을 실행해서 [컬럼 추가] 버튼 클릭 후, [컬럼 삭제] 버튼을 클릭했을 때 브라우저 개발자 도구 콘솔 창에 출력된 로그를 보여주는 화면입니다.

콘솔 창에 로그를 보면 email 컬럼이 생성되었다가, email 컬럼이 삭제된 것을 확인할 수 있습니다.

WebSquare
localhost/websquare/websquare.html?w2xPath=/project/SBS/multiRow_Data.xml

Multi Row Data

컬럼 추가

컬럼 삭제

아이디	성명	연락처	지역
P0001	홍길동	010-2323-4323	서울
P0002	김성현	010-5343-4323	인천
P0003	박만수	010-5322-4323	김포
P0004	김영호	010-1231-7343	부천
P0005	박기영	010-5343-6534	서울

Elements

Console

Sources

Network

Performance

Memory

Application

Security

Lighthouse

Recorder

Performance insights

top

Filter

Default levels

3 Issues: 1 2

1 hidden

DataList에 컬럼을 추가

VM286:25

VM286:26

(5) [{}]

0: {id: 'P0001', name: '홍길동', tel: '010-2323-4323', region: '서울', email: 'gdhong@test.com', ...}

1: {id: 'P0002', name: '김성현', tel: '010-5343-4323', region: '인천', email: '', ...}

2: {id: 'P0003', name: '박만수', tel: '010-5322-4323', region: '김포', email: '', ...}

3: {id: 'P0004', name: '김영호', tel: '010-1231-7343', region: '부천', email: '', ...}

4: {id: 'P0005', name: '박기영', tel: '010-5343-6534', region: '서울', email: '', ...}

length: 5

[[Prototype]]: Array(0)

DataList에 컬럼을 삭제

VM286:32

VM286:33

(5) [{}]

0: {id: 'P0001', name: '홍길동', tel: '010-2323-4323', region: '서울', rowStatus: 'U'}

1: {id: 'P0002', name: '김성현', tel: '010-5343-4323', region: '인천', rowStatus: 'R'}

2: {id: 'P0003', name: '박만수', tel: '010-5322-4323', region: '김포', rowStatus: 'R'}

3: {id: 'P0004', name: '김영호', tel: '010-1231-7343', region: '부천', rowStatus: 'R'}

4: {id: 'P0005', name: '박기영', tel: '010-5343-6534', region: '서울', rowStatus: 'R'}

length: 5

[[Prototype]]: Array(0)

2. DataList를 이용해서 단건의 데이터 처리 예제(singleRow_Data1.xml)

- 본 예제는 DataList을 이용해서 1건의 데이터를 처리하는 예제입니다.
- DataList의 initRowPostion을 0으로 세팅하면 DataList에 데이터가 세팅되는 순간 Row의 Postion을 첫번째 Row로 이동시켜주어서 GridView가 없이 DataList를 DataMap처럼 사용할 수 있습니다.
- 이 방식을 사용하면 DataList로 단건 데이터와 다건 데이터를 모두 사용할 수 있습니다.
- 다만, DataList는 Array 형태의 데이터만 세팅할 수 있기 때문에 서버에서 데이터를 전달할 때 1건의 데이터도 Array 형태로 전달해야 합니다.
- 단건 데이터(1개 Row)의 경우 DataMap의 사용을 추천드리지만, 부득이한 사정으로 DataList와 DataMap을 구분하면서 사용하기 어려운 경우, GridView가 없는 화면에서 DataList를 단건 데이터 처리 용으로도 사용할 수 있습니다.
- 대부분의 모든 프로젝트에서 단건 데이터(1개 Row)의 경우 DataMap을 사용하지만, SK 텔레콤 차세대 프로젝트에서 마이플랫폼의 DataSet을 DataCollection으로 전환할 때, DataSet을 데이터 Row 수에 따라, DataList와 DataMap으로 구분해서 전환 개발을 하기가 어려워서, 두 경우 모두 DataList에 처리한 사례가 있습니다.

```

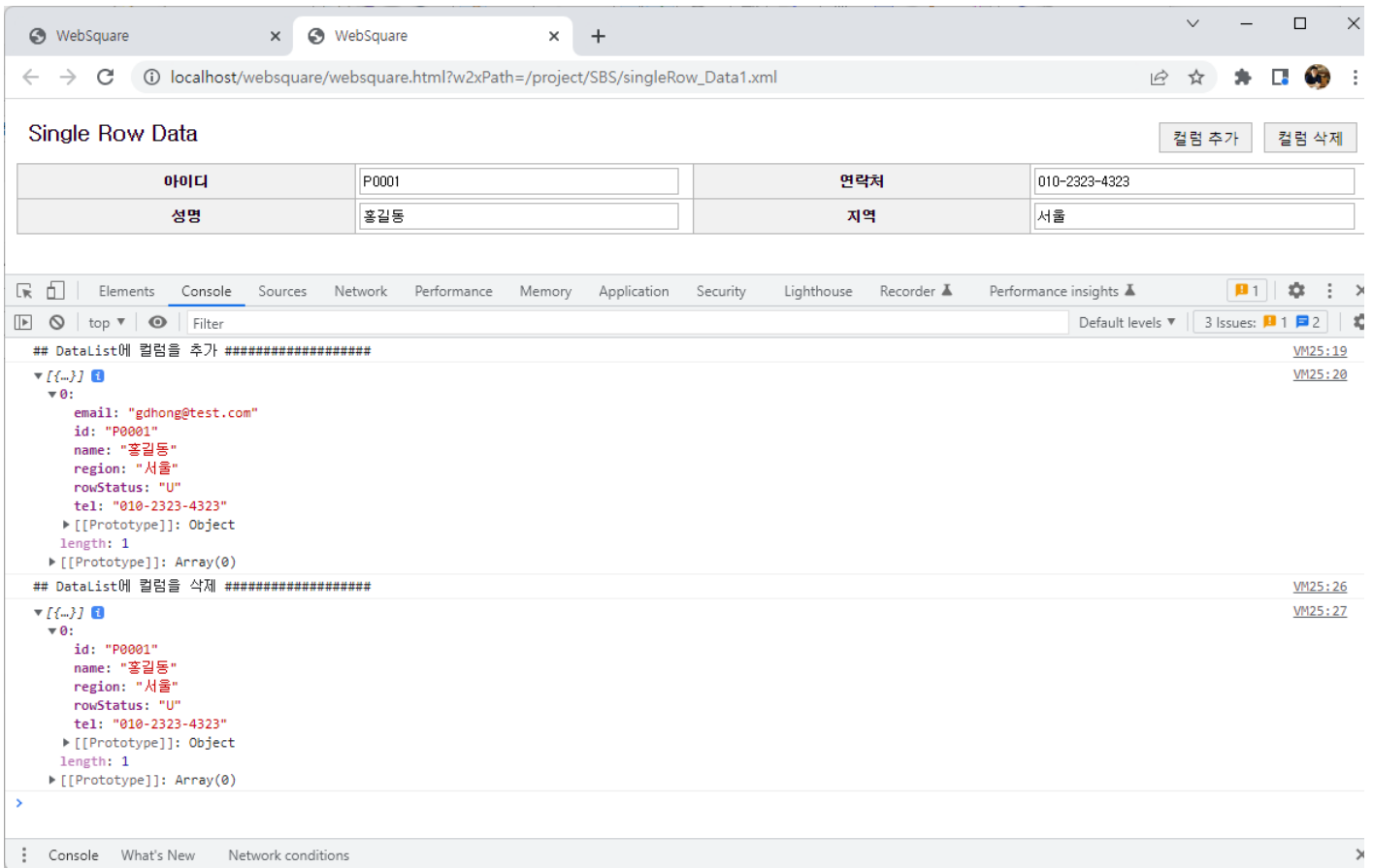
scwin.onpageload = function () {
// 1건의 데이터로 Array 형태로 정의해서 세팅해야 합니다.
var jsonData = [
{ "id": "P0001", "name": "홍길동", "tel": "010-2323-4323", "region": "서울" }
];
dlt_memberList.setJSON(jsonData);
};

scwin.btn_insertColumn_onclick = function (e) {
// DataList에 컬럼을 추가합니다.
com.insertColumn(dlt_memberList, "email", { name: "이메일", dataType: "text" });
dlt_memberList.setCellData(0, "email", "gdhong@test.com");
console.log("## DataList에 컬럼을 추가 #####");
console.log(dlt_memberList.getAllJSON());
};

scwin.btn_removeColumn_onclick = function (e) {
// DataList에 컬럼을 삭제합니다.
com.removeColumn(dlt_memberList, "email");
console.log("## DataList에 컬럼을 삭제 #####");
console.log(dlt_memberList.getAllJSON());
};

```

DataList(dlt_memberList)의 initRowPostion을 0으로 세팅하면, DataList에 데이터가 세팅될 때, Row Postion을 첫번째 Row로 이동시켜 줍니다.



3. DataMap를 이용해서 단건의 데이터 처리 예제(singleRow_Data2.xml)

- 본 예제는 DataMap을 이용해서 1건의 데이터를 처리하는 예제입니다.
- DataMap의 경우에는 컬럼을 추가하는 API만 웹스퀘어 엔진에서 제공하고, 삭제하는 API는 제공하지 않고 있어서 컬럼 추가만 가능합니다.

```
scwin.onpageload = function () {
// DataMap에 세팅할 때는 Array 형태 아닌 JSON(Key, Value) 형태로 데이터를 세팅해야 합니다.
var jsonData = { "id": "P0001", "name": "홍길동", "tel": "010-2323-4323", "region": "서울" };
dma_member.setJSON(jsonData);
};

scwin.btn_insertColumn_onclick = function (e) {
// DataMap에 컬럼을 추가합니다.
com.insertColumn(dma_member, "email", { name: "이메일", dataType: "text" });
dma_member.set("email", "gdhong@test.com");
console.log("## DataMap에 컬럼을 추가 #####");
console.log(dma_member.getJSON());
};
```

아래의 화면은 singleRow_Data2.xml 파일을 실행해서 [컬럼 추가] 버튼을 클릭했을 때 브라우저 개발자 도구 콘솔 창에 출력된 로그를 보여주는 화면입니다.

콘솔 창에 로그를 보면 email 컬럼이 생성된 것을 확인할 수 있습니다.

WebSquare

localhost/websquare/websquare.html?w2xPath=/project/SBS/singleRow_Data2.xml

Single Row Data

컬럼 추가

아이디	P0001	연락처	010-2323-4323
성명	홍길동	지역	서울

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights

top Filter Default levels 3 Issues: 1 2

```
## DataMap에 컬럼을 추가 #####
{
  id: 'P0001',
  name: '홍길동',
  tel: '010-2323-4323',
  region: '서울',
  email: 'gdhong@test.com'
}
id: "P0001"
name: "홍길동"
region: "서울"
tel: "010-2323-4323"
[[Prototype]]: Object
```

Console What's New Network conditions

4. DataList 및 DataMap 컬럼 제어를 위한 공통 함수 스크립트

```

var com = {};

/**
 * DataList 또는 DataMap 객체에 컬럼을 추가한다.
 *
 * @memberOf com
 * @function com.insertColumn
 * @date 2022.07.12
 * @param {Object} dataDcObj DataList 또는 DataMap 객체
 * @param {String} columnId 컬럼 아이디
 * @param {Object} columnInfo 컬럼 정보 JSON 객체 (name, dataType)
 * @author InswaveSystems
 * @example
com.insertColumn(dlt_memberList, "email", {name:"이메일", dataType:"text"});
 */
com.insertColumn = function (dataDcObj, columnId, columnInfo) {
  var dataType = dataDcObj.getObjectType().toLowerCase();
  if (dataType === "datamap") {
    columnInfo.id = columnId;
    dataDcObj.addColumn(columnInfo);
  } else if (dataType === 'datalist') {
    dataDcObj.insertColumn(columnId, columnInfo);
  }
};

/**
 * DataList 객체에 컬럼을 삭제한다.
 *
 * @memberOf com
 * @function com.removeColumn
 * @date 2022.07.12
 * @param {Object} dataListObj DataList 객체
 * @param {String} columnId 컬럼 아이디
 * @author InswaveSystems
 * @example
com.removeColumn(dlt_memberList, "email");
 */
com.removeColumn = function (dataListObj, columnId) {
  dataListObj.removeColumn(columnId);
};

```

1. com.insertColumn

- DataList와 DataMap에 컬럼을 추가하는 기능을 제공합니다.
- com.insertColumn 내부에서 DataCollection의 유형(DataList, DataMap)을 체크해서 각 DataCollection 유형에 따라 분기 처리하도록 구현했습니다.

2. com.removeColumn

- DataList에 컬럼을 추가하는 기능을 제공합니다.
- 현재 DataMap의 경우 컬럼을 삭제하는 기능을 엔진에서 제공하지 않아서, DataMap은 컬럼을 삭제할 수 없습니다



common.js



multiRow...Data.xml



singleRo...ata1.xml



singleRo...ata2.xml