

BIX5

위젯 커스터마이징 튜토리얼

Widget Customization Tutorial



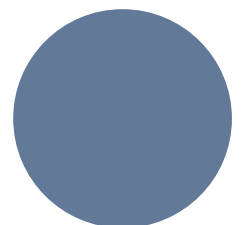
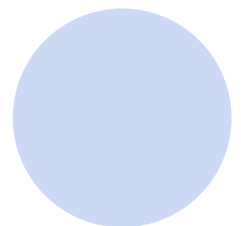
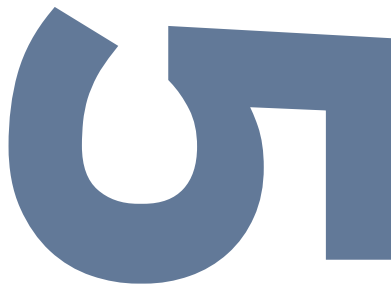
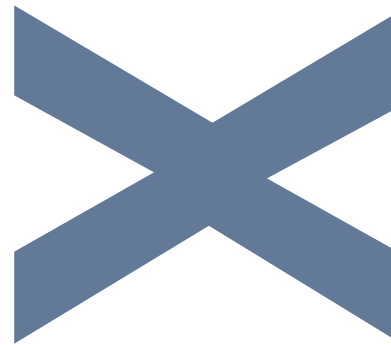
정품을 구입하신 고객에게는 기술상담 및 지원을 제공합니다.

(주)빅스소프트

서울시 영등포구 영신로 220, 610 호 (영등포동 8 가, KnK 디지털타워)

Tel. 02-2655-9784, bix5cs@bixsoft.net

BIXSOFT 빅스소프트



목차

1. 개발자가 확장할 수 있는 HTML 위젯 만들기	2
1.1. 버튼 위젯 만들어 보기	2
1.1.1. 소스	8
1.2. 카드 위젯 만들어 보기	10
1.2.1. 소스	19
1.3. 검색용 위젯 만들어 보기	20
1.3.1. 소스	25
1.4. 위젯간 이벤트 송/수신 예제 만들어 보기	26
1.4.1. 소스	33
2. 컴포넌트 위젯 확장하기	35
2.1. 차트 위젯 확장하기 I	35
2.1.1. 소스	38
2.2. 차트 위젯 확장하기 II	39
2.2.1. 소스	43
2.3. 그리드 위젯 확장하기 I	44
2.3.1. 소스	47
2.4. 그리드 위젯 확장하기 II	48
2.4.1. 소스	54
2.5. 맵차트 위젯 확장하기 I	56
2.5.1. 소스	60
2.6. 맵차트 위젯 확장하기 II	61
2.6.1. 소스	65

1. 개발자가 확장할 수 있는 HTML위젯 만들기

BIX5는 기본 제공 위젯 외에도 개발자가 직접 HTML 위젯을 만들 수 있는 기능을 지원합니다. HTML 위젯은 사용자가 HTML, CSS, JavaScript를 활용해 레이아웃과 이벤트를 자유롭게 정의할 수 있어 다양한 요구사항에 맞춰 위젯을 커스터마이징하고 확장할 수 있는 강력한 위젯입니다. 이를 통해 제품의 기능을 한층 더 유연하고 효율적으로 확장할 수 있습니다.

1.1. 버튼 위젯 만들어 보기

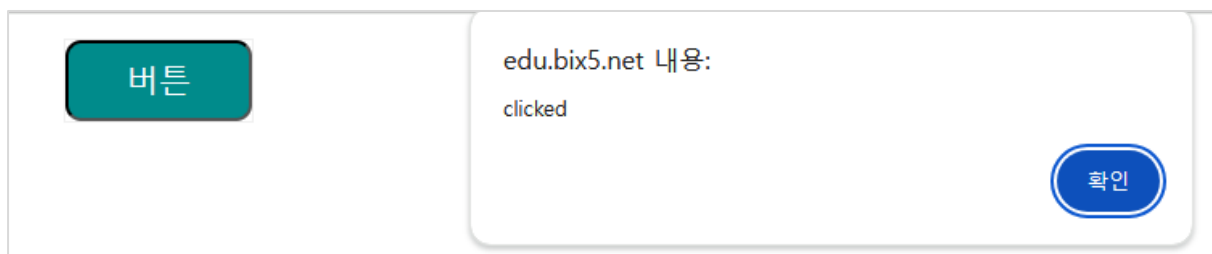
학습 목표

- HTML 위젯을 이용하여 버튼 위젯을 만들 수 있다.
- 클릭 이벤트를 등록하여 사용자 함수를 작성할 수 있다.
- 사용자 정의 메뉴를 만들 수 있다.

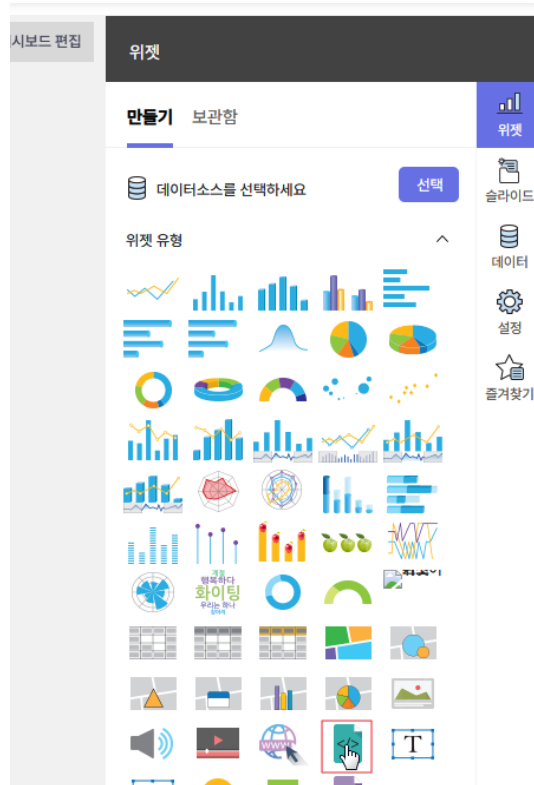
사전 요구 사항

- HTML, CSS, JavaScript를 다뤄본 경험
- BIX5를 이용하여 대시보드 제작한 경험
- 대시보드 개발자 권한으로 로그인

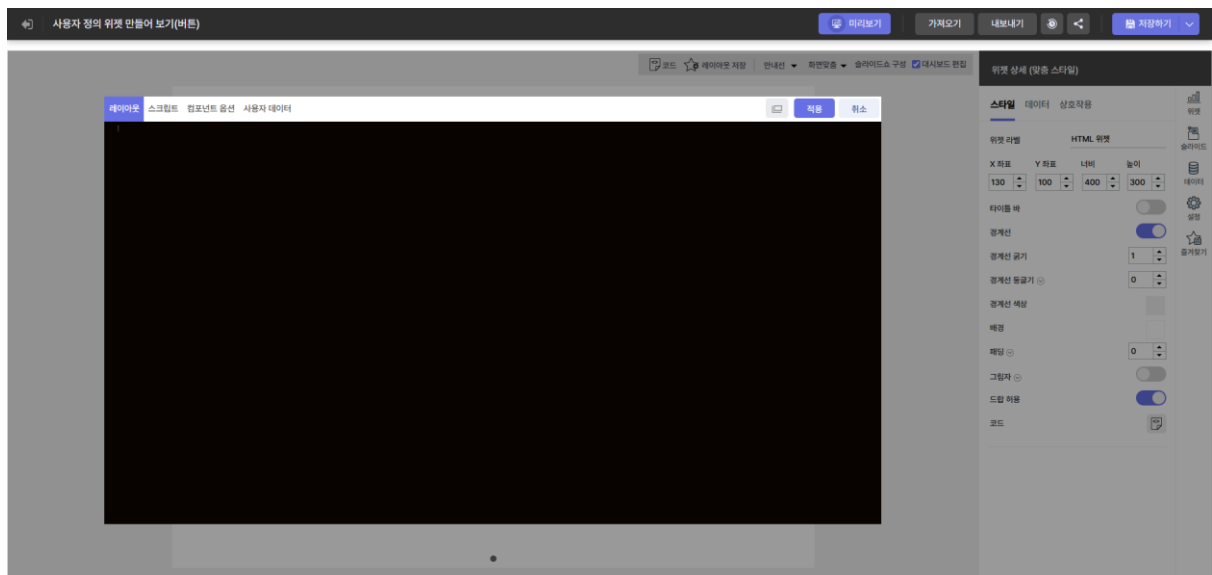
최종 결과



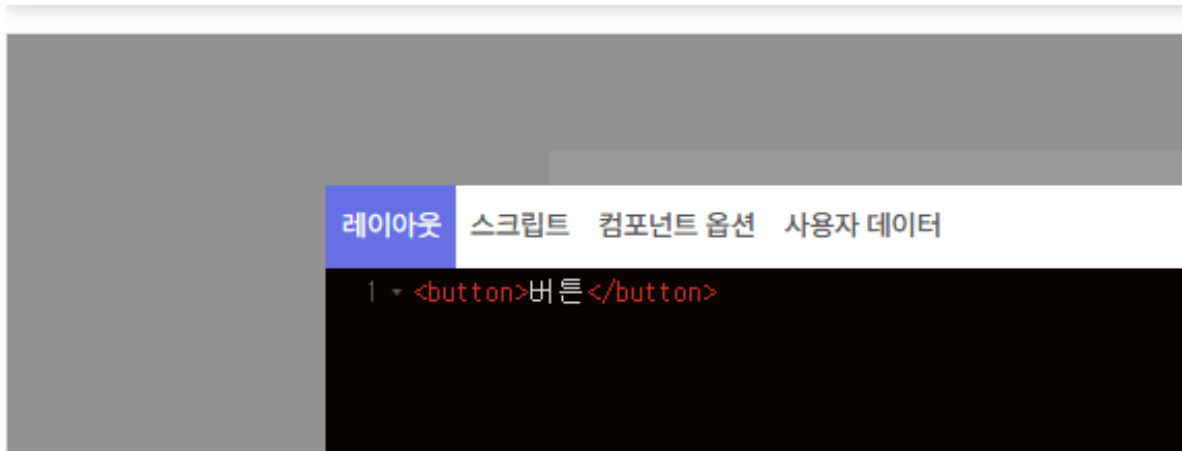
1. 위젯 만들기에서 HTML 위젯을 생성합니다.



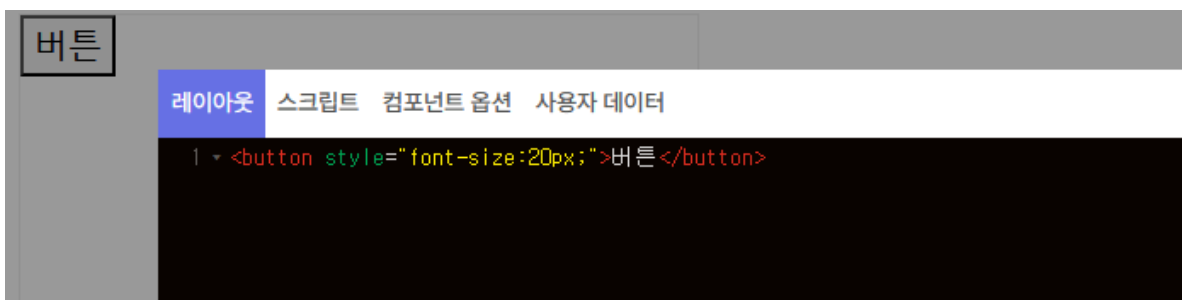
2. 만든 위젯의 코드 에디터로 진입합니다.



3. 레이아웃에 HTML 을 작성합니다.



4. 스타일을 작성합니다.



스타일 태그도 작성 가능합니다.



스타일 적용 범위를 위젯으로 제한하기 위하여 #{{id}}를 붙여 작성합니다. 그렇지 않을 경우 문서 전체

범위에 스타일이 적용됩니다.

템플릿 엔진의 문법 및 사용법에 대한 자세한 내용은 BIX5_Server_헬퍼_함수_가이드.pdf를 참고해주세요.

5. 클릭 이벤트를 등록합니다.

HTML 위젯

레이아웃

```

1 <style>
2   #{{id}} button{
3     width:100%;
4     height:100%;
5     background-color:darkcyan;
6     color:#ffffff;
7     border-radius:10px;
8   }
9 </style>
10 <button style="font-size:20px;" onclick="widget._btnClickHandler(event);">버튼</button>

```

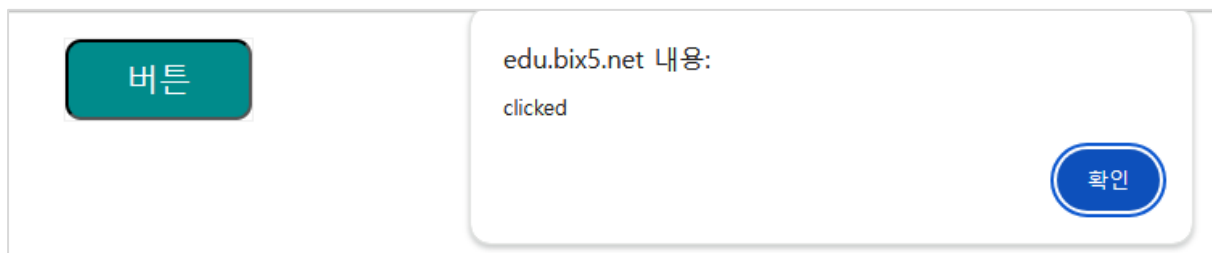
스크립트

```

1 <script>
2   widget._btnClickHandler = function(event){
3     alert("clicked");
4   }
5

```

6. 미리보기에서 실행해봅니다.



7. 사용자 정의 메뉴를 만들기 위해 코드 에디터 → 컴포넌트 옵션에 진입하여 `propertyOptions`을 작성합니다. 자세한 작성 방법은 BIX5_Server_위젯 사용자 정의 메뉴 가이드.pdf를 참고하세요.

레이아웃 스크립트 컴포넌트 옵션 사용자 데이터

```

1 {
2   "customStyle": true,
3   "propertyOptions": [
4     {
5       "label": "라벨",
6       "control": "TextInput",
7       "attribute": "label"
8     }
9   ],
10  "layoutMetaData": {},
11  "isMultiView": false,
12  "multiViewRow": 2
13 }
14 ],
15 "layoutMetaData": {
16   "label": "버튼"
17 },
18 "isMultiView": false,|

```

```

{
  "label": "라벨",
  "control": "TextInput",
  "attribute": "label"
}

```

8. 레이아웃을 동적으로 구성하도록 처리합니다.

레이아웃 스크립트 컴포넌트 옵션 사용자 데이터

```

1 <style>
2   #{id} button{
3     width:100%;
4     height:100%;
5     background-color:darkcyan;
6     color:#ffffff;
7     border-radius:10px;
8   }
9 </style>
10 <button style="font-size:20px;" onclick="widget._btnClickHandler(event);">{{meta.label}}</button>

```

9. 생성된 메뉴에서 텍스트를 입력하면 라벨이 변경되는 것을 확인할 수 있습니다.

교육_나만의 버튼 위젯 만들어 보기

리뷰하기가져오기내보내기저장하기

코드레이아웃 저장인라인화면모음순라이트박스 구성대시보드 편집

버튼입니다

위젯 상세 (맞춤 스타일)

스타일

데이터 상호작용

위젯 라벨HTML 위젯

X 좌표Y 좌표너비높이

351511852

타이틀 바

경계선

경계선 굵기

1

경계선 둥글기

0

경계선 색상

배경

패딩

0

그림자

드롭 차움

코드

라벨

버튼입니다

1.1.1. 소스

레이아웃

```
<style>
    #{{id}} button{
        width:100%;
        height:100%;
        background-color:darkcyan;
        color:#ffffff;
        border-radius:10px;
    }
</style>
<button style="font-size:20px;"
onclick="widget._btnClickHandler(event);">{{meta.label}}</button>
```

스크립트

```
<script>
    widget.componentReadyHandler = function(componentId){
    }

    widget._btnClickHandler = function(event){
        alert("clicked");
    }
    widget.componentRemoveHandler = function(){

    }
</script>
```

컴포넌트 옵션

```
{
    "customStyle": true,
    "propertyOptions": [
        {
            "label": "라벨",
            "control": "TextInput",
            "attribute": "label"
        }
    ],
    "layoutMetaData": {
        "label": "버튼입니다"
    },
}
```

```
"isMultiView": false,
"multiViewRow": 2,
"multiViewColumn": 2,
"multiViewSubCategoryField": {},
"multiViewFillType": null,
"marginLeft": 0,
"marginTop": 0,
"marginRight": 0,
"marginBottom": 0,
"isAutoServiceStart": true,
"dedicated": false,
"isDataLimit": false,
"dataLimit": 10000,
"referenceResources": [],
"defaultFontSize": 16,
"useDataTransform": false,
"dataTransform": {
  "transpose": {
    "index": [],
    "columns": [],
    "values": [],
    "valueName": null
  }
},
"bindingSource": "",
"referenceSources": [],
"classList": [],
"interactiveFiltering": false,
"dynamicFiltering": true
}
```

1.2. 카드 위젯 만들어 보기

학습 목표

- HTML 위젯을 이용하여 나만의 카드 위젯을 만들고 데이터셋을 연동하여 출력할 수 있다.
- 사용자 정의 메뉴를 만들 수 있다.

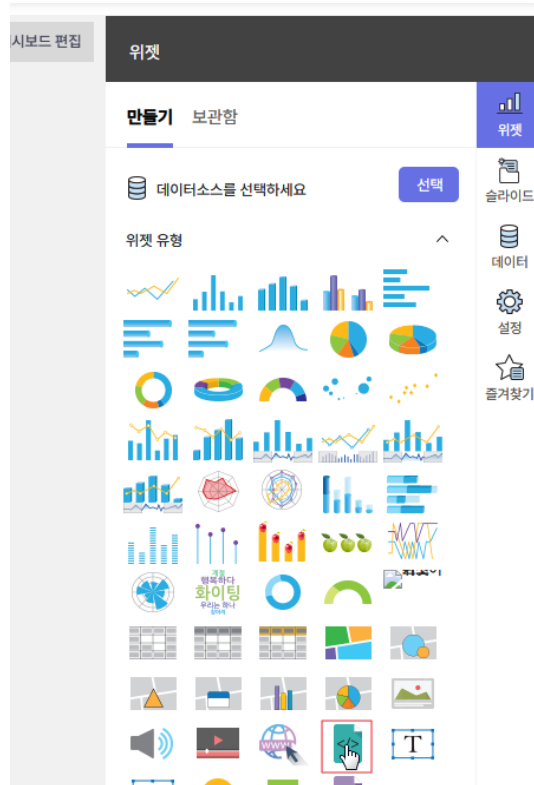
사전 요구 사항

- HTML, CSS, JavaScript 를 다뤄본 경험
- BIX5 를 이용하여 대시보드 제작한 경험
- 대시보드 개발자 권한으로 로그인

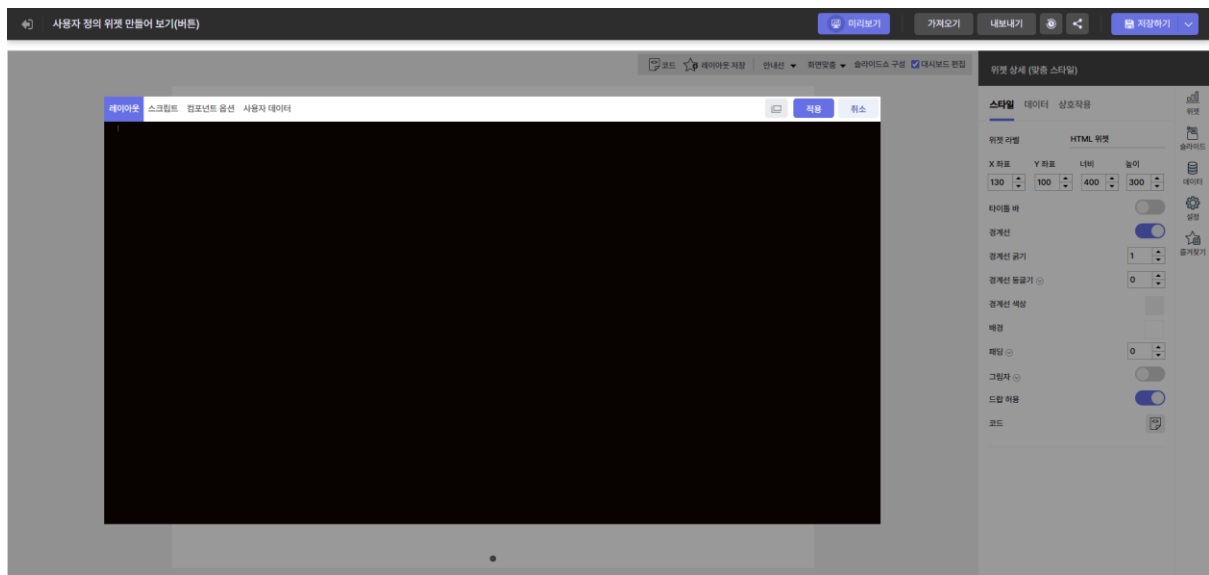
최종 결과



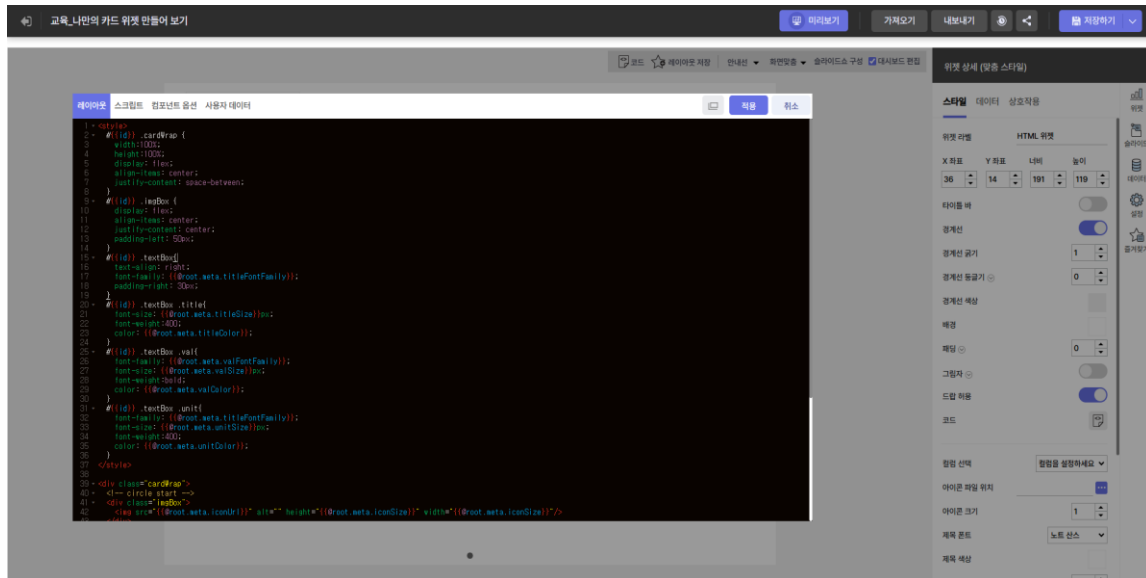
1. 위젯 만들기에서 HTML 위젯을 생성합니다.



2. 만든 위젯의 코드 에디터로 진입합니다.



3. 레이아웃을 작성합니다. 아래 내용을 복사하여 레이아웃 코드 편집 창에 붙여넣기 합니다.



```
<style>
  #{{id}} .cardWrap {
    width:100%;
    height:100%;
    display: flex;
    align-items: center;
    justify-content: space-between;
  }
  #{{id}} .imgBox {
    display: flex;
    align-items: center;
    justify-content: center;
    padding-left: 50px;
  }
  #{{id}} .textBox{
    text-align: right;
    font-family: {{@root.meta.titleFontFamily}};
    padding-right: 30px;
  }
  #{{id}} .textBox .title{
    font-size: {{@root.meta.titleSize}}px;
    font-weight:400;
    color: {{@root.meta.titleColor}};
  }
  #{{id}} .textBox .val{
    font-family: {{@root.meta.valFontFamily}};
    font-size: {{@root.meta.valSize}}px;
    font-weight:bold;
  }
</style>
```

```

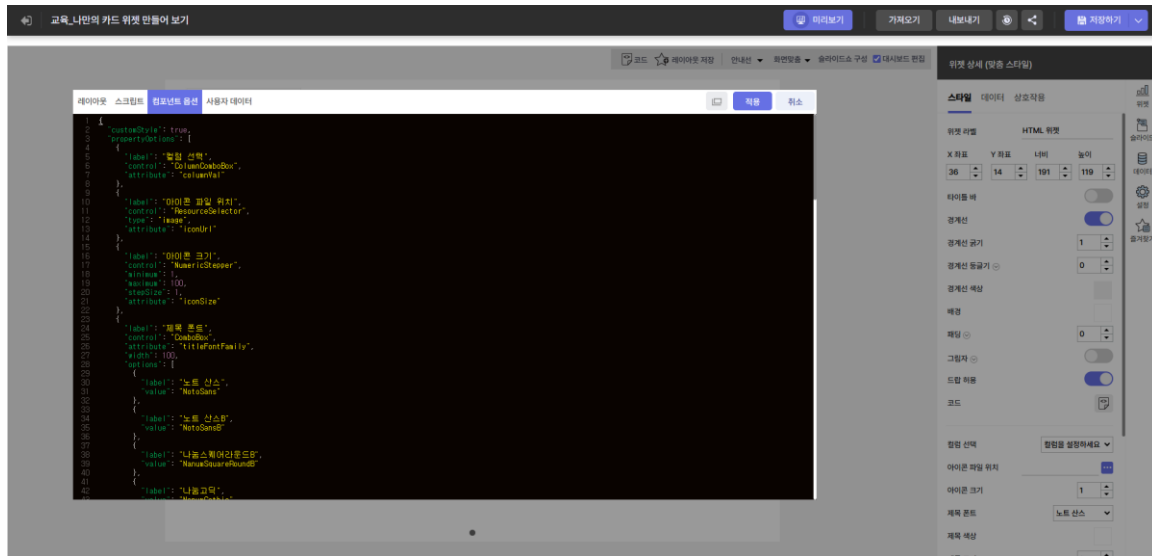
        color: {{@root.meta.valColor}};
    }
    #{{id}} .textBox .unit{
        font-family: {{@root.meta.titleFontFamily}};
        font-size: {{@root.meta.unitSize}}px;
        font-weight:400;
        color: {{@root.meta.unitColor}};
    }
</style>

<div class="cardWrap">
    <!-- circle start -->
    <div class="imgBox">
        {{#if @root.meta.iconUrl}}
        
        {{/if}}
    </div>
    <!-- circle start -->

    <!-- text start -->
    <div class="textBox">
        <div class="title">{{@root.meta.columnVal.label}}</div>
        <div class="val">
            {{#if data}}
            {{#with data.[0]}}
            {{DataFieldFormatter this @root.meta.columnVal}}
            {{/with}}
            {{else}}
            -
            {{/if}}
            <span class="unit">{{@root.meta.unitText}}</span>
        </div>
    </div>
    <!-- text end -->
</div>

```

4. 사용자 정의 메뉴를 만들기 위해 컴포넌트 옵션에 다음을 복사, 붙여넣기 합니다.



```
{
  "customStyle": true,
  "propertyOptions": [
    {
      "label": "컬럼 선택",
      "control": "ColumnComboBox",
      "attribute": "columnVal"
    },
    {
      "label": "아이콘 파일 위치",
      "control": "ResourceSelector",
      "type": "image",
      "attribute": "iconUrl"
    },
    {
      "label": "아이콘 크기",
      "control": "NumericStepper",
      "minimum": 1,
      "maximum": 100,
      "stepSize": 1,
      "attribute": "iconSize"
    },
    {
      "label": "제목 폰트",
      "control": "ComboBox",
      "attribute": "titleFontFamily",
      "width": 100,
      "options": [
        {

```

```

        "label": "노트 산스",
        "value": "NotoSans"
    },
    {
        "label": "노트 산스 B",
        "value": "NotoSansB"
    },
    {
        "label": "나눔스퀘어라운드 B",
        "value": "NanumSquareRoundB"
    },
    {
        "label": "나눔고딕",
        "value": "NanumGothic"
    },
    {
        "label": "나눔고딕 B",
        "value": "NanumGothicB"
    },
    {
        "label": "맑은 고딕",
        "value": "Malgun Gothic"
    },
    {
        "label": "굴림",
        "value": "gulim"
    },
    {
        "label": "Roboto",
        "value": "Roboto"
    }
]
},
{
    "label": "제목 색상",
    "control": "ColorSelector",
    "attribute": "titleColor"
},
{
    "label": "제목 크기",
    "control": "NumericStepper",
    "minimum": 1,
    "maximum": 100,
    "stepSize": 1,
    "attribute": "titleSize"
},
{

```



```

    "label": "단위",
    "control": "TextInput",
    "attribute": "unitText"
  },
  {
    "label": "단위 색상",
    "control": "ColorSelector",
    "attribute": "unitColor"
  },
  {
    "label": "단위 크기",
    "control": "NumericStepper",
    "minimum": 1,
    "maximum": 100,
    "stepSize": 1,
    "attribute": "unitSize"
  },
  {
    "label": "값 폰트",
    "control": "ComboBox",
    "attribute": "valFontFamily",
    "width": 100,
    "options": [
      {
        "label": "노트 산스",
        "value": "NotoSans"
      },
      {
        "label": "노트 산스 B",
        "value": "NotoSansB"
      },
      {
        "label": "나눔스퀘어라운드 B",
        "value": "NanumSquareRoundB"
      },
      {
        "label": "나눔고딕",
        "value": "NanumGothic"
      },
      {
        "label": "나눔고딕 B",
        "value": "NanumGothicB"
      },
      {
        "label": "맑은 고딕",
        "value": "Malgun Gothic"
      }
    ]
  },

```

```

        {
            "label": "굴림",
            "value": "gulim"
        },
        {
            "label": "Roboto",
            "value": "Roboto"
        }
    ]
},
{
    "label": "값 색상",
    "control": "ColorSelector",
    "attribute": "valColor"
},
{
    "label": "값 크기",
    "control": "NumericStepper",
    "minimum": 1,
    "maximum": 100,
    "stepSize": 1,
    "attribute": "valSize"
}
],
"layoutMetaData": {
    "columns": [],
    "columnVal": {},
    "iconUrl": "",
    "unitText": "",
    "iconSize": 38,
    "titleFontFamily": "NotoSansB",
    "titleColor": "#000000",
    "titleSize": 20,
    "unitColor": null,
    "unitSize": 12,
    "valFontFamily": "NotoSans",
    "valColor": "#000000",
    "valSize": 21
},
"isMultiView": false,
"multiViewRow": 2,
"multiViewColumn": 2,
"multiViewSubCategoryField": {},
"multiViewFillType": null,
"marginLeft": 0,
"marginTop": 0,
"marginRight": 0,

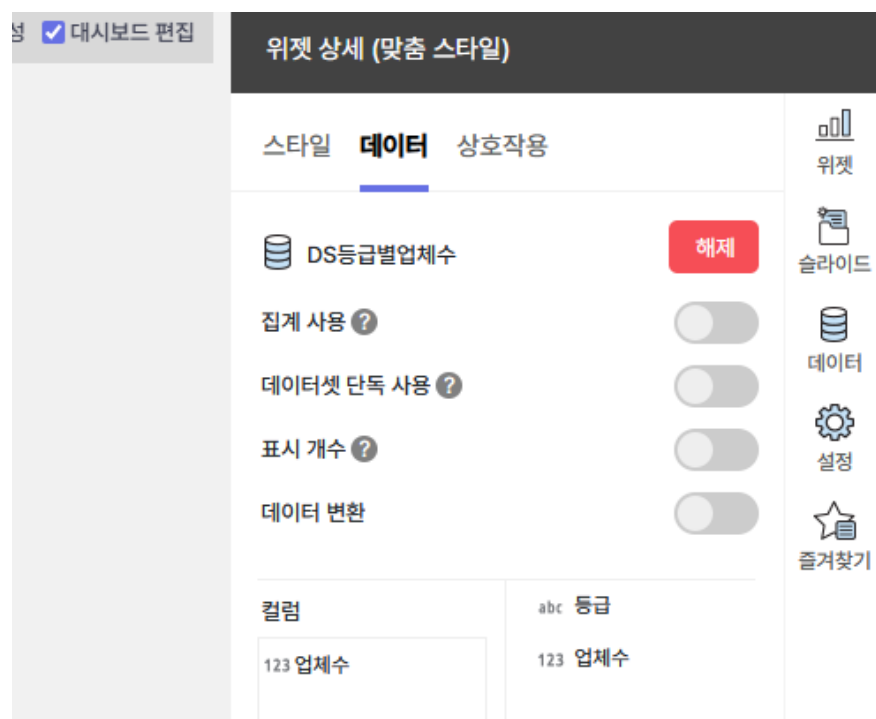
```

```

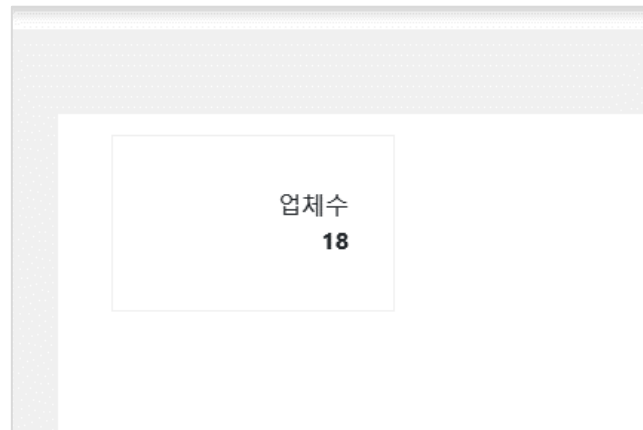
"marginBottom": 0,
"isAutoServiceStart": true,
"dedicated": false,
"isDataLimit": false,
"dataLimit": 10000,
"referenceResources": [
],
"defaultFontSize": 16,
"useDataTransform": false,
"dataTransform": {
  "transpose": {
    "index": [],
    "columns": [],
    "values": [],
    "valueName": ""
  }
},
"bindingSource": "",
"referenceSources": [],
"classList": [],
"interactiveFiltering": false,
"dynamicFiltering": true
}

```

5. 카드 위젯이 만들어졌습니다. 데이터 메뉴로 이동하여 샘플 데이터 소스 → DS 등급별업체수 데이터셋을 바인딩하고 업체수를 컬럼으로 설정합니다.



6. 카드 위젯에 값이 나오는 것을 확인할 수 있습니다.



7. 카드 위젯을 클릭하여 우측 상세 메뉴에서 옵션을 조정합니다.



1.2.1. 소스

본문 참조하세요.

1.3. 검색용 위젯 만들어 보기

학습 목표

- 클릭 이벤트를 등록하여 사용자 함수를 작성할 수 있다.
- HTML 위젯을 이용하여 데이터셋 필터 값을 입력 받아 원격데이터서비스에 파라미터를 전달할 수 있다.

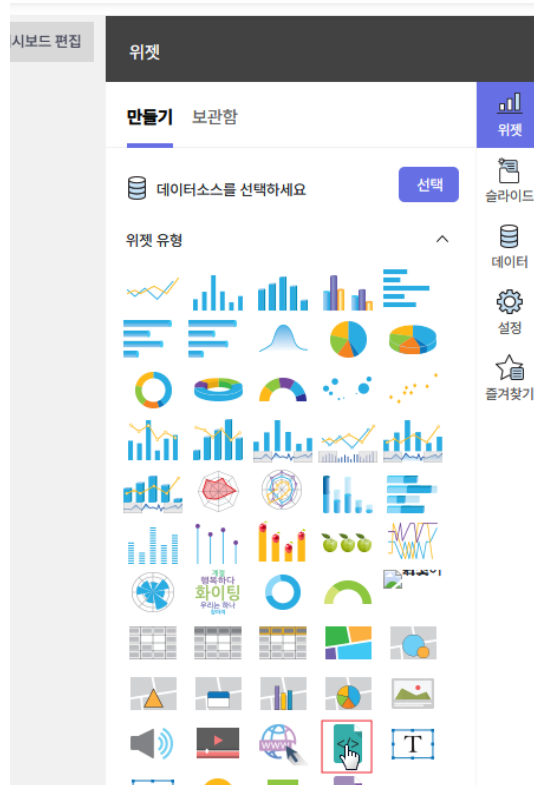
사전 요구 사항

- HTML, CSS, JavaScript 를 다뤄본 경험
- BIX5 를 이용하여 대시보드 제작한 경험
- 대시보드 개발자 권한으로 로그인

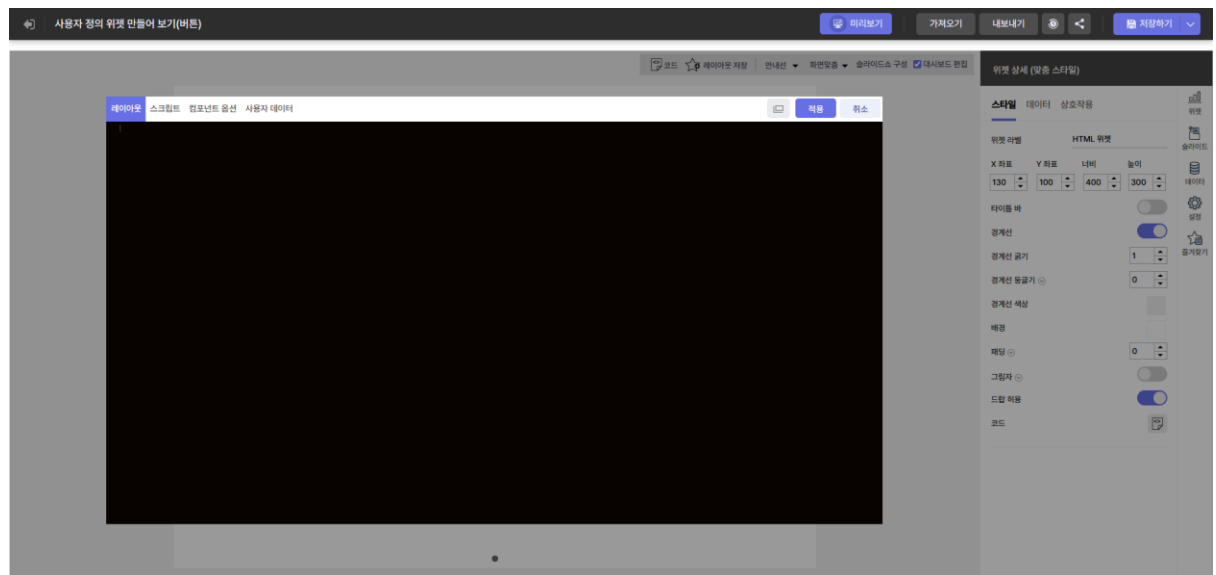
최종 결과

일자	월	요일	날씨	방류량	유입량	증설유입량	기존유입량	방류수_...	유입수_...	유입수_...	유입수_...
2021/01/05	01월	월	맑음	5,588	5,724	5,199	785	7.498	293.5	6.382	56.836
2021/01/12	01월	월	눈	6,299	6,180	5,810	903	6.51	118.5	3.819	40.728
2021/01/19	01월	월	맑음	5,624	5,568	5,214	806	6.232	295.5	5.7	54.406
2021/01/26	01월	월	비	5,314	5,368	4,926	817	6.854	232	8.172	59.63
2021/02/02	02월	월	흐림	5,615	5,599	5,427	792	6.998	135.3	4.654	48.17
2021/02/09	02월	월	맑음	5,517	5,823	5,557	687	6.527	253.5	6.06	60.932
2021/02/16	02월	월	눈	5,433	5,717	5,286	786	6.471	161.7	4.282	47.882
2021/02/23	02월	월	맑음	5,767	6,036	5,881	766	6.715	224	5.47	62.778
2021/03/02	03월	월	맑음	4,444	2,800	4,602	588	5.615	310	4.44	37.002

1. 위젯 만들기에서 HTML 위젯을 생성합니다.



2. 만든 위젯의 코드 에디터로 진입합니다.



3. 레이아웃에 HTML 을 작성합니다.

레이아웃 스크립트 컴포넌트 옵션 사용자 데이터

```
1 <input class="searchInput" type="text">
2 <button onclick="widget._searchHandler(event);">검색 </button>
```

4. 적용을 누르면 검색용 위젯이 만들어집니다.

검색

5. 검색용 위젯에 샘플 데이터 소스 → 하수처리 데이터_쿼리 데이터 셋을 바인딩 합니다. 해당 데이터셋에는 요일 파라미터 처리가 되어 있습니다.

맞춤 쿼리

맞춤 쿼리
실행결과
데이터 요약

+ DS등급별업체수

+ 제품판매량

+ 하수처리데이터

```

1 SELECT 일자, 월, 요일, 날씨, 방류량, 유입량, 증설유입량, 기존유입량, 방류수_총질소량
2 , 유입수_산소요구량, 유입수_총인량, 유입수_총질소량
3 FROM 하수처리데이터
4 WHERE 1=1
5 <#if 요일?? && 요일 != ">
6 AND 요일 = :요일
7 </#if>

```

코드
레이아웃 저장
안내선
화면맞춤
슬라이드쇼 구성
대시보드 편집

검색

스타일
데이터
상호작용

하수처리 데이터_쿼리
해제

슬라이드

집계 사용 ?

☐

데이터셋 단독 사용 ?

☐

표시 개수 ?

☐

위젯

데이터

설정

6. 다시 검색용 위젯의 코드 에디터창 → 스크립트로 이동하여 버튼 클릭 핸들러 내용을 작성합니다.

레이아웃
스크립트
컴포넌트 옵션
사용자 데이터

```

1 <script>
2 widget._searchHandler = function(event){
3     var searchKeyword = widget.querySelector(".searchInput").value;
4     var rds = widget.getBindingSourceObject();
5     rds.setParam({요일:searchKeyword});
6     rds.serviceStart();
7 }
8
9 /**
10 * 데이터 변형 사용자 정의 함수를 등록합니다.
11 * 콜백함수에 인자로 topic, payload가 전달됩니다.
12 * payload: 수신된 데이터. 해당 값은 참조된 값이기 때문에 유의하여 사용 바랍니다.
13 */
14 widget.setTransformSourceHandler(function(topic, payload){
15     return null;
16 });
17 </script>
  
```

버튼 클릭 시 input text 의 값을 조회하여 현재 위젯에 바인딩 된 서비스객체를 얻어 파라미터를 설정하고 serviceStart 를 호출하여 데이터를 다시 조회 합니다. 데이터를 조회할 때 기존에 있던 레이아웃을 지우고 새로 밀어 넣으므로 기존 레이아웃을 유지하고자 setTransformSourceHandler 에서 return null 처리를 합니다.

7. 검색 결과를 확인하기 위하여 하단에 그리드 위젯을 생성하고 검색용 위젯과 같은 데이터셋인 하수처리 데이터_쿼리 데이터 셋을 그리드 위젯에 바인딩합니다.

레이아웃
스크립트
컴포넌트 옵션
사용자 데이터

일자	월	요일	날씨	방류량	유입량	중성유입량	기존유입량	방류수...	유입수...	유입수...	유입수...
2021/01/01	01월	목	맑음	5,554	6,087	4,936	788	8,371	216	4,703	56,011
2021/01/02	01월	금	맑음	5,815	5,935	5,166	776	9,183	234.7	5,203	58,919
2021/01/03	01월	토	맑음	5,742	6,182	5,001	797	8,734	263	5,849	35,107
2021/01/04	01월	일	맑음	5,307			798	6,622	236	5,731	67,636
2021/01/05	01월	월	맑음	5,588	5,724	5,199	785	7,498	293.5	6,382	56,836
2021/01/06	01월	화	눈	5,359	6,037	4,938	791	6,464	204.5	5,345	55,991
2021/01/07	01월	수	맑음	5,144	5,420	4,585	827	8,495	246.5	4,935	50,879
2021/01/08	01월	목	맑음	5,931	5,909	5,329	790	6,793	177.3	4,271	47,755
2021.01.09 ... 01월	금	맑음		5,932	6,155	5,358	792	8,268	136	10,943	58,724

레이아웃
스크립트
컴포넌트 옵션
사용자 데이터

하수처리 데이터_쿼리
삭제

입력

날짜

월

요일

날씨

방류량

유입량

중성유입량

기존유입량

방류수_총합소량

유입수_총합소량

출력

날짜

월

요일

날씨

방류량

유입량

중성유입량

기존유입량

방류수_총합소량

유입수_총합소량

8. 검색창에 요일 중 하나를 입력하고 버튼을 클릭하여 검색합니다.

고려_나만의 검색용 위젯 만들어 보기

위젯 만들기

스타일 데이터 상호작용

위젯 라벨

스타일 - style1

X 좌표 Y 좌표 너비 높이

13 56 999 300

타이틀 바

검색선

검색선 통글자

배경

패딩

그림자

드롭 차움

코드

헤더 색상

헤더 텍스트

헤더 텍스트 색상

행 색상

행 텍스트

선택 색상

마우스 오버 색상

행 높이 설정

일자	월	요일	날씨	방류량	유입량	중설유입량	기존유입량	방류수...	유입수...	유입수...	유입수...
2021/01/01	01월	목	맑음	5,554	6,087	4,936	788	8,371	216	4,703	56,011
2021/01/08	01월	목	맑음	5,931	5,909	5,329	790	6,793	177.3	4,271	47,755
2021/01/15	01월	목	흐림	5,536	5,836	5,091	817	6,221	237	4,697	49,498
2021/01/22	01월	목	흐림	5,695	5,092	4,790	794	6,075	165.3	4,114	47,334
2021/01/29	01월	목	맑음	5,185	5,508	4,790	814	6,28	224.6	4,851	47,201
2021/02/05	02월	목	맑음	5,353	5,770	5,226	770	7,226	238	6,615	71,864
2021/02/12	02월	목	흐림	5,513	5,600	5,092	905	5,072	198.4	4,637	47,943
2021/02/19	02월	목	맑음	5,402	6,390	5,504	792	7,327	130.5	4,012	47,339
2021/02/26	02월	목	흐림	5,161	5,983	5,161	736	7,585	174	3,71	41,171

9. 빈 값을 검색하면 전체 목록이 조회되는 것을 확인하실 수 있습니다

고려_나만의 검색용 위젯 만들어 보기

위젯 만들기

스타일 데이터 상호작용

위젯 라벨

스타일 - style1

X 좌표 Y 좌표 너비 높이

13 56 999 300

타이틀 바

검색선

검색선 통글자

배경

패딩

그림자

드롭 차움

코드

헤더 색상

헤더 텍스트

헤더 텍스트 색상

행 색상

행 텍스트

선택 색상

마우스 오버 색상

행 높이 설정

일자	월	요일	날씨	방류량	유입량	중설유입량	기존유입량	방류수...	유입수...	유입수...	유입수...
2021/01/01	01월	목	맑음	5,554	6,087	4,936	788	8,371	216	4,703	56,011
2021/01/02	01월	금	맑음	5,815	5,935	5,166	776	9,183	234.7	5,203	58,919
2021/01/03	01월	토	맑음	5,742	6,182	5,001	797	8,734	263	5,849	35,107
2021/01/04	01월	일	맑음	5,507	5,724	5,199	785	7,498	293.5	6,382	56,836
2021/01/05	01월	월	맑음	5,588	6,037	4,938	791	6,464	204.5	5,345	55,991
2021/01/06	01월	화	눈	5,359	5,420	4,585	827	8,495	246.5	4,935	50,879
2021/01/07	01월	수	맑음	5,931	5,909	5,329	790	6,793	177.3	4,271	47,755
2021/01/08	01월	목	맑음	5,931	5,909	5,329	790	6,793	177.3	4,271	47,755
2021/01/09	01월	금	맑음	5,931	5,909	5,329	790	6,793	177.3	4,271	47,755

1.3.1. 소스

레이아웃

```
<style>
  #{{id}} button{
    border:none;
    background-color:#6670e4;
    color:#ffffff;
    border-radius:7px;
    padding: 4px 10px;
  }
</style>
<input class="searchInput" type="text">
<button onclick="widget._searchHandler(event);">검색</button>
```

스크립트

```
<script>
  widget.componentReadyHandler = function(componentId){

  }

  widget._searchHandler = function(event){
    var searchKeyword = widget.querySelector(".searchInput").value;
    var rds = widget.getBindingSourceObject();
    rds.setParam({요일:searchKeyword});
    rds.serviceStart();
  }

  widget.setTransformSourceHandler(function(topic, payload){
    return null;
  });

  widget.componentRemoveHandler = function(){

  }
</script>
```

1.4. 위젯간 이벤트 송/수신 예제 만들어 보기

학습 목표

- BIX5 사용자 정의 이벤트를 만들어 보내고 받을 수 있다.

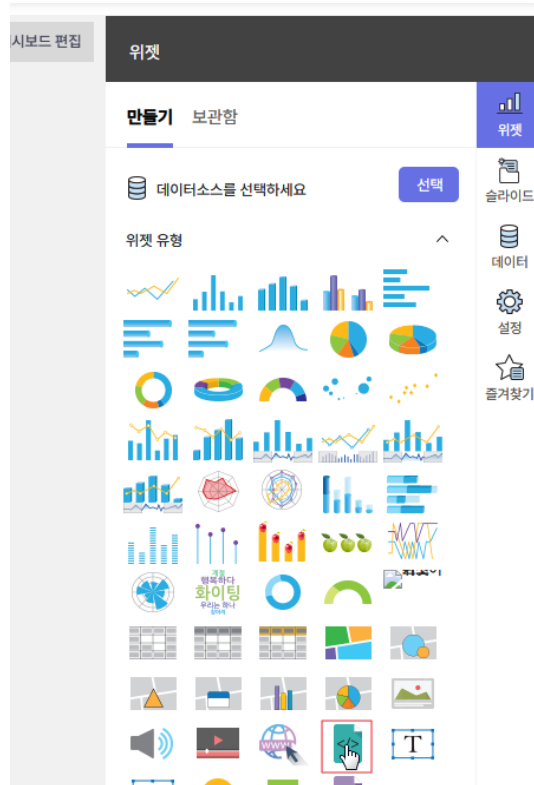
사전 요구 사항

- HTML, CSS, JavaScript 를 다뤄본 경험
- BIX5 를 이용하여 대시보드 제작한 경험
- 대시보드 개발자 권한으로 로그인

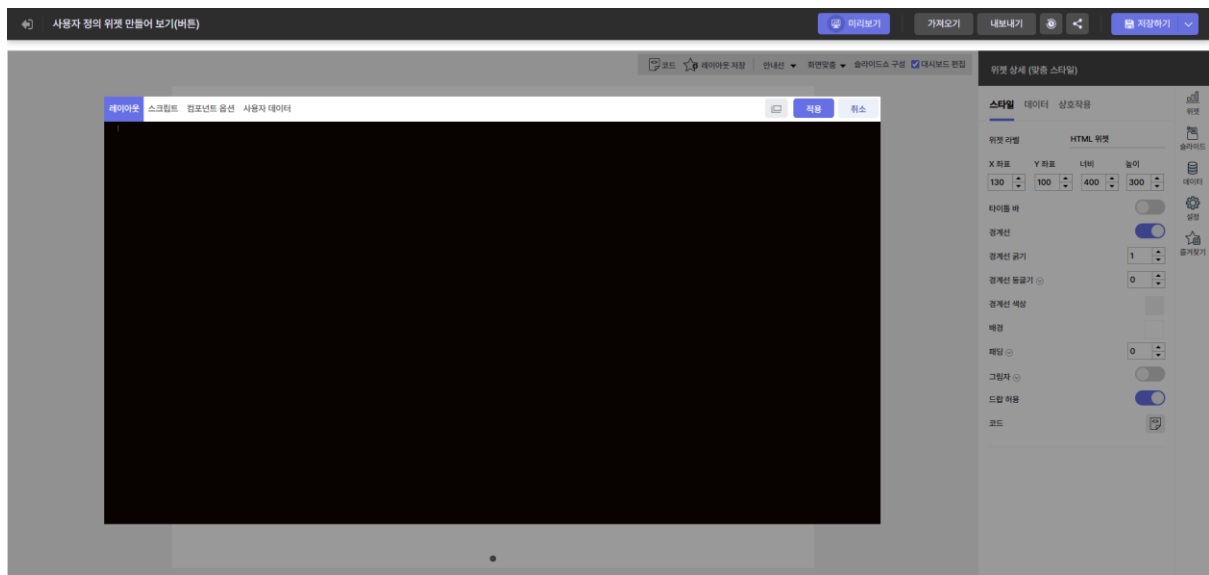
최종 결과

일자	월	요일	날씨	방류량	유입량	증설유입량	기존유입량	방류수_...	유입수_...	유입수_...	유입수_...
2021/01/05	01월	월	맑음	5,588	5,724	5,199	785	7.498	293.5	6.382	56.836
2021/01/12	01월	월	눈	6,299	6,180	5,810	903	6.51	118.5	3.819	40.728
2021/01/19	01월	월	맑음	5,624	5,568	5,214	806	6.232	295.5	5.7	54.406
2021/01/26	01월	월	비	5,314	5,368	4,926	817	6.854	232	8.172	59.63
2021/02/02	02월	월	흐림	5,615	5,599	5,427	792	6.998	135.3	4.654	48.17
2021/02/09	02월	월	맑음	5,517	5,823	5,557	687	6.527	253.5	6.06	60.932
2021/02/16	02월	월	눈	5,433	5,717	5,286	786	6.471	161.7	4.282	47.882
2021/02/23	02월	월	맑음	5,767	6,036	5,881	766	6.715	224	5.47	62.778
2021/03/02	03월	월	맑음	4,444	2,800	4,602	588	5.615	310	4.44	37.002

1. 위젯 만들기에서 HTML 위젯을 생성합니다.



2. 만든 위젯의 코드 에디터로 진입합니다.



3. 레이아웃에 HTML 을 작성합니다.

레이아웃 스크립트 컴포넌트 옵션 사용자 데이터

```
1 <input class="searchInput" type="text">
2 <button onclick="widget._searchHandler(event);">검색 </button>
```

4. 적용을 누르면 검색용 위젯이 만들어집니다.

검색

5. 검색용 위젯에 샘플 데이터 소스 → 하수처리 데이터_쿼리 데이터 셋을 바인딩 합니다. 해당 데이터셋에는 요일 파라미터 처리가 되어 있습니다.

맞춤 쿼리

맞춤 쿼리
실행결과
데이터 요약

+ DS등급별업체수

+ 제품판매량

+ 하수처리데이터

```

1 SELECT 일자, 월, 요일, 날씨, 방류량, 유입량, 증설유입량, 기존유입량, 방류수_총질소량
2 , 유입수_산소요구량, 유입수_총인량, 유입수_총질소량
3 FROM 하수처리데이터
4 WHERE 1=1
5 <#if 요일?? && 요일 != ">
6 AND 요일 = :요일
7 </#if>

```

코드
레이아웃 저장
안내선
화면맞춤
슬라이드쇼 구성
대시보드 편집

검색

스타일
데이터
상호작용

하수처리 데이터_쿼리
해제

집계 사용 ?

데이터셋 단독 사용 ?

표시 개수 ?

위젯

슬라이드

데이터

설정

6. 다시 검색용 위젯의 코드 에디터 창 → 스크립트로 이동하여 버튼 클릭 핸들러 내용을 작성합니다.

레이아웃 **스크립트** 컴포넌트 옵션 사용자 데이터

```

1 <script>
2   widget._searchHandler = function(event){
3     var searchKeyword = widget.querySelector(".searchInput").value;
4
5     var newEvent = BIX5.dashboard.createEvent("search");
6     newEvent.detail = {
7       searchKeyword: searchKeyword
8     };
9
10    var slide = widget.getSlide();
11    slide.dispatchEvent(newEvent);
12
13  }
14 </script>

```

```

<script>
  widget._searchHandler = function(event){
    var searchKeyword = widget.querySelector(".searchInput").value;

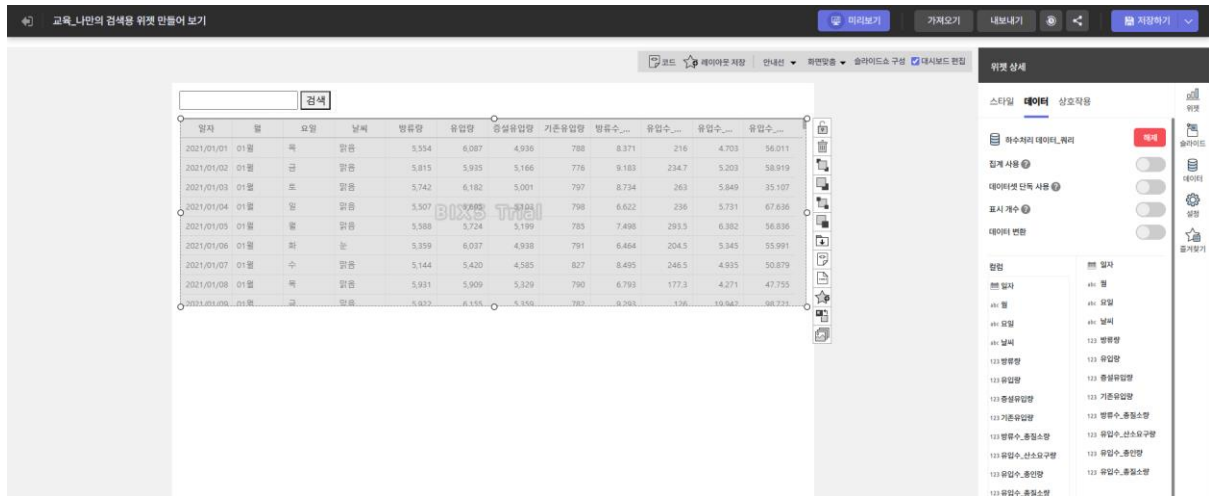
    var newEvent = BIX5.dashboard.createEvent("search");
    newEvent.detail = {
      searchKeyword : searchKeyword
    };

    var slide = widget.getSlide();
    slide.dispatchEvent(newEvent);
  }
</script>

```

버튼 클릭 시 input text 의 값을 조회하고 사용자 이벤트를 생성하여 슬라이드 객체에 보냅니다. 이벤트명은 사용자 임의로 지정하면 됩니다.

7. 검색 결과를 확인하기 위하여 하단에 그리드 위젯을 생성하고 검색용 위젯과 같은 데이터셋인 하수처리 데이터_쿼리 데이터 셋을 그리드 위젯에 바인딩합니다.



8. 그리드 위젯의 코드 에디터 창 → 스크립트로 이동하여 이벤트 리스너를 작성합니다.



```
<script>
  widget.componentReadyHandler = function(componentId){
    var slide = widget.getSlide();
    slide.addEventListener("search", widget._searchHandler);
  }

  widget._searchHandler = function(event){
    var rds = widget.getBindingSourceObject();
    rds.setParam({요일:event.detail.searchKeyword});
    rds.serviceStart();
  }
</script>
```

위젯이 사용할 준비가 된 `componentReadyHandler` 에서 슬라이드 객체를 얻어 이벤트 리스너를 등록합니다. `search` 이벤트는 이전 검색용 위젯에서 버튼을 클릭하면 송출하는 이벤트 명입니다. 리스너를 등록하고 이벤트를 받았을 때 해야할 일을 작성합니다. 예제에서는 현재 위젯(그리드)에 바인딩 된 데이터서비스를 얻어 이벤트를 통해 받은 값을 파라미터에 설정하고 서비스를 시작하여 데이터를 다시 가지고 오는 예제입니다.

9. 검색창에 요일 중 하나를 입력하고 버튼을 클릭하여 검색합니다.

교육_나만의 검색용 위젯 만들어 보기

위젯 보기 | 가져오기 | 내보내기 | 저장하기

코드 | 레이아웃 저장 | 연내선 | 화면모음 | 슬라이드쇼 구성 | 대시보드 편집

위젯 상세

스타일 | 데이터 | 상호작용

위젯 라벨 | 그리드 - style1

X 좌표 | Y 좌표 | 너비 | 높이

13 | 56 | 999 | 300

타이틀 바

검색선

검색선 통글기

배경

패딩

그림자

드랍 허용

코드

헤더 색상

헤더 텍스트

헤더 텍스트 색상

행 색상

행 텍스트

선택 색상

마우스 오버 색상

행 높이 설정

일자	월	요일	날씨	방류량	유입량	중설유입량	기존유입량	방류수...	유입수...	유입수...	유입수...
2021/01/01	01월	목	맑음	5,554	6,087	4,936	788	8.371	216	4,703	56,011
2021/01/08	01월	목	맑음	5,931	5,909	5,329	790	6.793	177.3	4,271	47,755
2021/01/15	01월	목	흐림	5,536	5,836	5,091	817	6.221	237	4,697	49,498
2021/01/22	01월	목	흐림	5,695	5,093	5,254	794	6.875	165.3	4,114	47,334
2021/01/29	01월	목	맑음	5,185	5,508	4,780	814	6.28	224.6	4,851	47,201
2021/02/05	02월	목	맑음	5,353	5,770	5,226	770	7.226	238	6,615	71,864
2021/02/12	02월	목	흐림	5,513	5,600	5,092	905	5.072	198.4	4,637	47,943
2021/02/19	02월	목	맑음	5,402	6,390	5,504	792	7.327	130.5	4,012	47,339
2021/02/26	02월	목	흐림	5,163	5,983	5,163	736	7.585	174	3,71	41,171

10. 빈 값을 검색하면 전체 목록이 조회되는 것을 확인하실 수 있습니다

교육_나만의 검색용 위젯 만들어 보기

위젯 보기 | 가져오기 | 내보내기 | 저장하기

코드 | 레이아웃 저장 | 연내선 | 화면모음 | 슬라이드쇼 구성 | 대시보드 편집

위젯 상세

스타일 | 데이터 | 상호작용

위젯 라벨 | 그리드 - style1

X 좌표 | Y 좌표 | 너비 | 높이

13 | 56 | 999 | 300

타이틀 바

검색선

검색선 통글기

배경

패딩

그림자

드랍 허용

코드

헤더 색상

헤더 텍스트

헤더 텍스트 색상

행 색상

행 텍스트

선택 색상

마우스 오버 색상

행 높이 설정

일자	월	요일	날씨	방류량	유입량	중설유입량	기존유입량	방류수...	유입수...	유입수...	유입수...
2021/01/01	01월	목	맑음	5,554	6,087	4,936	788	8.371	216	4,703	56,011
2021/01/02	01월	금	맑음	5,815	5,935	5,166	776	9.183	234.7	5,203	58,919
2021/01/03	01월	토	맑음	5,742	6,182	5,001	797	8.734	263	5,849	35,107
2021/01/04	01월	일	맑음	5,507	5,595	5,034	798	6.622	236	5,731	67,636
2021/01/05	01월	월	맑음	5,588	5,724	5,199	785	7.498	293.5	6,382	56,836
2021/01/06	01월	화	눈	5,359	6,037	4,938	791	6.464	204.5	5,345	55,991
2021/01/07	01월	수	맑음	5,144	5,420	4,585	827	8.495	246.5	4,935	50,879
2021/01/08	01월	목	맑음	5,931	5,909	5,329	790	6.793	177.3	4,271	47,755
2021/01/09	01월	금	맑음	5,927	6,155	5,159	782	9.791	136	10,843	98,731

1.4.1. 소스

레이아웃

```
<style>
  #{{id}} button{
    border:none;
    background-color:#6670e4;
    color:#ffffff;
    border-radius:7px;
    padding: 4px 10px;
  }
</style>
<input class="searchInput" type="text">
<button onclick="widget._searchHandler(event);">검색</button>
```

스크립트(검색용 위젯)

```
<script>
  widget.componentReadyHandler = function(componentId){

  }

  widget._searchHandler = function(event){
    var searchKeyword = widget.querySelector(".searchInput").value;

    var newEvent = BIX5.dashboard.createEvent("search");
    newEvent.detail = {
      searchKeyword: searchKeyword
    };

    var slide = widget.getSlide();
    slide.dispatchEvent(newEvent);
  }

  widget.setTransformSourceHandler(function(topic, payload){
    return null;
  });

  widget.componentRemoveHandler = function(){

  }
</script>
```

스크립트(그리드 위젯)

```
<script>
  widget.componentReadyHandler = function(componentId){
    var slide = widget.getSlide();
    slide.addEventListener("search", widget._searchHandler);
  }

  widget._searchHandler = function(event){
    var searchKeyword = event.detail.searchKeyword;

    var rds = widget.getBindingSourceObject();
    rds.setParam({요일:searchKeyword});
    rds.serviceStart();
  }

  widget.setTransformSourceHandler(function(topic, payload){
    return payload;
  });

  widget.componentRemoveHandler = function(){

  }
</script>
```

2. 콤포넌트 위젯 확장하기

BIX5 는 차트, 그리드, 맵차트 위젯을 기본으로 지원하고 있습니다. 이 위젯들로 데이터를 효과적으로 시각화 할 수 있습니다. 또한, 기본 제공 기능 외에도 사용자 정의 함수를 활용해 특정 시점의 이벤트를 얻어 위젯을 필요에 맞게 커스터마이징 할 수 있습니다.

2.1. 차트 위젯 확장하기 I

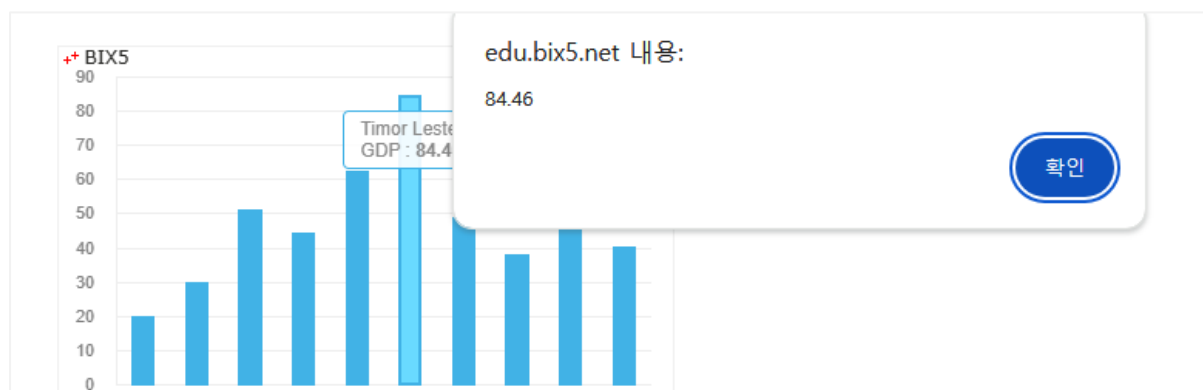
학습 목표

- 컬럼 2D 차트 아이템 클릭 이벤트를 등록하여 사용자 함수를 작성할 수 있다.

사전 요구 사항

- HTML, CSS, JavaScript 를 다뤄본 경험
- BIX5 를 이용하여 대시보드 제작한 경험
- 대시보드 개발자 권한으로 로그인
- 데이터셋 : 샘플 데이터 소스 - 하수처리 데이터
- 시나리오 : 월별 방류량 그래프 출력 후 차트 아이템을 클릭하면 값을 대화 상자로 출력한다.

최종 결과



1. 위젯 만들기에서 컬럼 차트를 생성하고 코드 에디터 → 스크립트로 이동합니다.

The screenshot shows the 'Widget Maker' interface in BIX 5. The main area displays a column chart titled 'GDP' with data for five countries: South Sudan, Sierra Leone, Paraguay, Iraq, and Gambia. The y-axis represents GDP values ranging from 0 to 90. The right sidebar shows the 'Widget Maker' menu with various chart types and a 'Script' tab selected.

Country	GDP (approx.)
South Sudan	20
Sierra Leone	45
Paraguay	60
Iraq	85
Gambia	65

The screenshot shows the 'Script Editor' interface in BIX 5. The main area displays the code for the widget, which is a JavaScript script. The code defines a widget with a title 'GDP' and a description 'GDP of selected countries'. It uses the 'column' chart type and sets the data source to 'GDP'.

```

<script>
  /**
   * component를 사용할 준비가 되면 호출합니다.
   *
   * widget.componentReadyHandler = function(componentId){
   *   if(!document.getElementById('GDP')){
   *     document.getElementById(componentId).setAttribute("json");
   *     document.getElementById(componentId).setAttribute("GDP", "DataSources/Colum-Column.json");
   *   }
   * }
   *
   * 데이터 변환 사용자 정의 함수를 등록합니다.
   * 출력값에 인자로 topic, payload가 전달됩니다.
   * payload: 수신된 데이터. 해당 값은 참조된 값이기 때문에 유지하며 사용 가능합니다.
   *
   * widget.transformSourceHandler=function(topic, payload){
   *   return payload;
   * };
   *
   * component가 실행되기 전 호출합니다.
   *
   * widget.componentPreHandler = function(){
   *   //
   * }
   */
</script>
  
```

The right sidebar shows the 'Widget Settings' panel with the 'Data' tab selected. It displays the 'Country' field with a value of 'Country' and the 'GDP' field with a value of 'GDP'.

2. 아이템 클릭 이벤트를 받기 위해 위젯 객체에 itemClickJsFunction 을 등록합니다.

레이아웃 스크립트 컴포넌트 옵션 사용자 데이터

```
1 <script>
2 /**
3  * component를 사용할 준비가 되면 호출합니다
4  */
5 widget.componentReadyHandler = function(componentId){
6   if(!widget.isBindingSource()){
7     document.getElementById(componentId).setDataType("json");
8     document.getElementById(componentId).setDataURL(BIX5.getAssetsPath() + "Datasources/Column-Column.json");
9   }
10 }
11
12 widget.itemClickJsFunction = function(seriesId, displayText, index, data, values) {
13
14 }
15
16 /**
17  * 데이터 변형 사용자 정의 함수를 등록합니다.
18  * 콜백함수에 인자로 topic, payload가 전달됩니다.
19  * payload: 수신된 데이터. 해당 값은 참조된 값이기 때문에 유의하여 사용 바랍니다.
20  */
21
22 widget.setTransformSourceHandler(function(topic, payload){
```

위젯 객체에 widget.사용자정의함수를 등록하면 해당 이벤트때 등록된 함수를 호출하는 기능을 일부 제공하고 있습니다. 사용하고자 하는 사용자정의함수의 자세한 사용법은 차트 컴포넌트의 API 에서 확인하실 수 있습니다.

<https://demo.bix5.net/SDK/Docs/Dashboard/API/ChartWidget.html#event:itemClickJsFunction>

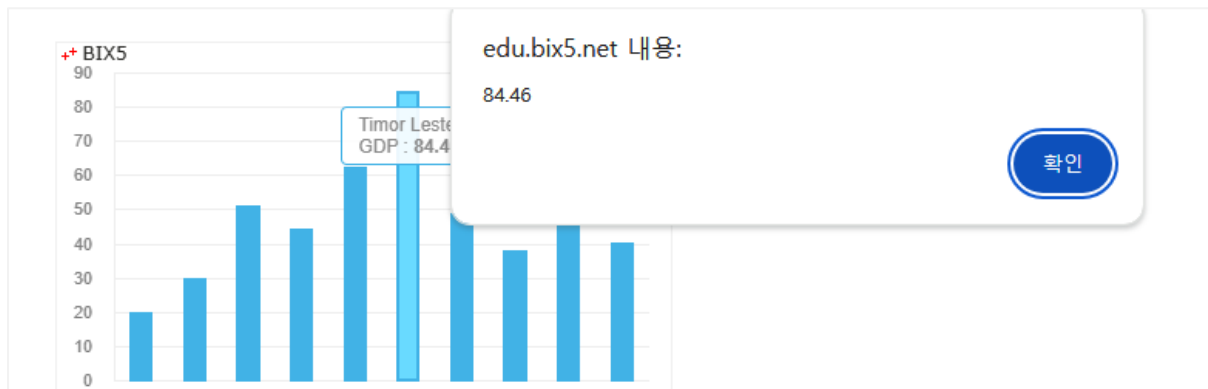
3. 차트 아이템을 클릭하면 값을 대화상자로 출력하도록 작성합니다.

레이아웃 스크립트 컴포넌트 옵션 사용자 데이터

```
1 <script>
2 /**
3  * component를 사용할 준비가 되면 호출합니다
4  */
5 widget.componentReadyHandler = function(componentId){
6   if(!widget.isBindingSource()){
7     document.getElementById(componentId).setDataType("json");
8     document.getElementById(componentId).setDataURL(BIX5.getAssetsPath() + "Datasources/Column-Column.json");
9   }
10 }
11
12 widget.itemClickJsFunction = function(seriesId, displayText, index, data, values) {
13   alert(values[1]);
14 }
15
16 /**
17  * 데이터 변형 사용자 정의 함수를 등록합니다.
18  * 콜백함수에 인자로 topic, payload가 전달됩니다.
19  * payload: 수신된 데이터. 해당 값은 참조된 값이기 때문에 유의하여 사용 바랍니다.
20  */
21
22 widget.setTransformSourceHandler(function(topic, payload){
```

```
widget.itemClickJsFunction = function(seriesId, displayText, index, data,
values) {
  alert(values[1]);
}
```

4. 적용 후 미리보기에서 차트 시리즈(막대)를 클릭하면 값을 대화상자로 출력하는 것을 확인하실 수 있습니다.



2.1.1. 소스

스크립트

```
<script>
    widget.componentReadyHandler = function(componentId){
        if(!widget.isBindingSource()){
            document.getElementById(componentId).setDataType("json");
            document.getElementById(componentId).setDataURL(BIX5.getAssetsPath
() + "Datasources/Column-Column.json");
        }
    }

    widget.itemClickJsFunction = function(seriesId, displayText, index, data,
values) {
        alert(values[1]);
    }

    widget.componentRemoveHandler = function(){
    }
</script>
```

2.2. 차트 위젯 확장하기 II

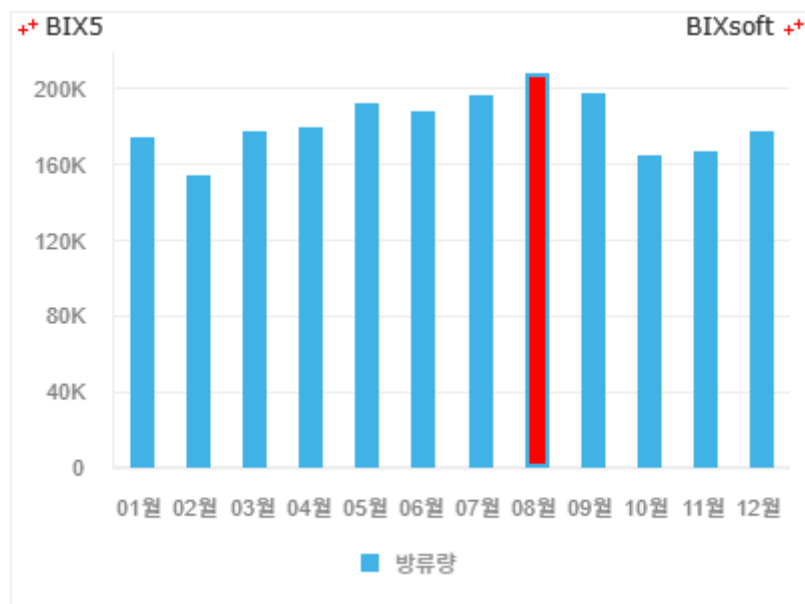
학습 목표

- 컬럼 2D 차트 컴포넌트 사용자 정의 함수를 이용하여 시리즈 색상을 변경할 수 있다.

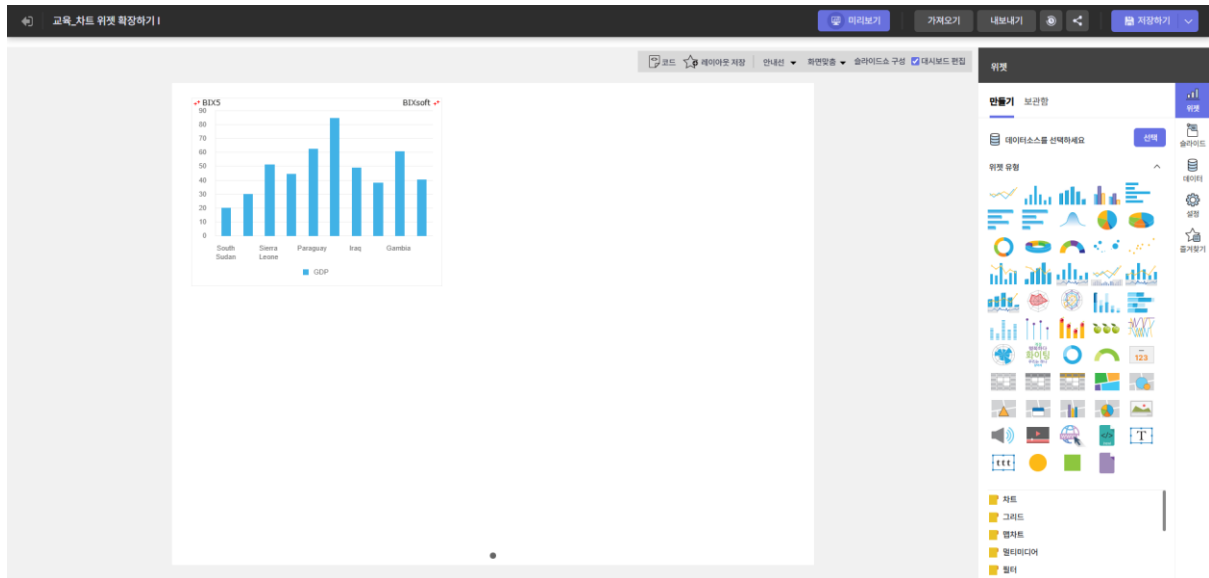
사전 요구 사항

- HTML, CSS, JavaScript 를 다뤄본 경험
- BIX5 를 이용하여 대시보드 제작한 경험
- 대시보드 개발자 권한으로 로그인
- 데이터셋: 샘플 데이터 소스 - 하수처리 데이터
- 시나리오 : 월별 방류량 그래프 작성 후 방류량이 200,000 이상이면 빨간색으로 변경한다.

최종 결과



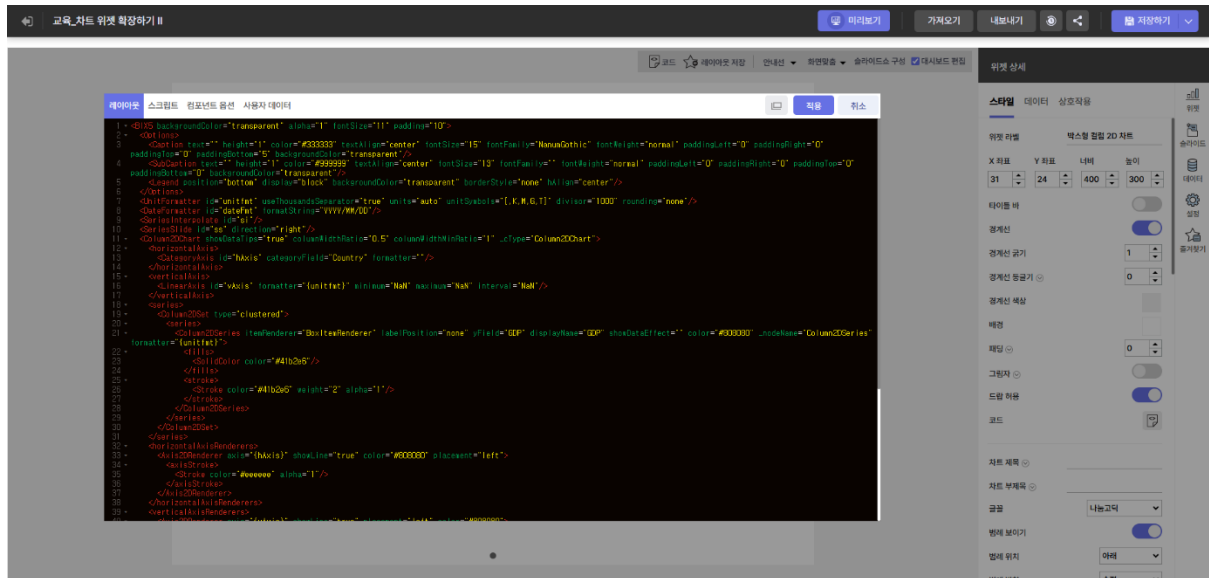
1. 위젯 만들기에서 컬럼 차트를 생성합니다.



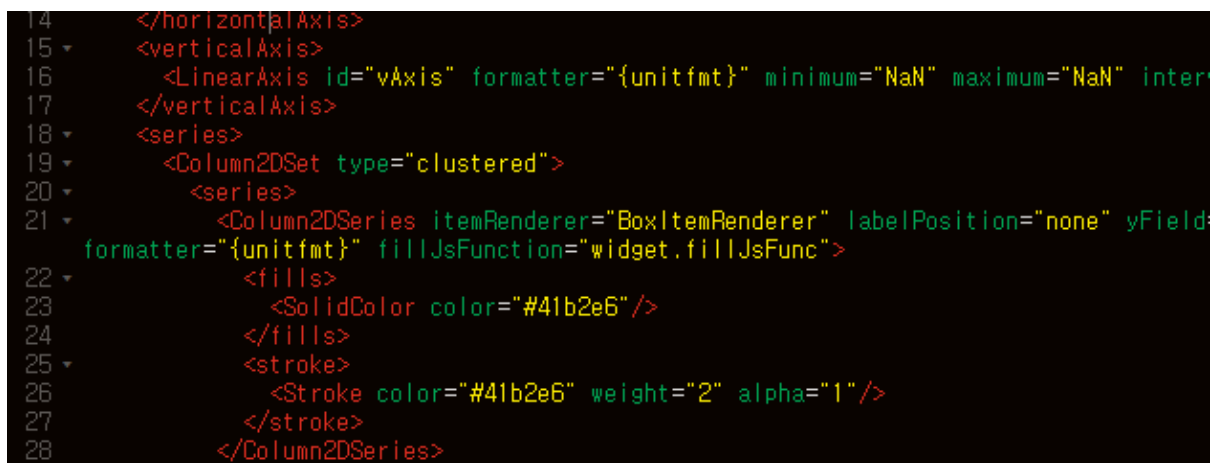
2. 차트 위젯에 샘플 데이터 → 하수처리 데이터를 바인딩 합니다. 그리고 위젯 상세 → 데이터로 이동하여 집계 사용을 켜고 수평축 필드에 '월'필드, 수직축 필드에 '방류량'필드를 설정합니다.



3. 차트 위젯의 코드 에디터 → 레이아웃으로 이동합니다.



4. 조건에 따라 막대의 색상을 변경하기 Column2DSeries 에 fillJsFunction 을 등록합니다.



fillJsFunction="widget.fillJsFunc"

함수를 등록할 때 'widget'은 예약어로 위젯 객체를 가리킵니다. 현재 예시에서는 Column2DSeries 의 fillJsFunction 에 widget 객체의 fillJsFunc 함수를 등록한 것입니다.

5. 스크립트 편집 창에서 앞서 2 에서 등록한 함수를 생성하고 작성합니다.
 월별 방류량이 200,000 이상이면 빨간색을 리턴 하여 조건에 따라 시리즈의 색상을 변경하도록 합니다. 색상을 리턴 하지 않으면 검정색이 기본으로 나오므로 조건을 잘 따져 색상을 리턴 합니다.

레이아웃 **스크립트** 컴포넌트 옵션 사용자 데이터

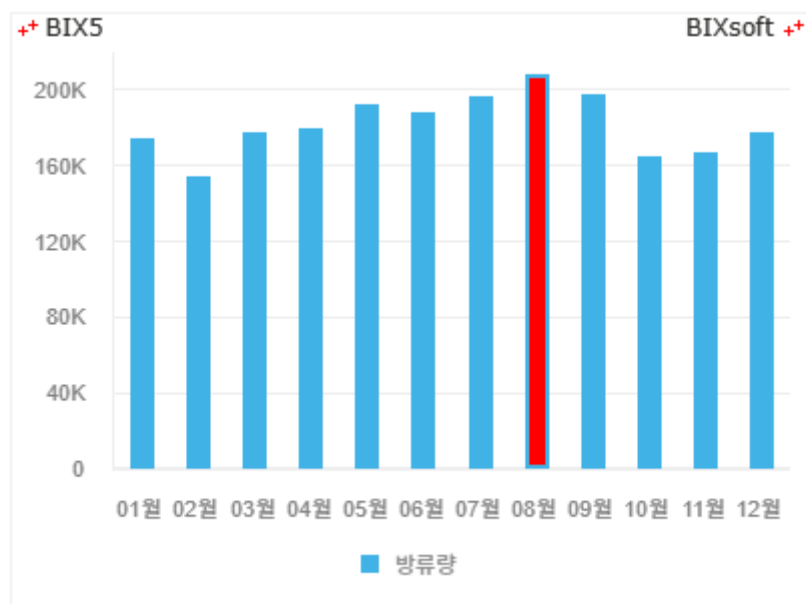
```
1 <script>
2 widget.fillJsFunc = function(seriesId, index, data, values) {
3   if(values[1] >= 200000)
4     return "#ff0000"; //값이 20만 이상이면 빨간색을 리턴
5   else
6     return {color:"#41b2e6", alpha:1}; //JSON 형식으로 리턴할 수도 있습니다.
7 }
```

```
widget.fillJsFunc = function(seriesId, index, data, values) {
  if(values[1] >= 200000)
    return "#ff0000";
  else
    return {color:"#41b2e6", alpha:1};
}
```

사용하고자 하는 사용자정의함수의 자세한 사용법은 차트 컴포넌트의 API 에서 확인하실 수 있습니다.

<https://demo.bix5.net/SDK/Docs/Chart/API/Column2DSeries.html#fillJsFunction>

6. 적용을 눌러 결과를 확인합니다.



2.2.1. 소스

스크립트

```
<script>
  widget.componentReadyHandler = function(componentId){
    if(!widget.isBindingSource()){
      document.getElementById(componentId).setDataType("json");
      document.getElementById(componentId).setDataURL(BIX5.getAssetsPath
() + "Datasources/Column-Column.json");
    }
  }

  widget.fillJsFunc = function(seriesId, index, data, values) {
    if(values[1] >= 200000)
      return "#ff0000";
    else
      return {color:"#41b2e6", alpha:1};
  }

  widget.componentRemoveHandler = function(){
  }
</script>
```

2.3. 그리드 위젯 확장하기 I

학습 목표

- 그리드 아이템 클릭(셀 클릭) 이벤트를 등록하여 사용자 함수를 작성할 수 있다.

사전 요구 사항

- HTML, CSS, JavaScript 를 다뤄본 경험
- BIX5 를 이용하여 대시보드 제작한 경험
- 대시보드 개발자 권한으로 로그인
- 데이터셋: 샘플 데이터 소스 - 하수처리 데이터
- 시나리오 : 하수처리 데이터에서 일자, 월, 요일, 날씨, 방류량, 유입량, 증설유입량, 기존유입량 출력 후 아이템을 클릭하면 값을 대화 상자로 출력한다.

최종 결과

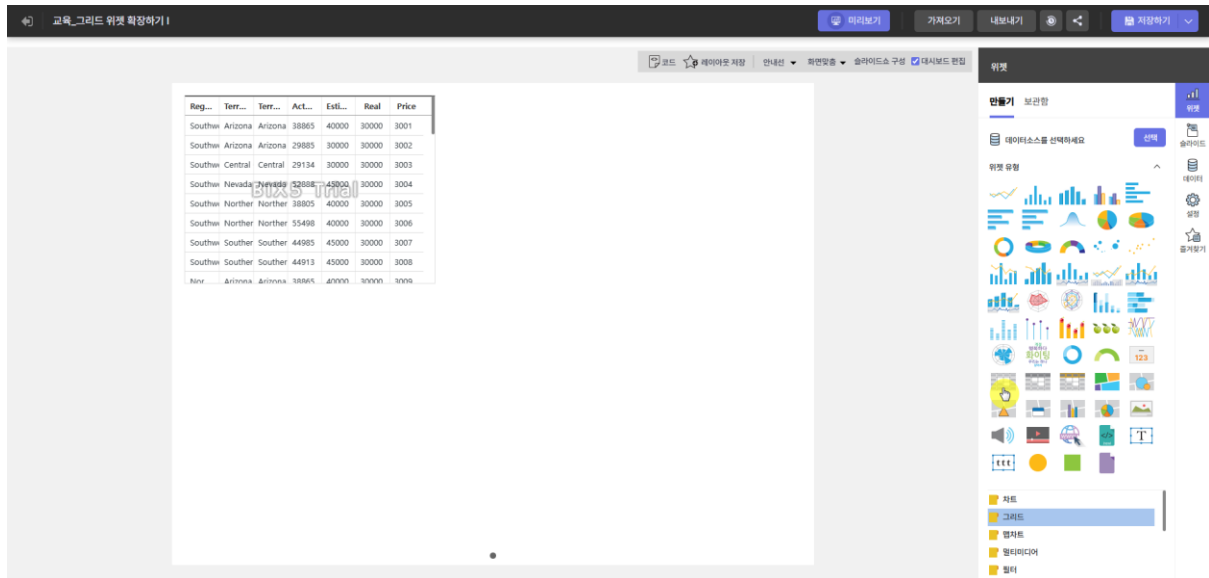
edu.bix5.net 내용:

2021/01/01

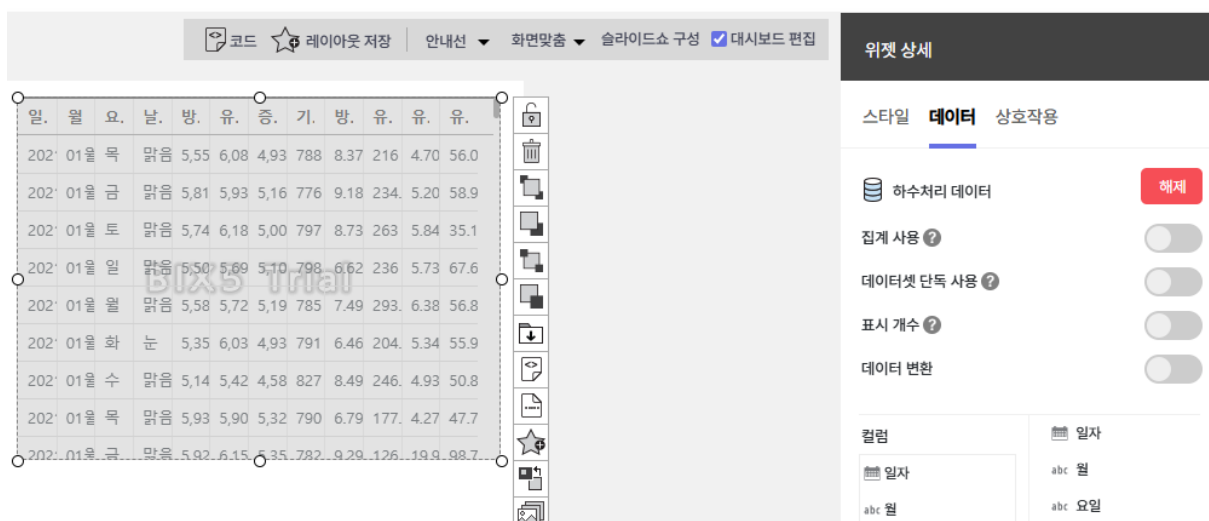
확인

일자	월	요일	날씨	방류량	유입량	증설유입량	기존유입량	유입수_...	유입수_...	유입수_...	
2021/01/01	01월	목	맑음	5,507	5,695	5,103	798	6.622	216	4.703	56.011
2021/01/02	01월	금	맑음	5,588	5,724	5,199	785	7.498	234.7	5.203	58.919
2021/01/03	01월	토	맑음	5,359	6,037	4,938	791	6.464	263	5.849	35.107
2021/01/04	01월	일	맑음	5,144	5,420	4,585	827	8.495	236	5.731	67.636
2021/01/05	01월	월	맑음	5,931	5,909	5,329	790	6.793	293.5	6.382	56.836
2021/01/06	01월	화	눈	5,922	6,155	5,359	782	9.293	204.5	5.345	55.991
2021/01/07	01월	수	맑음	5,931	5,909	5,329	790	6.793	246.5	4.935	50.879
2021/01/08	01월	목	맑음	5,931	5,909	5,329	790	6.793	177.3	4.271	47.755
2021/01/09	01월	금	맑음	5,922	6,155	5,359	782	9.293	126	19.942	98.721

1. 위젯 만들기에서 그리드를 생성합니다.



2. 그리드 위젯에 샘플 데이터 → 하수처리 데이터를 바인딩 합니다.



3. 그리드 위젯의 코드 에디터 → 스크립트로 이동합니다.

레이아웃 스크립트 컴포넌트 옵션 사용자 데이터

```

1 <script>
2 /**
3  * component를 사용할 준비가 되면 호출합니다
4  */
5 widget.componentReadyHandler = function(componentId){
6   if(!widget.isBindingSource()){
7     document.getElementById(componentId).setDataType("json");
8     document.getElementById(componentId).setDataURLEx(BIX5.getAssetsPath() + "Datasources/data2.json");
9   }
10 }
11

```

4. 아이템 클릭 이벤트 리스너를 등록합니다. 데이터를 얻어 일자를 대화상자에 출력합니다.

레이아웃 스크립트 컴포넌트 옵션 사용자 데이터

```

1 <script>
2 widget.itemClickJsFunction = function(event){
3   var data = event.itemRenderer.getData();
4   alert(data["일자"]);
5 }

```

5. 대시보드 미리보기에서 임의의 row 를 클릭합니다. 클릭하면 alert 창에 선택한 row 의 일자를 출력합니다.

일자	월	요일		edu.bix5.net 내용:					유입수_...	유입수_...	유입수_...
2021/01/01	01월	목	맑음	2021/01/01					216	4.703	56.011
2021/01/02	01월	금	맑음						234.7	5.203	58.919
2021/01/03	01월	토	맑음						263	5.849	35.107
2021/01/04	01월	일	맑음	5,507	5,685	5,103	798	6.622	236	5.731	67.636
2021/01/05	01월	월	맑음	5,588	5,724	5,199	785	7.498	293.5	6.382	56.836
2021/01/06	01월	화	눈	5,359	6,037	4,938	791	6.464	204.5	5.345	55.991
2021/01/07	01월	수	맑음	5,144	5,420	4,585	827	8.495	246.5	4.935	50.879
2021/01/08	01월	목	맑음	5,931	5,909	5,329	790	6.793	177.3	4.271	47.755
2021/01/09	01월	금	맑음	5,922	6,155	5,359	782	9.293	126	19.942	98.721

2.3.1. 소스

스크립트

```
<script>
    widget.componentReadyHandler = function(componentId){
        if(!widget.isBindingSource()){
            document.getElementById(componentId).setDataType("json");
            document.getElementById(componentId).setDataURLEx(BIX5.getAssetsPath() + "Datasources/data2.json");
        }
    }

    widget.itemClickJsFunction = function(event){
        var data = event.itemRenderer.getData();
        alert(data["일자"]);
    }

    widget.componentRemoveHandler = function(){
    }
</script>
```


2.4. 그리드 위젯 확장하기 II

학습 목표

- 그리드의 HTML 렌더러 사용자 정의 함수를 이용하여 게이지형 차트를 표시할 수 있다

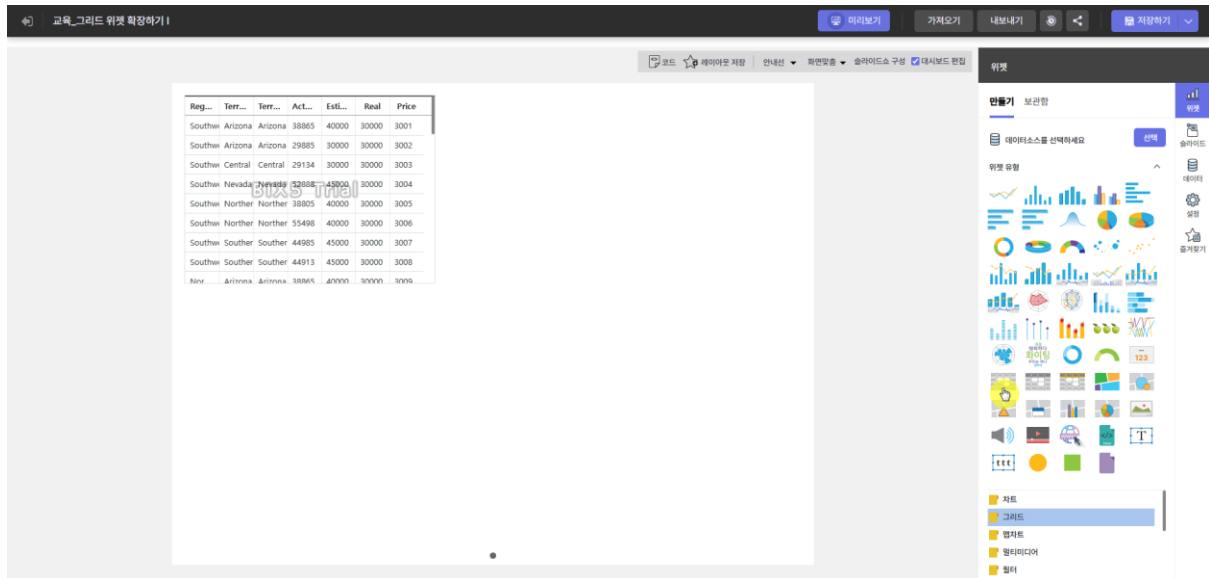
사전 요구 사항

- HTML, CSS, JavaScript 를 다뤄본 경험
- BIX5 를 이용하여 대시보드 제작한 경험
- 대시보드 개발자 권한으로 로그인
- 데이터셋: 샘플 데이터 소스 - 하수처리 데이터
- 시나리오 : 하수처리 데이터에서 일자, 월, 요일, 날씨, 방류량, 유입량, 증설유입량, 기존유입량 출력 후 방류량 컬럼에 방류량/6000 값을 게이지형 차트를 표시한다.

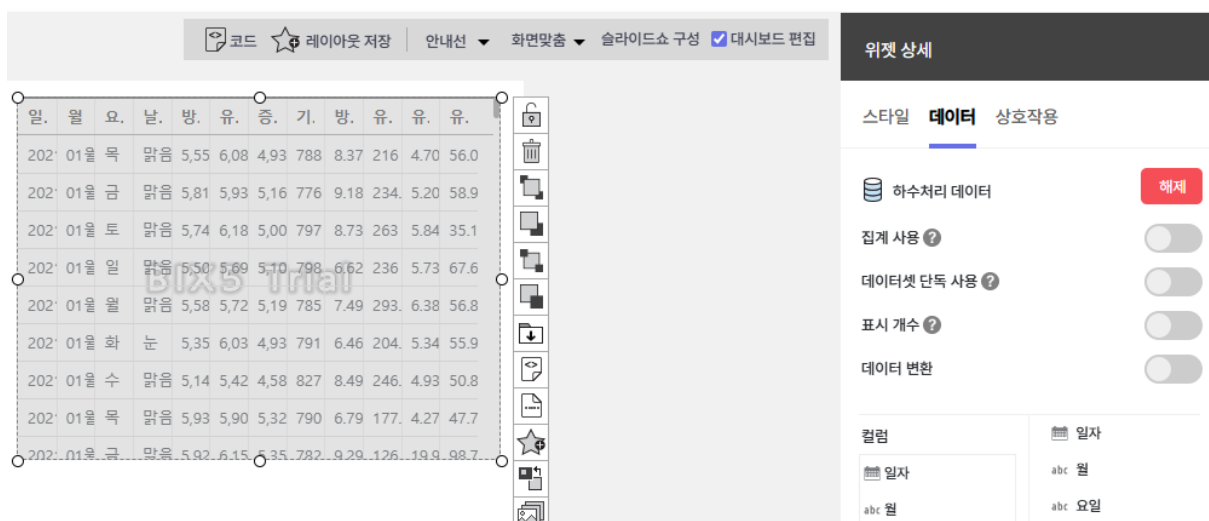
최종 결과

일자	월	요일	날씨	방류량	유입량	증설유입량	기존유입량
2021/01/01	01월	목	맑음	5554	6,087	4,936	788
2021/01/02	01월	금	맑음	5815	5,935	5,166	776
2021/01/03	01월	토	맑음	5742	6,182	5,001	797
2021/01/04	01월	일	맑음	5507	5,695	5,103	798
2021/01/05	01월	월	맑음	5588	5,724	5,199	785
2021/01/06	01월	화	눈	5359	6,037	4,938	791
2021/01/07	01월	수	맑음	5144	5,420	4,585	827
2021/01/08	01월	목	맑음	5931	5,909	5,329	790

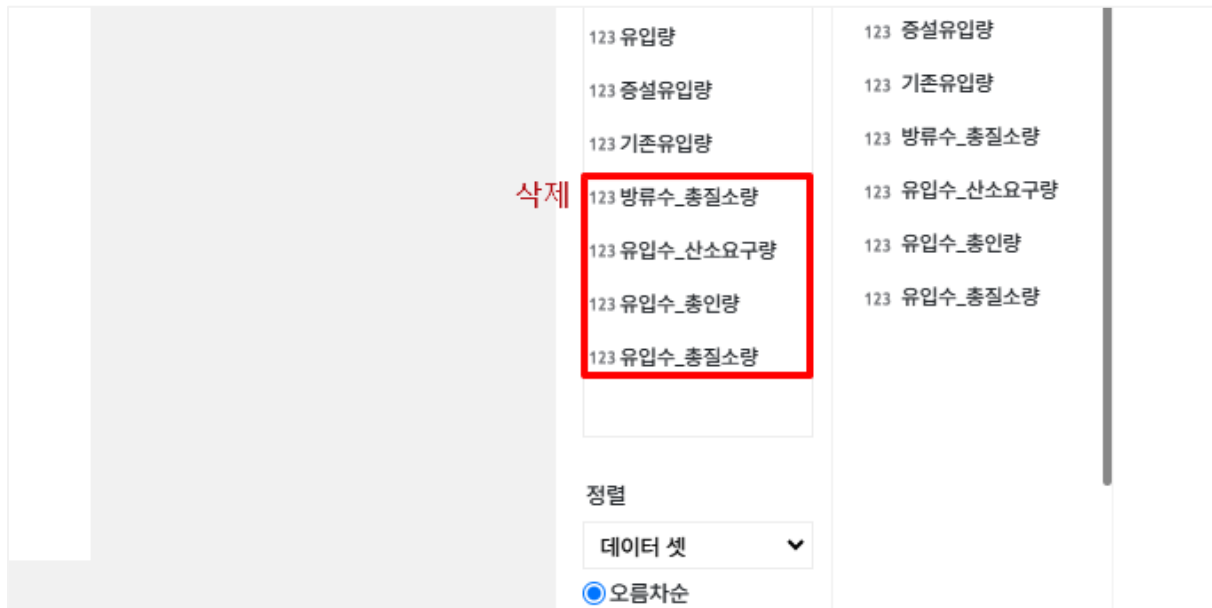
1. 위젯 만들기에서 그리드를 생성합니다.



2. 그리드 위젯에 샘플 데이터 → 하수처리 데이터를 바인딩 합니다.



3. 방류수_총질소량, 유입수_산소요구량, 유입수_총인량, 유입수_총질소량 필드를 삭제합니다.



123 유입량

123 증설유입량

123 기존유입량

123 방류수_총질소량

123 유입수_산소요구량

123 유입수_총인량

123 유입수_총질소량

123 증설유입량

123 기존유입량

123 방류수_총질소량

123 유입수_산소요구량

123 유입수_총인량

123 유입수_총질소량

정렬

데이터 셋

오름차순

4. 상단의 스타일 메뉴로 이동하여 컬럼 렌더러→방류량의 렌더러를 HTML 로 변경합니다.



4,585	827
5,329	790
5,359	782

텍스트 정렬

컬럼 렌더러

일자

월

요일

날씨

방류량

유입량

증설유입량

기존유입량

컬럼 필터링

텍스트

HTML

이미지

여러줄 텍스트

텍스트

줄여찾기

5. 위젯의 코드에디터 → 레이아웃으로 이동하여 방류량 DataGridColumn 에 labelJsFunction 을 등록합니다. labelJsFunction 을 사용하기 위해 formatter 를 삭제합니다.

레이아웃 스크립트 컴포넌트 옵션 사용자 데이터

```
1 <BIX5>
2 <NumberFormatter id="numfmt" useThousandsSeparator="true"/>
3 <DataGrid headerColors="#FFFFFF, #FFFFFF" color="#31393F" alternatingItemColors="[transparent,transparent]" backgroundColor="transparent"
  headerStyleName="w207BBF88FBA6B790B1F8E546AA345A39_HeaderStyle" paddingTop="7" paddingBottom="7" headerPaddingTop="7" headerPaddingBottom="7"
4 <columns>
5   <DataGridColumn dataField="일자" headerText="일자"/>
6   <DataGridColumn dataField="월" headerText="월"/>
7   <DataGridColumn dataField="요일" headerText="요일"/>
8   <DataGridColumn dataField="날씨" headerText="날씨"/>
9   <DataGridColumn dataField="방류량" headerText="방류량" textAlign="right" itemRenderer="HtmlItem" labelJsFunction="widget.labelJsFunc"/>
10  <DataGridColumn dataField="유입량" headerText="유입량" textAlign="right">
11    <formatter>
12      <NumberFormatter useThousandsSeparator="true"/>
13    </formatter>
14  </DataGridColumn>
15  <DataGridColumn dataField="증설유입량" headerText="증설유입량" textAlign="right">
16    <formatter>
```

함수를 등록할 때 'widget'은 예약어로 위젯 객체를 가리킵니다. 현재 예시에서는 DataGridColumn 의 LabelJsFunction 에 widget 객체의 labelJsFunc 함수를 등록한 것입니다.

예시(생략된 레이아웃. DataGridColumn 만 복사하여 사용하세요.)

```
<columns>
  <DataGridColumn dataField="방류량" headerText="방류량" textAlign="right"
itemRenderer="HtmlItem" labelJsFunction="widget.labelJsFunc">
  </DataGridColumn>
  ...
</columns>
```

6. 상단 탭의 스크립트로 이동하여 5 에서 등록한 widget.labelJsFunc 를 작성합니다. value/6000 값을 게이지형 차트를 표시합니다.

레이아웃 **스크립트** 컴포넌트 옵션 사용자 데이터

```
1 <script>
2 | widget.labelJsFunc = function(item, value, column){
3   var percent = (value / 6000 * 100).toFixed(0);
4   return '<div style="width:100%; height:18px; text-align:left;"><div style="width:'+pe
   text-align:center; padding-right:8px; width:100%; height:18px;">'+value+'</div></div>';
5 }
```

```
<script>
  widget.componentReadyHandler = function(componentId){
    if(!widget.isBindingSource()){
      document.getElementById(componentId).setDataType("json");
      document.getElementById(componentId).setDataURLEx(BIX5.getAssetsPa
th() + "Datasources/data2.json");
    }
  }

  widget.labelJsFunc = function(item, value, column){
    var percent = (value / 6000 * 100).toFixed(0);
    return '<div style="width:100%; height:18px; text-align:left;"><div
style="width:'+percent+'%; height:18px; background-color:#FFCC00;"><div
style="position:absolute; text-align:center; padding-right:8px; width:100%;
height:18px;">'+value+'</div></div>';
  }

  widget.componentRemoveHandler = function(){
  }
</script>
```

등록한 함수의 리턴 값으로 그리드 셀에 표시합니다. 컬럼에 아이템 렌더러가 HTML 렌더러로 등록되어 있어 콜백함수에서 리턴 하는 문자열을 HTML 그대로 출력하게 됩니다.

사용하고자 하는 사용자정의함수의 자세한 사용법은 차트 컴포넌트의 API 에서 확인하실 수 있습니다.

<https://demo.bix5.net/SDK/Docs/Grid/API/DataGridColumn.html#labelJsFunction>

7. 적용을 눌러 확인합니다.

일자	월	요일	날씨	방류량	유입량	증설유입량	기존유입량
2021/01/01	01월	목	맑음	5554	6,087	4,936	788
2021/01/02	01월	금	맑음	5815	5,935	5,166	776
2021/01/03	01월	토	맑음	5742	6,182	5,001	797
2021/01/04	01월	일	맑음	5507	5,695	5,103	798
2021/01/05	01월	월	맑음	5588	5,724	5,199	785
2021/01/06	01월	화	눈	5359	6,037	4,938	791
2021/01/07	01월	수	맑음	5144	5,420	4,585	827
2021/01/08	01월	목	맑음	5931	5,909	5,329	790

2.4.1. 소스

레이아웃

```

<BIX5>
  <NumberFormatter id="numfmt" useThousandsSeparator="true"/>
  <DataGrid headerColors="#FFFFFF, #FFFFFF" color="#31393F"
alternatingItemColors="[transparent,transparent]"
backgroundColor="transparent" _cType="DataGrid"
headerStyleName="w207BBF88FBA6B790B1F8E546AA345A39_HeaderStyle" paddingTop="7"
paddingBottom="7" headerPaddingTop="7" headerPaddingBottom="7" fontSize="13">
    <columns>
      <DataGridColumn dataField="일자" headerText="일자"/>
      <DataGridColumn dataField="월" headerText="월"/>
      <DataGridColumn dataField="요일" headerText="요일"/>
      <DataGridColumn dataField="날씨" headerText="날씨"/>
      <DataGridColumn dataField="방류량" headerText="방류량"
textAlign="right" itemRenderer="HtmlItem"
labelJsFunction="widget.labelJsFunc"/>
      <DataGridColumn dataField="유입량" headerText="유입량"
textAlign="right">
        <formatter>
          <NumberFormatter useThousandsSeparator="true"/>
        </formatter>
      </DataGridColumn>
      <DataGridColumn dataField="증설유입량" headerText="증설유입량"
textAlign="right">
        <formatter>
          <NumberFormatter useThousandsSeparator="true"/>
        </formatter>
      </DataGridColumn>
      <DataGridColumn dataField="기존유입량" headerText="기존유입량"
textAlign="right">
        <formatter>
          <NumberFormatter useThousandsSeparator="true"/>
        </formatter>
      </DataGridColumn>
    </columns>
  </DataGrid>
</BIX5>

```

스크립트

```

<script>
    widget.componentReadyHandler = function(componentId){
        if(!widget.isBindingSource()){
            document.getElementById(componentId).setDataType("json");
            document.getElementById(componentId).setDataURLEx(BIX5.getAssetsPath() + "Datasources/data2.json");
        }
    }

    widget.labelJsFunc = function(item, value, column){
        var percent = (value / 6000 * 100).toFixed(0);
        return '<div style="width:100%; height:18px; text-align:left;"><div style="width: '+percent+'%; height:18px; background-color:#FFCC00;"><div style="position:absolute; text-align:center; padding-right:8px; width:100%; height:18px;">'+value+'</div></div>';
    }

    widget.componentRemoveHandler = function(){
    }
</script>

```


2.5. 맵차트 위젯 확장하기 I

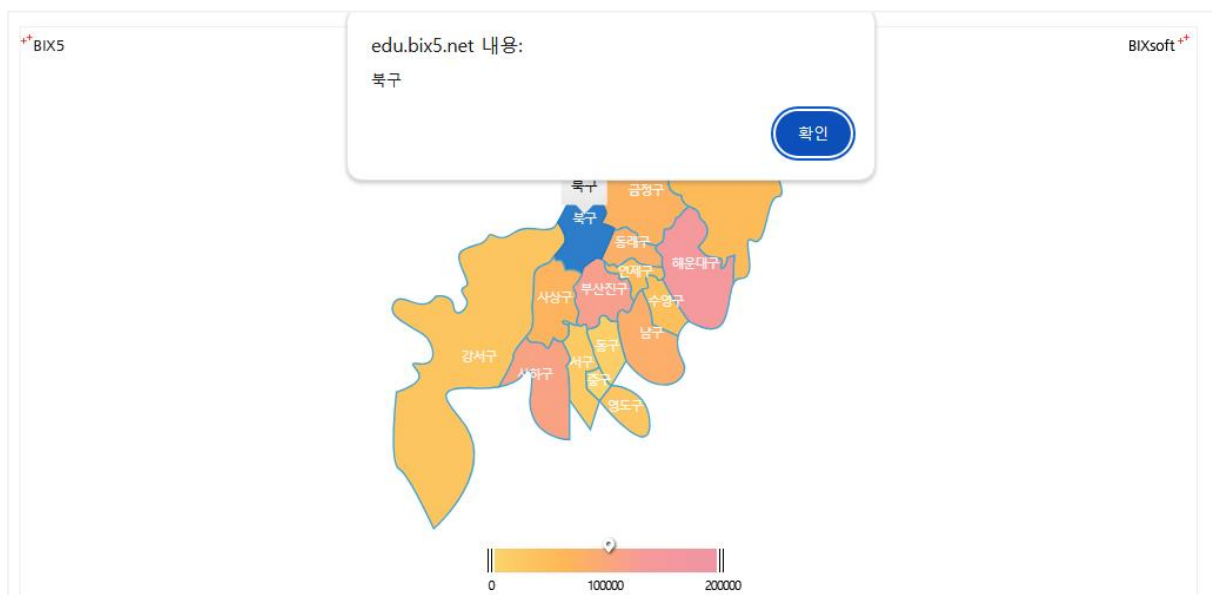
학습 목표

- 맵차트 맵클릭 이벤트를 등록하여 사용자 함수를 작성할 수 있다.

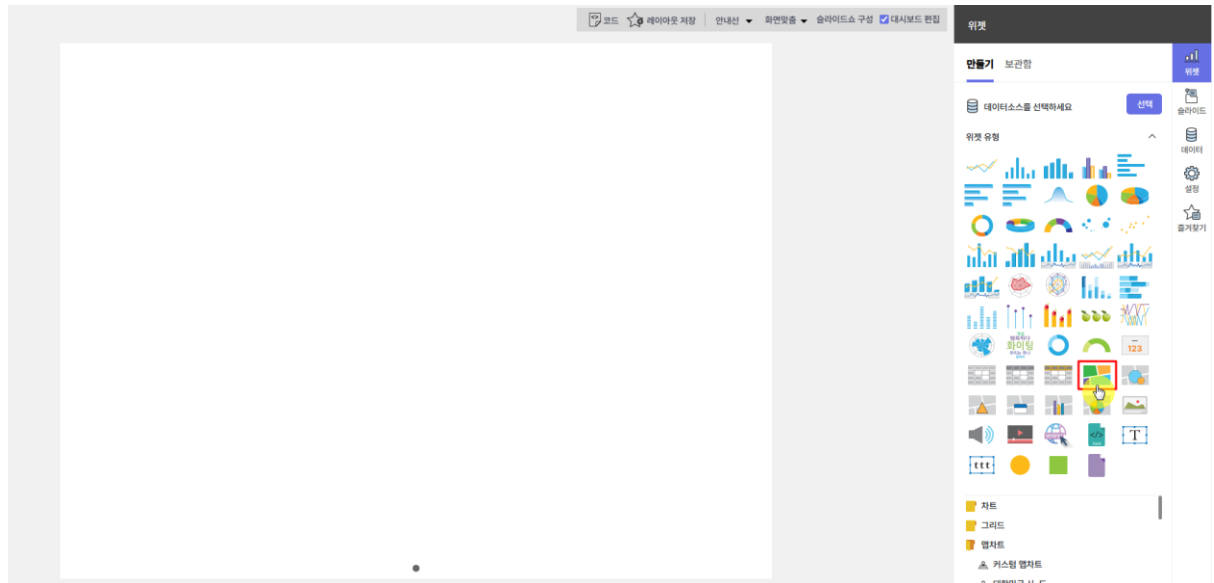
사전 요구 사항

- HTML, CSS, JavaScript 를 다뤄본 경험
- BIX5 를 이용하여 대시보드 제작한 경험
- 대시보드 개발자 권한으로 로그인
- 데이터셋: 샘플 데이터 소스 -부산시유동인구
- 시나리오: 부산시유동인구 데이터를 부산시 지도에 출력하고 맵클릭 이벤트를 등록하여 값을 대화상자로 출력한다.

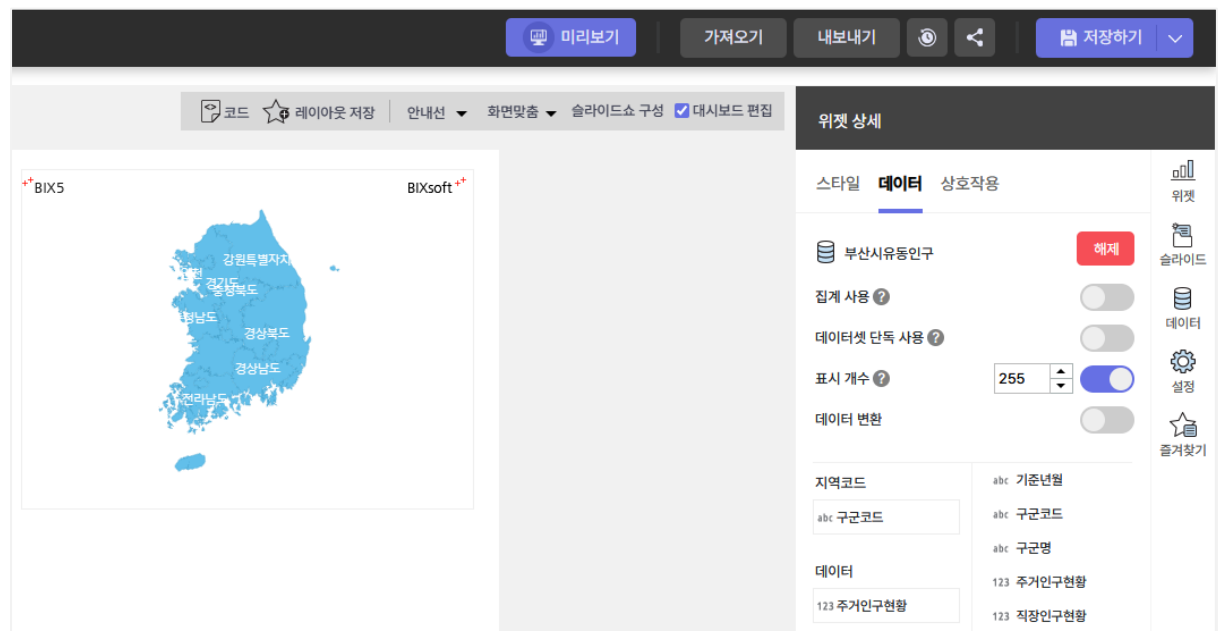
최종 결과



1. 위젯 만들기에서 맵차트 위젯을 생성합니다. (또는 목록에서 대한민국 드릴다운)



2. 맵차트 위젯에 샘플 데이터 → 부산시유동인구를 바인딩 하고 지역코드에 구군코드, 데이터에 주거인구현황 필드를 설정합니다.



3. 스타일 메뉴에서 옵션을 설정합니다.

지도 → 대한민국 드릴다운, 루트코드 → 부산



범례 → 데이터 영역 범례, 데이터 최댓값 → 200,000 데이터 간격 → 100,000



4. 코드 에디터 → 스크립트로 이동하여 지도클릭 이벤트를 받기 위해 widget 객체에 mapChangeJsFunction 함수를 등록합니다.

레이아웃 **스크립트** 컴포넌트 옵션 사용자 데이터

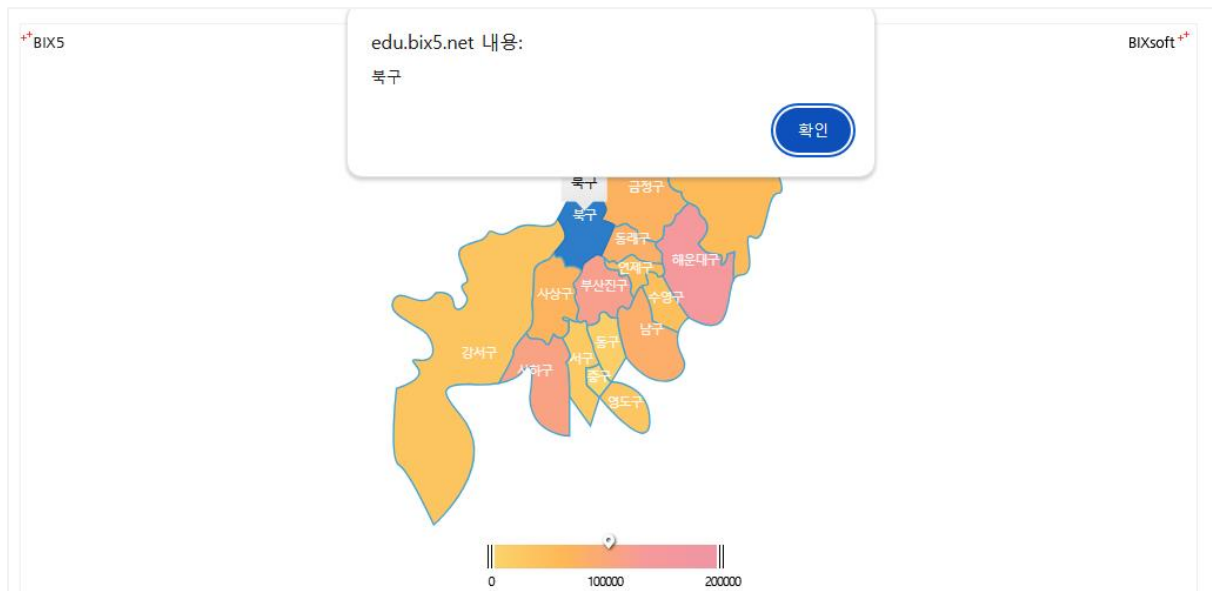
```
1 <script>
2   widget.mapChangeJsFunction = function(code, label, data){
3     alert(label);
4   }
5 </script>
```

```
<script>
  widget.mapChangeJsFunction = function(code, label, data){
    alert(label);
  }
</script>
```

사용하고자 하는 사용자정의함수의 자세한 사용법은 컴포넌트의 API 에서 확인하실 수 있습니다.

<https://demo.bix5.net/SDK/Docs/MapChart/API/MapChart.html#event:mapChangeJsFunction>

5. 적용 후 미리보기에서 지도클릭 시 지역명을 대화상자로 출력합니다.



2.5.1. 소스

스크립트

```
<script>
  widget.componentReadyHandler = function(componentId){
  }

  widget.mapChangeJsFunction = function(code, label, data){
    alert(label);
  }

  widget.componentRemoveHandler = function(){
  }
</script>
```

2.6. 맵차트 위젯 확장하기 II

학습 목표

- 맵차트 사용자 정의 함수를 이용하여 데이터팁(툴팁)을 변경 할 수 있다.

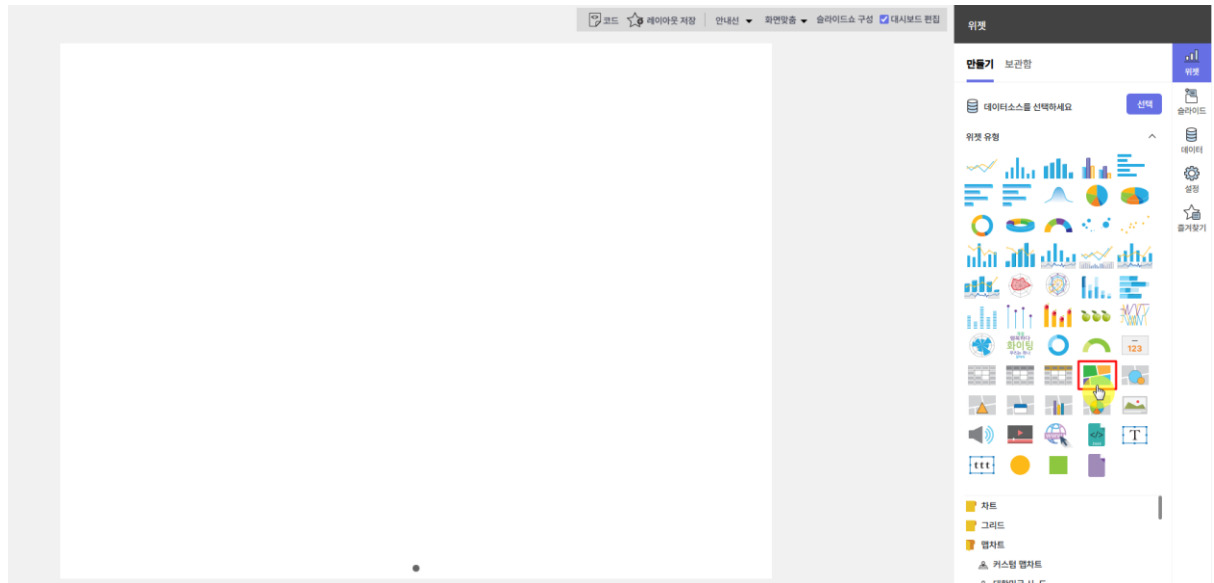
사전 요구 사항

- HTML, CSS, JavaScript 를 다뤄본 경험
- BIX5 를 이용하여 대시보드 제작한 경험
- 대시보드 개발자 권한으로 로그인
- 데이터셋: 샘플 데이터 소스 -부산시유동인구
- 시나리오: 부산시유동인구 데이터를 부산시 지도에 출력하고 데이터팁에 구군명, 주거인구현황을 출력한다.

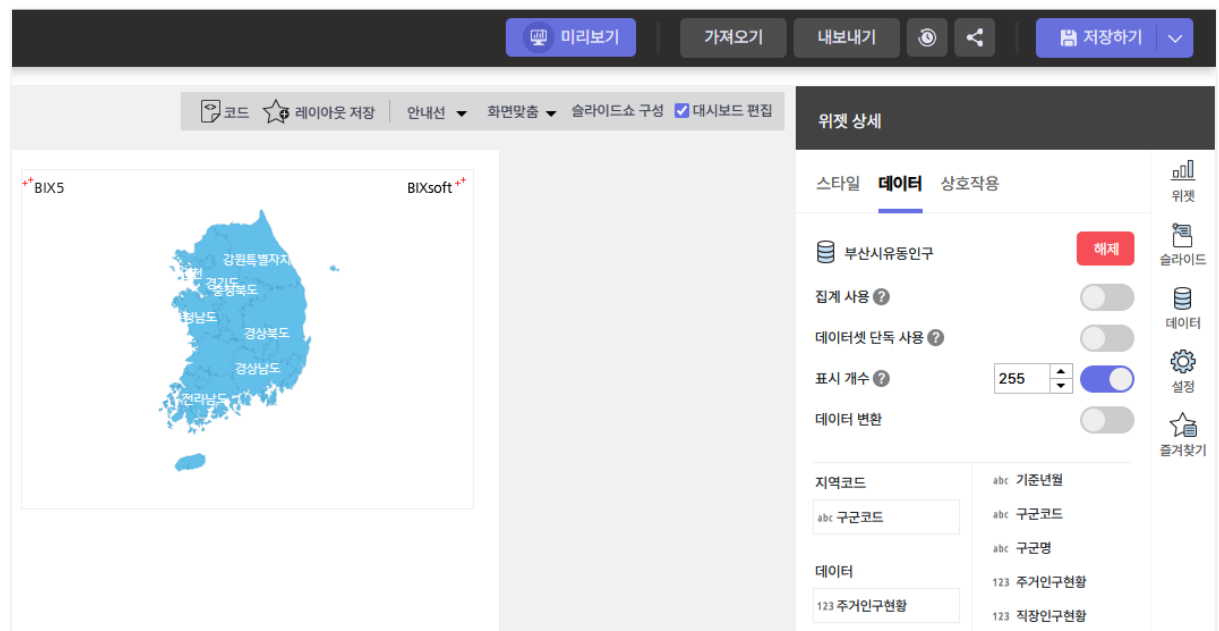
최종 결과



1. 위젯 만들기에서 맵차트 위젯을 생성합니다. (또는 목록에서 대한민국 드릴다운)



2. 맵차트 위젯에 샘플 데이터 → 부산시유동인구를 바인딩 하고 지역코드에 구군코드, 데이터에 주거인구현황 필드를 설정합니다.



3. 스타일 메뉴에서 옵션을 설정합니다.

지도 → 대한민국 드릴다운, 루트코드 → 부산



4. 코드에디터 → 레이아웃에서 MapChart 에 widget 객체의 dataTipJsFunc 함수를 dataTipJsFunction 을 등록하고 스크립트에 widget.dataTipJsFunc 를 작성합니다.

레이아웃 스크립트 **컴포넌트 옵션** 사용자 데이터

```

1 <BIX5 backgroundColor="transparent" alpha="1" padding="
2 <MapChart drillDownEnabled="true" showDataTips="true"
  dataTipJsFunction = "widget.dataTipJsFunc">
3 <series>
4 <MapSeries localFillByRange="[]" rangeLegendDataF
  displayName="Map" selectionFontColor="#5244da" _nodeName
5 <stroke>
6 <Stroke color="#48a5d0" weight="1" alpha="1"/
7 </stroke>
  
```

```

<MapChart drillDownEnabled="true" showDataTips="true" dataTipType="Type3"
_cType="MapChart" rootCode="2600000000" dataTipJsFunction=
"widget.dataTipJsFunc">
  
```


레이아웃 스크립트 컴포넌트 옵션 사용자 데이터

```

1 <script>
2 widget.dataTipJsFunc = function(seriesId, code, label, data){
3     return label + "<br>" + data["주거인구현황"];
4 }
5 </script>

```

```

<script>
widget.dataTipJsFunc = function(seriesId, code, label, data){
    return label + "<br>" + data["주거인구현황"];
}
</script>

```

사용하고자 하는 사용자정의함수의 자세한 사용법은 컴포넌트의 API에서 확인하실 수 있습니다.

<https://demo.bix5.net/SDK/Docs/MapChart/API/MapChart.html#event:dataTipJsFunction>

5. 미리보기에서 지역에 마우스 오버하여 데이터 툴팁 내용을 확인합니다.



2.6.1. 소스

레이아웃

```
<BIX5 backgroundColor="transparent" alpha="1" padding="10">
  <MapChart drillDownEnabled="true" showDataTips="true" dataTipType="Type3"
    _cType="MapChart" rootCode="2600000000"
    dataTipJsFunction="widget.dataTipJsFunc">
    <series>
      <MapSeries localFillByRange="[]" rangeLegendDataField=""
        fontSize="11" interactive="true" selectionMarking="line" color="#ffffff"
        labelPosition="inside" displayName="Map" selectionFontColor="#5244da"
        _nodeName="MapSeries" areaDataCode="구군코드">
        <stroke>
          <Stroke color="#48a5d0" weight="1" alpha="1"/>
        </stroke>
        <rollOverFill>
          <SolidColor color="#2e7dca"/>
        </rollOverFill>
        <selectionStroke>
          <Stroke color="#48a5d0" weight="1" alpha="1"/>
        </selectionStroke>
        <localFill>
          <SolidColor color="#62bfea"/>
        </localFill>
      </MapSeries>
    </series>
  </MapChart>
</BIX5>
```

스크립트

```
<script>
  widget.componentReadyHandler = function(componentId){
    if(!widget.isBindingSource()){
    }
  }

  widget.dataTipJsFunc = function(seriesId, code, label, data){
    return label + "<br>" + data["주거인구현황"];
  }

  widget.componentRemoveHandler = function(){
  }
</script>
```

감사합니다