

---

# BIX5 Dashboard for HTML5 사용설명서

**Version 21.0**



정품을 구입하신 고객에게는 기술상담 및 지원을 제공합니다

(이 매뉴얼은 BIX5 Enterprise 기준으로 작성되었습니다.)

서울시 영등포구 영신로 220, 610 호

(영등포동 8 가, KnK 디지털타워)

(Tel : 02-2655-9784, [bix5cs@bix5.net](mailto:bix5cs@bix5.net))

**(주)빅스소프트**

# **목 차**

1. 개요.....	4
1.1. BIX5 Dashboard 의 주요 특징.....	4
1.2. 제품 구성 정보.....	4
1.3. 시스템 요구사항.....	4
1.4. 대시보드 시스템 구성 환경 설명.....	4
1.5. 제품의 구성요소.....	5
2. BIX5 Dashboard 설치 및 라이선스 등록.....	6
2.1. BIX5 Dashboard 설치.....	6
2.2. BIX5 Dashboard 설치 수행자.....	6
2.3. BIX5 라이선스 등록하기.....	6
3. 대시보드의 기본 구조 및 용어 설명.....	7
3.1. 대시보드 기본구조 설명.....	7
3.2. 대시보드 설명.....	8
3.3. 설정창 설명.....	9
3.4. Widget 설명.....	15
3.4.1. Widget 레이아웃.....	15
3.4.2. Widget 타입.....	16
4. 대시보드 생성하기.....	18
5. 대시보드의 속성, 함수 및 이벤트.....	28
5.1. BIX5.dashboard.....	28
5.2. Dashboard.....	30
5.3. Slide.....	33
5.4. Widget.....	34
5.5. SideBar.....	37
6. 위젯 설정 방법.....	39
6.1. Text.....	39
6.2. Chart.....	40
6.3. Grid.....	43
6.4. MapChart.....	46
6.5. Html.....	48
6.6. Iframe.....	49
6.7. Video.....	50
6.8. Youtube.....	51
6.9. Clock.....	52
6.10. Image.....	53
6.11. Sound.....	54
7. 위젯 간 연동 방법.....	55
8. 제품 유지보수.....	58
9. BIX5 Dashboard 의 기타 유의사항 설명.....	59
9.1. BIX5 Dashboard 에서 지원하지 않는 기능.....	59
9.2. 대시보드 사용시 유의사항.....	59
9.3. 위젯 아이디 설정 시 유의사항.....	59

9.4. 위젯 콘텐츠 설정 시 유의사항.....	59
----------------------------	----

## 1. 개요

### 1.1. BIX5 Dashboard 의 주요 특징.

BIX5 Dashboard 는 ActiveX 기술을 배제한 순수 자바스크립트로 개발된 대시보드 솔루션을 제공합니다. HTML, .NET, JSP, PHP, ASP, ColdFusion 등과 같은 웹 스크립트 언어에 구애받지 않고, 대시보드를 생성할 수 있습니다.

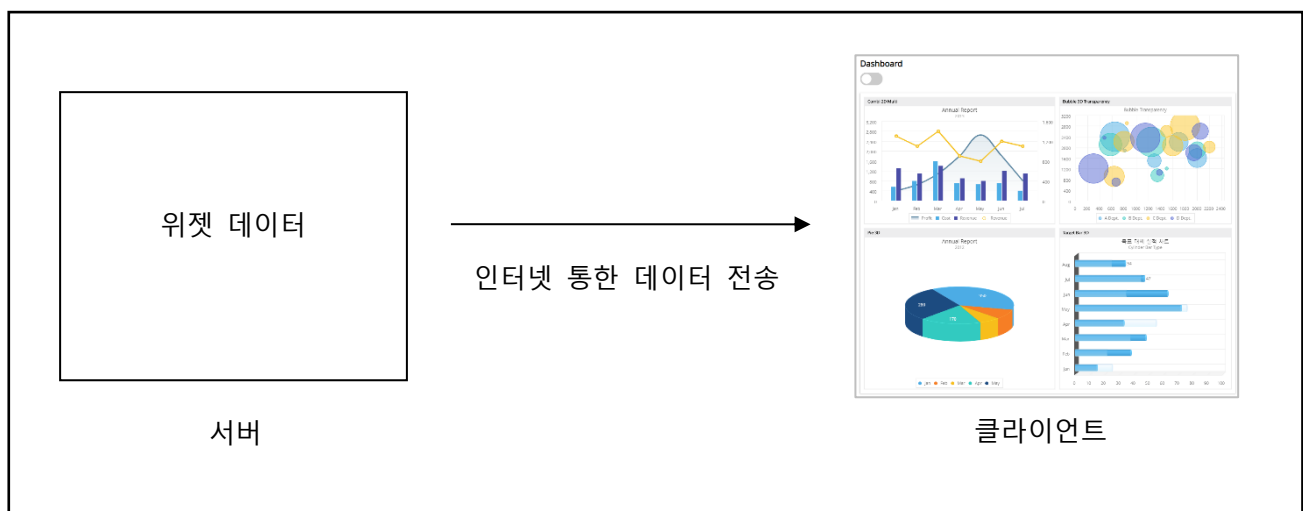
### 1.2. 제품 구성 정보

라이선스 증서 (1 부), 제품 (이메일 또는 CD 제공)

### 1.3. 시스템 요구사항

- 서버 사이드 : 순수 자바스크립트와 HTML5 기술을 바탕으로 두기 때문에 톰캣, IIS, 웹로직, 웹스피어, 제우스, LCDS 등등 모든 WAS 서버에서 작동하며, **서버 스크립트 언어에 의존적이지 않습니다.**
- 클라이언트 사이드 : HTML5 의 Canvas 및 SVG 를 정상적으로 지원하는 브라우저가 필요합니다.
- 크롬 브라우저 버전 4.0 이상부터 지원합니다.

### 1.4. 대시보드 시스템 구성 환경 설명



<그림 1 서버와 클라이언트 간의 구성도>

대시보드를 생성하려면 위젯의 메타정보를 담은 배열이 필요합니다. x좌표, y좌표, 위젯의 너비, 높이 등을 대시보드에 전달하면 대시보드는 전달된 데이터를 토대로 위젯을 생성합니다.

대시보드의 데이터 연동은 다음과 같은 형태를 지원합니다.

- 자바스크립트에서 배열을 선언하고 값을 넣어 array 객체를 전달하는 형태.

이에 대한 자세한 설명은 제공된 CD 의 샘플을 실행시켜 다음 항목에서 확인할 수 있습니다.

## 1.5. 제품의 구성요소

제공된 BIX5 Dashboard CD 의 내용은 아래와 같습니다.

디렉토리 구조는 BIX5, LicenseKey, Docs, Samples 로 구성되어 있으며 각각 디렉토리 역할은 아래와 같습니다.

### 가. BIX5

제품의 자바스크립트(BIX5.js 등) 및 필요한 이미지가 위치하고 있으며, 실제 제품을 실행하는데 필요한 최소한의 파일이 들어 있습니다. 적용시 BIX5 디렉토리를 전체로 서버에 올리고 해당 url 을 적용하시면 작업이 편리합니다.

### 나. LicenseKey

BIX5 라이선스가 있는 폴더입니다. (BIX5License.js)

### 다. Docs

설명서는 PDF 파일 형태로 제공됩니다.

### 라. Samples

각종 예제를 볼 수 있도록 작성된 html 과 xml 파일입니다.

## 2. BIX5 Dashboard 설치 및 라이선스 등록

### 2.1. BIX5 Dashboard 설치

제품을 구동할 서버로 다운로드하십시오.

### 2.2. BIX5 Dashboard 설치 수행자

제품의 설치에 클라이언트가 수행하게 됩니다.

### 2.3. BIX5 라이선스 등록하기

대시보드를 생성하기 위해서는 BIX5 라이선스를 등록하셔야 합니다. 제공된 시디의 다음경로에 있는 파일이 대시보드에 삽입할 라이선스입니다.

- /LicenseKey/BIX5License.js

라이선스를 등록하는 방법은 위 자바스크립트 파일을 <head> 태그 안에 삽입하시면 됩니다.

다른 작업을 하실 필요는 없습니다.

```
<html>
<head>

...
...
<!--BIX5 Dashboard 라이선스 등록 완료 모습 -->
<script src="BIX5License.js" language="javascript"></script>

<!-- 사용자 정의 설정 시작 -->
<script language="JavaScript" type="text/javascript">

...
...
...

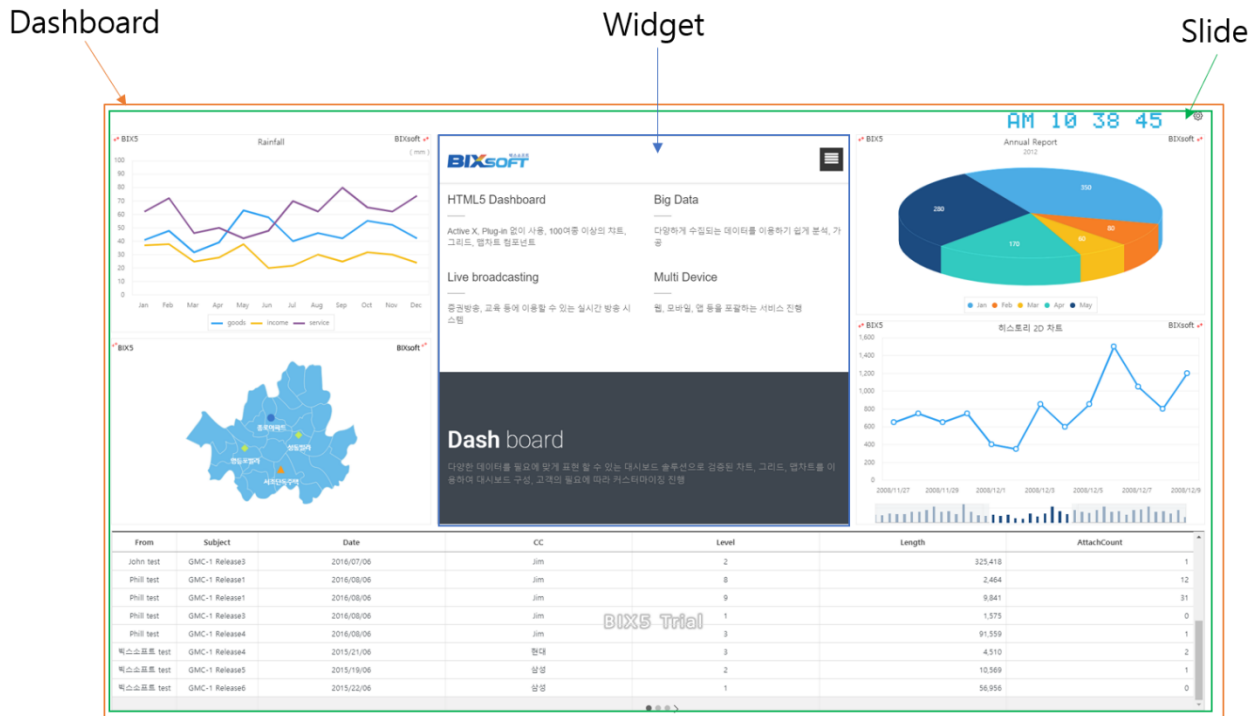
```

<예제 1 BIX5 라이선스 등록 예제>

### 3. 대시보드의 기본 구조 및 용어 설명

대시보드 생성에 앞서 대시보드가 어떤 구조로 이루어져 있는지 알아보도록 하겠습니다.

#### 3.1. 대시보드 기본구조 설명



<그림 2 대시보드 기본구조>

대시보드는 크게 Dashboard 와 Slide, Widget 으로 구성되어 있습니다.

- Dashboard : Slide 추가, 삭제, 배치 등 관리 기능을 하는 컨테이너입니다.
- Slide : Widget 을 추가, 삭제, 배치 등 관리 기능을 하는 컨테이너입니다.
- Widget : 대시보드에 표현하고자 하는 콘텐츠를 담는 컨테이너 입니다.

### 3.2. 대시보드 설명

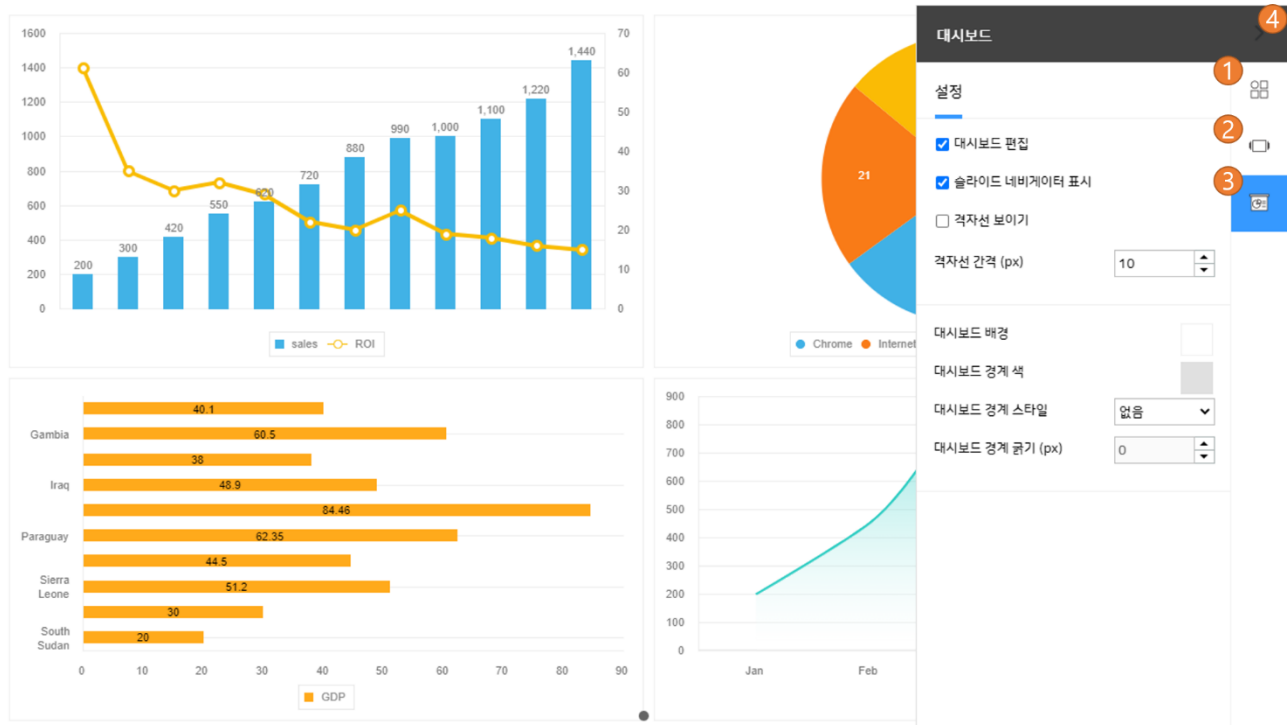


<그림 3 대시보드 설명>

- ① 설정창을 나오게 하는 버튼입니다.
- ② 슬라이드를 이동할 수 있는 네비게이터입니다. <, > 모양의 아이콘을 누르면 이전, 다음 슬라이드로 이동하고 원형 아이콘을 누르면 해당 순서의 슬라이드로 이동합니다.
- ③ 대시보드가 수정 가능한 상태일 때, 위젯을 클릭하면 이동, 리사이징이 가능합니다.
- ④ 해당 영역 위에 마우스 커서를 올리면 리사이징이 가능한 커서 모양으로 변경이 되고 마우스 다운 후 드래그 하면 리사이징을 할 수 있습니다.
- ⑤ 위젯을 클릭하면 위젯 삭제 및 위젯을 가장 위로, 위로, 아래로, 가장 아래로 이동하는 기능의 버튼이 있습니다.

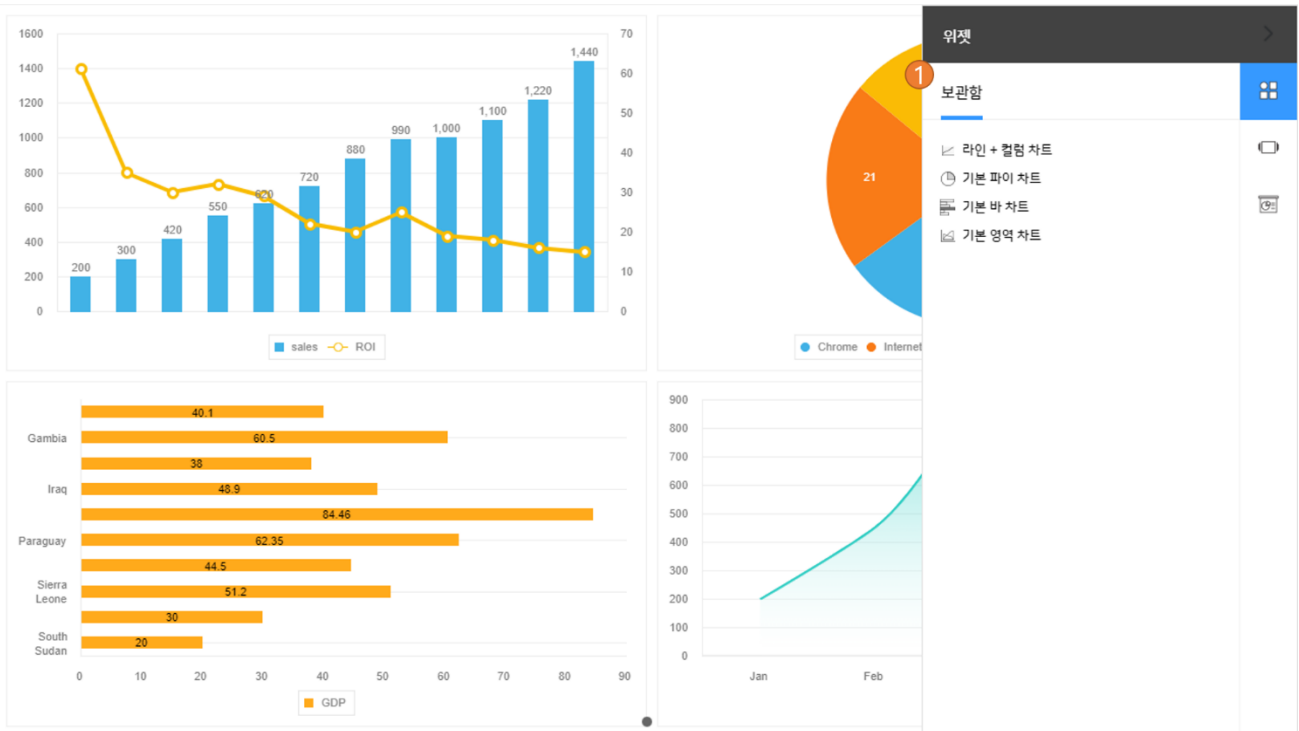


### 3.3. 설정창 설명



<그림 4 대시보드 설정창 설명>

- ① 위젯 설정 메뉴입니다.
- ② 슬라이드 설정 메뉴입니다.
- ③ 대시보드 설정 메뉴입니다.
- ④ 설정창이 닫힙니다.



<그림 5 위젯 설정 메뉴 설명>

① 위젯 보관함 메뉴입니다.

- 리스트 아이템을 클릭하면 커서가 변경됩니다. 이후 슬라이드 영역을 클릭 또는 드래그하여 위젯을 슬라이드에 추가할 수 있습니다.
- 리스트 아이템을 더블클릭하면 해당 위젯이 현재 슬라이드에 추가됩니다.
- 위젯 리스트에서 위젯 아이템을 슬라이드에 존재하는 위젯에 드랍하면 위젯이 교체됩니다.  
(브라우저에 따라 IFrame 위젯은 drag&drop 이 작동되지 않을 수 있습니다.)
- 대시보드의 setWidgetPool 을 통해 대시보드 보관함 리스트를 설정할 수 있습니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    ...
    //사이드바의 위젯 보관함 설정
    var widgets = [{label:"폴더",
    children:[
        {label:"빈 위젯", icon:"blank", type:"blank"}
    ]}];

    dashboard.setWidgetPool(widgets);
}, "100%", "100%", {});
```

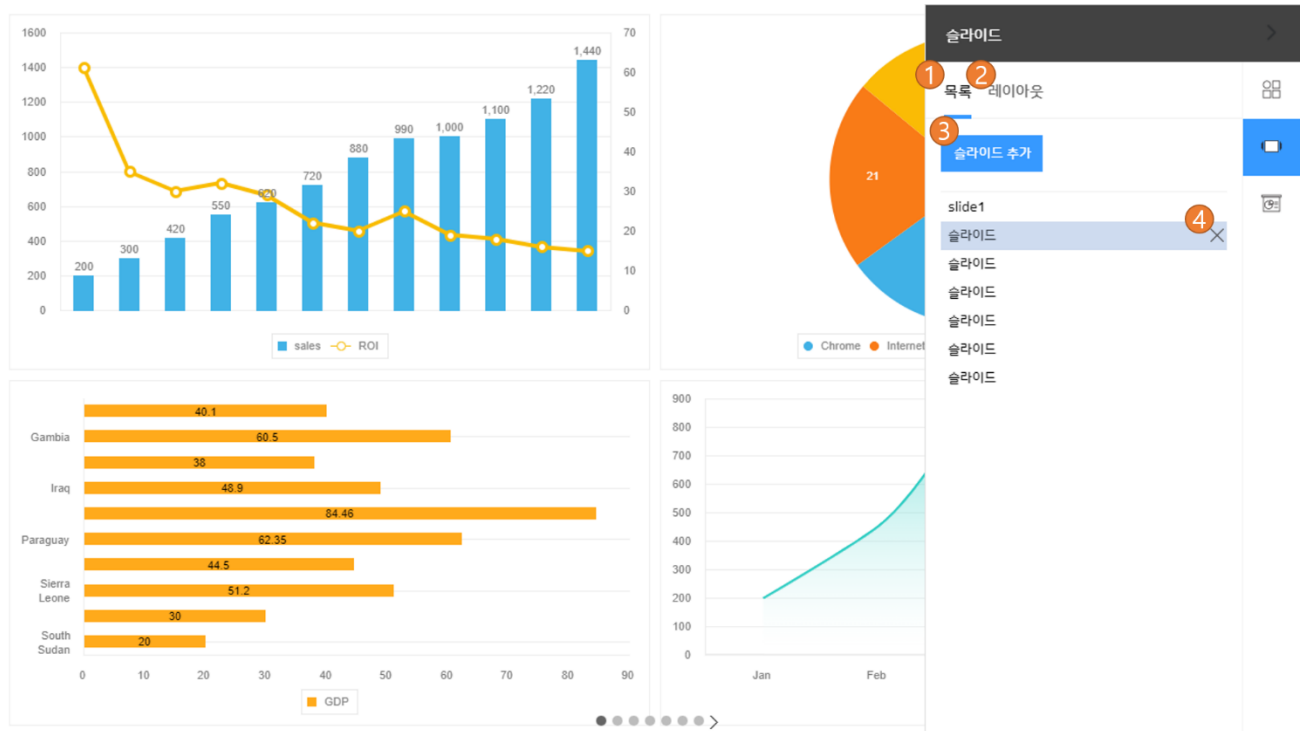


그림 6 슬라이드 설정 메뉴 설명

- ① 슬라이드 목록 메뉴입니다.
  - ② 슬라이드 레이아웃 메뉴입니다.
  - ③ 슬라이드가 추가됩니다.
  - ④ 슬라이드를 삭제합니다. (마우스 오버시 나타남)
- 슬라이드 리스트 아이템을 클릭하면 해당 슬라이드로 이동합니다.



그림 7 슬라이드 설정 메뉴 설명2

- ① 레이아웃이 새 슬라이드에 추가됩니다.
  - ② 레이아웃이 현재 슬라이드에 추가됩니다. 슬라이드의 위젯은 유지됩니다.
- 기존 슬라이드에 있던 위젯을 선택 후 ctrl+x 를 눌러 잘라낼 수 있습니다. 그리고 레이아웃 빈 위젯을 선택한 후 ctrl+v 를 눌러 붙여넣기가 가능합니다.

● 단축키에 대한 설명

단축키	설명
ctrl + c	복사
ctrl + x	잘라내기
ctrl + v	붙여넣기

- 슬라이드 레이아웃 리스트를 설정하는 방법

사이드바의 **setSlideLayouts()**로 위젯 만들기에 위젯을 설정할 수 있습니다. 기본적으로 제품에서 **slidelayouts.js** 를 제공하고 있으며 이를 설정하실 수 있습니다.

```
<head>
  <script type="text/javascript" src="/BIX5/JS/slidelayouts.js" defer></script>
  <script>
    BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
      dashboard.getSideBar().setSlideLayouts(slidelayouts);
    }, "100%", "100%", {});
  </script>
</head>
```

-

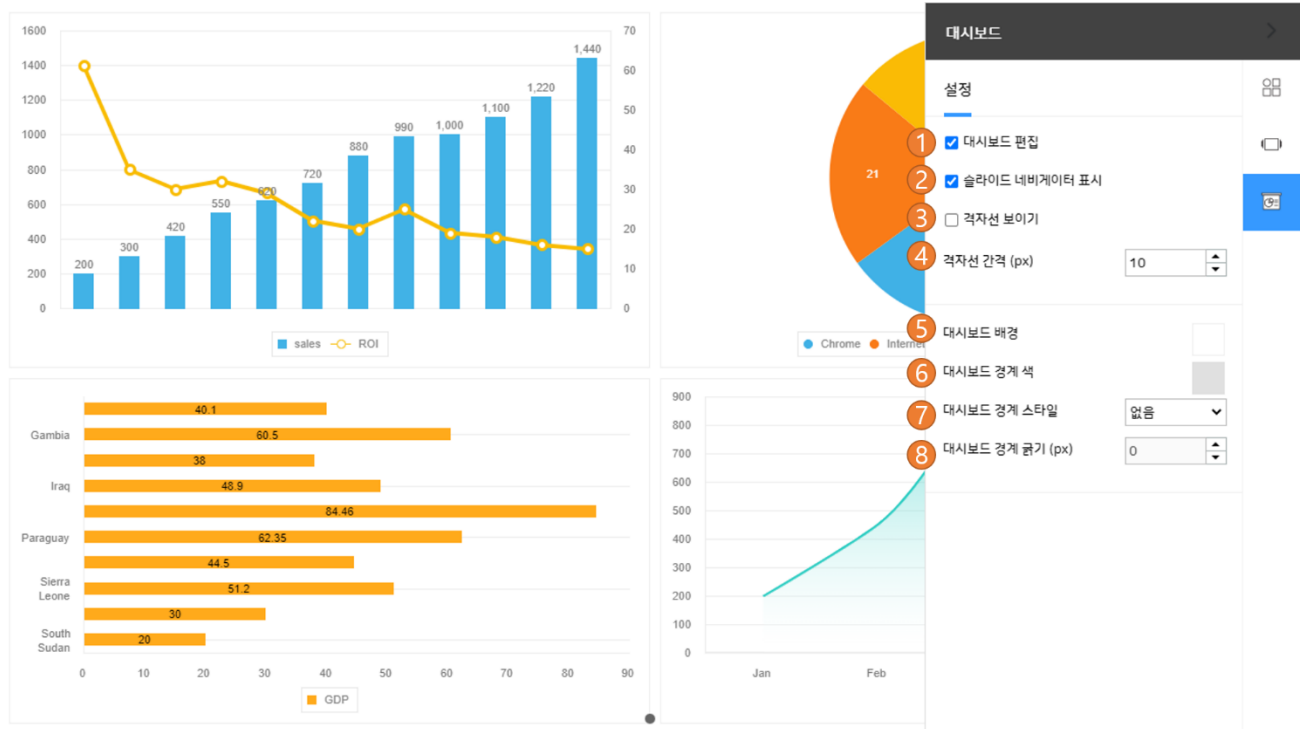
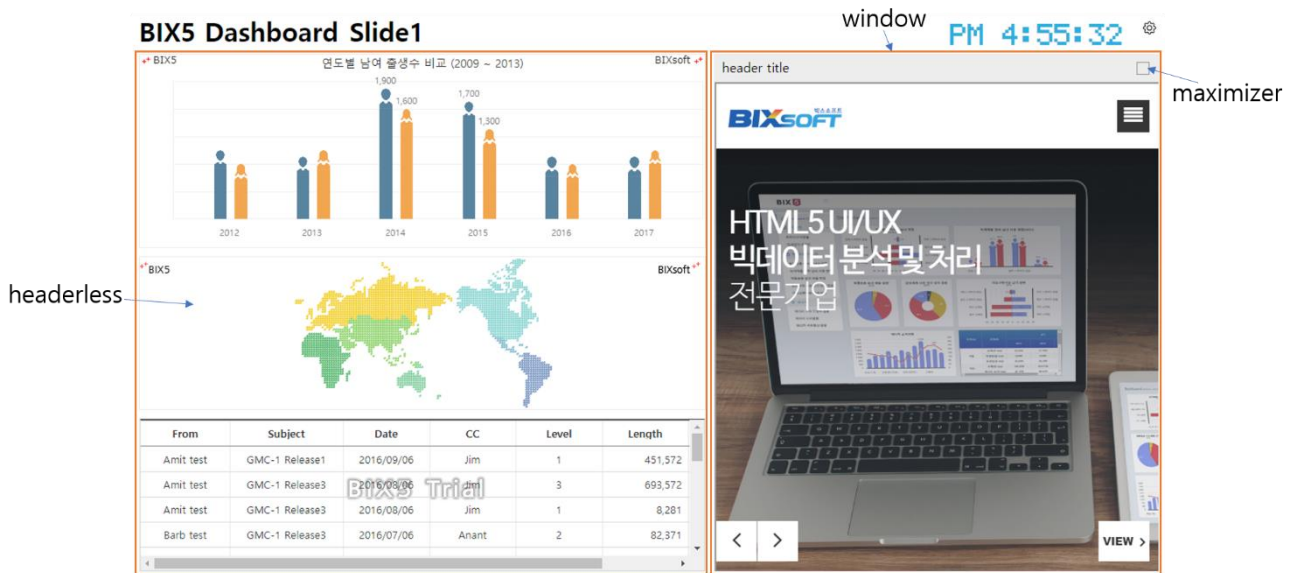


그림 8 대시보드 설정 메뉴 설명

- ① 대시보드 수정 여부를 설정합니다.
- ② 대시보드 하단의 슬라이드를 이동시킬 수 있는 네비게이터 표시 여부를 설정합니다.
- ③ 격자선 보이기 여부를 설정합니다.
- ④ 격자선 간격을 설정합니다. 픽셀 단위입니다. (30px 까지 가능)
- ⑤ 대시보드 배경 색상을 설정합니다.
- ⑥ 대시보드 경계 색상을 설정합니다.
- ⑦ 대시보드 경계 스타일을 설정합니다. 지원하는 옵션은 없음, 실선, 도트, 이중선을 입니다.
- ⑧ 대시보드 경계 굵기를 설정합니다. (30px 까지 가능)

### 3.4. Widget 설명

#### 3.4.1. Widget 레이아웃

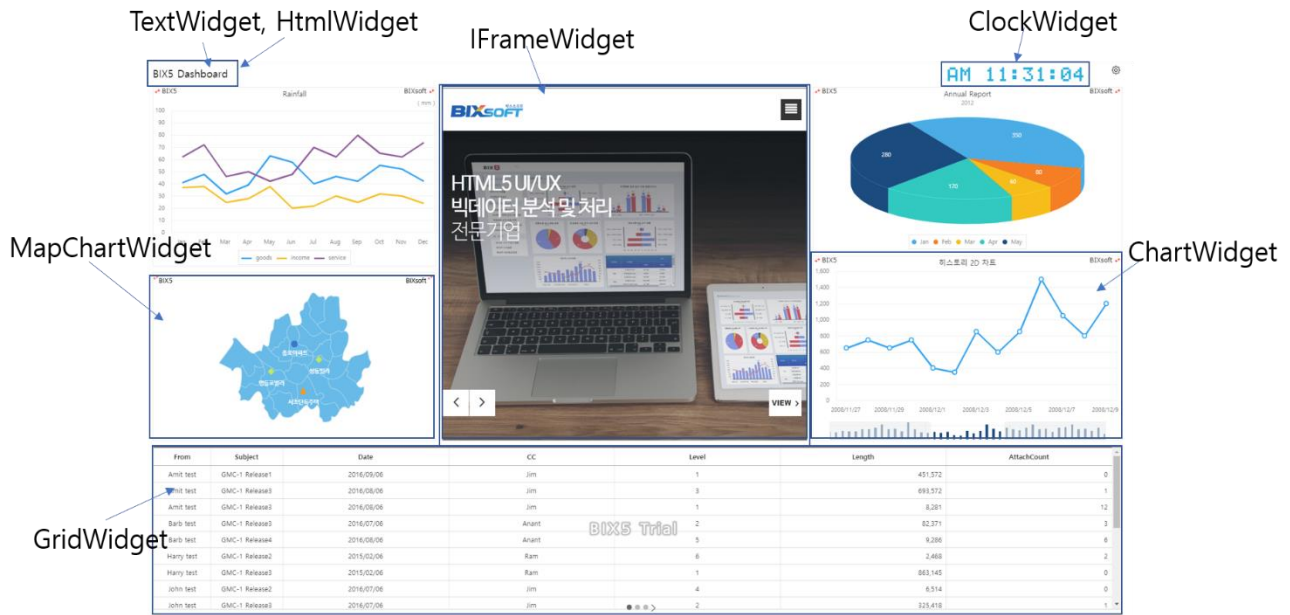


<그림 9 Widget 레이아웃>

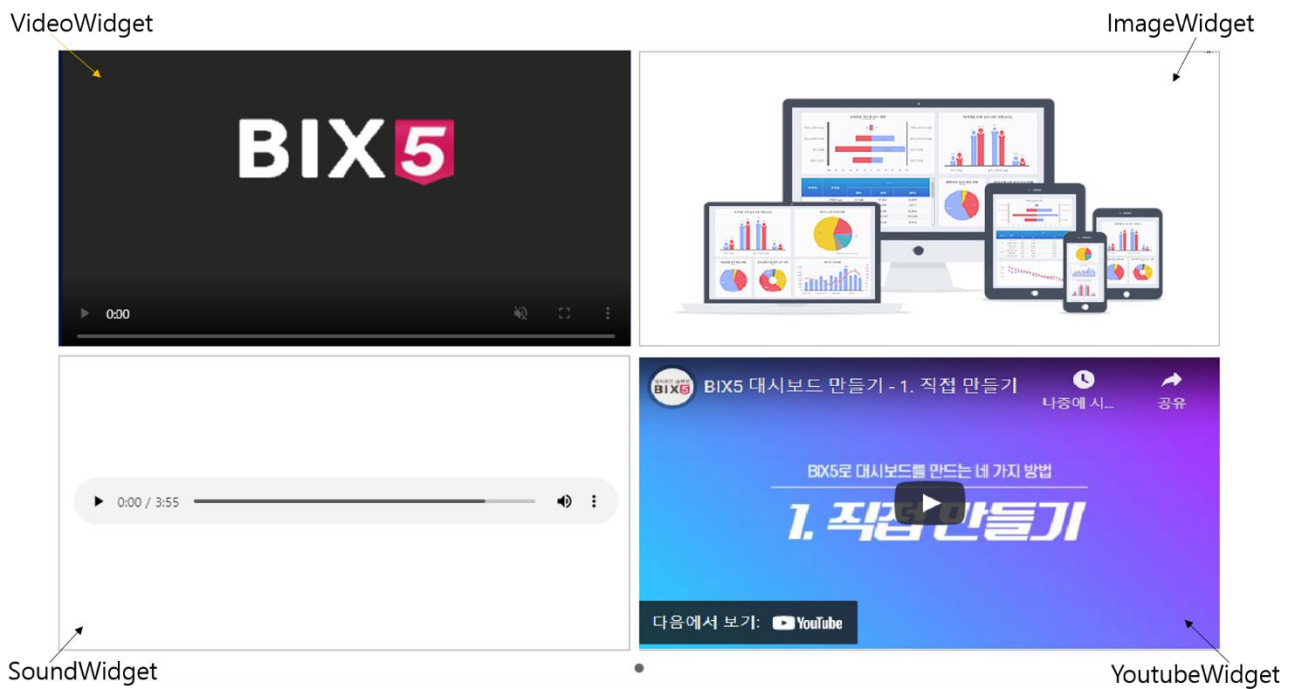
위젯 레이아웃 타입은 headerless 와 window 가 있습니다.

- headless : TitleBar 가 없는 위젯 레이아웃입니다. Border 를 제외한 영역이 content 영역입니다.
- window : TitleBar 가 있는 위젯 레이아웃입니다. TitleBar 에는 위젯 title 이 출력되고 대시보드가 수정 불가능한 상태일 때 maximizer 버튼으로 위젯을 슬라이드 크기만큼 리사이징 시킬 수 있습니다. Border 와 TitleBar 영역을 제외한 영역이 content 영역입니다.

### 3.4.2. Widget 타입



<그림 10 Widget 타입>



<그림 11 Widget 타입/2>



위젯은 총 11 종의 위젯이 있습니다.

- ClockWidget : 시계를 출력하는 위젯입니다.
- TextWidget : 텍스트를 출력하는 위젯입니다.
- ChartWidget : 콘텐츠 영역에 BIX5 Chart 를 설정할 수 있는 위젯입니다.
- GridWidget : 콘텐츠 영역에 BIX5 Grid 를 설정할 수 있는 위젯입니다.
- MapChartWidget : 콘텐츠 영역에 BIX5 MapChart 를 설정할 수 있는 위젯입니다.
- IFrameWidget : 콘텐츠 영역에 iframe 을 설정할 수 있는 위젯입니다.
- HtmlWidget : HTML 스트링을 이용해 콘텐츠 영역을 설정할 수 있는 위젯입니다.
- ImageWidget : 콘텐츠 영역에 image 를 설정할 수 있는 위젯입니다.
- VideoWidget : 콘텐츠 영역에 video 를 설정할 수 있는 위젯입니다.
- SoundWidget : 콘텐츠 영역에 audio 를 설정할 수 있는 위젯입니다.
- YoutubeWidget : 콘텐츠 영역에 Youtube 플레이어를 설정할 수 있는 위젯입니다.

## 4. 대시보드 생성하기

제공된 샘플을 참고하여 대시보드를 생성하여 보도록 하겠습니다.

대시보드를 생성하기 위하여 필수로 필요한 자원과 위치입니다.

(아래의 예제는 작업폴더안에 아래의 경로를 그대로 만들었다고 가정한 예제입니다.)

- BIX5 JS 파일 : BIX5/JS/BIX5.js
- 라이선스 파일: LicenseKey/BIX5License.js
- 샘플 HTML 파일 : Samples/Dashboard/
- BIX5 CSS 파일 : BIX5/Assets/BIX5.css
- 대시보드 CSS 파일 : BIX5/Assets/BIX5Dashboard.css

1. HTML 의 <head>에 필요한 css 와 js 파일을 include 합니다.

```
<!-- BIX5 CSS -->
<link rel="stylesheet" type="text/css" href="../../BIX5/Assets/BIX5.css"/>
<!-- BIX5 Dashboard CSS -->
<link rel="stylesheet" type="text/css"
href="../../BIX5/Assets/BIX5Dashboard.css"/>
<!-- BIX5 라이선스 -->
<script type="text/javascript" src="../../LicenseKey/BIX5License.js"></script>
<!-- BIX5 라이브러리 -->
<script type="text/javascript" src="../../BIX5/JS/BIX5.js"></script>
```

2. <body>에 대시보드가 생성될 <div>를 생성합니다

```
<div class="content">
  <!-- 대시보드가 삽입될 DIV -->
  <div id="dashboardHolder"></div>
</div>
```

3. BIX5 제품의 Assets 경로를 설정합니다.

```
<script>
  // 대시보드 생성 전 BIX5 제품의 Assets 경로를 설정합니다.
  BIX5.setAssetsPath("../../BIX5/Assets/");
</script>
```

4. <script>에 대시보드를 생성하는 코드를 작성합니다.

```
<script>
  //대시보드를 생성합니다.
  //파라미터
  // 1. 대시보드의 id (자바스크립트의 변수 규칙을 따릅니다.)
  // 2. 대시보드가 위치할 div의 id ( 대시보드의 부모 div의 id입니다. )
  // 3. 대시보드가 생성될 준비가 되면 실행되는 함수입니다.
  //    파라미터로 Dashboard 전달됩니다.
  // 4. 대시보드의 가로 사이즈
  // 5. 대시보드의 세로 사이즈
  // 6. 대시보드 옵션 object
```

```
//6-1. enabled 대시보드 수정 여부
//6-2. Gridlines 격자선 보이기 여부.

BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){},
"1000", "600", {enabled:false, gridlines:false});

</script>
```

5. Dashboard readyHandler 에서 위젯의 메타정보를 설정합니다.

```
<script>
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    // id : 위젯의 id
    // x : 위젯의 x좌표
    // y : 위젯의 y좌표
    // width : 위젯의 너비
    // height : 위젯의 높이
    // layoutType : 위젯 레이아웃 종류
    // title : 위젯의 타이틀
    // type : 콘텐츠 영역에 설정할 타입.
    // contentUrl : html url
    // contentHtml : html string
    // options
    // border : 보더를 설정합니다.
    // borderColor : 보더의 색상을 설정합니다.
    // borderWidth : 보더의 굵기를 설정합니다.

    var slides = [{slideName:"slide1",
        widgets:[
            {id:"helloWorld", x:320,y:220,width:200,height:50,
            layoutType:"headless", type:"html",
            contentUrl:"./Widgets/Html/helloWorld.html",
            options:{border:true, borderColor:"#ccc", borderWidth:"2px"}}
        ]}
    ];

    dashboard.setData(slides);

    dashboard.setSlideWidth(850);
    dashboard.setSlideHeight(560);
}, "1000", "600", {enabled:false});

</script>
```

위젯의 메타정보로 설정 가능한 속성값은 다음과 같습니다.

위젯	속성명	type	기본값	설명
위젯 공통	id	string	random	위젯의 아이디. 자바스크립트 변수 규칙을 따릅니다.
	x	number	0	위젯의 x 좌표.
	y	number	0	위젯의 y 좌표.
	width	number	0	위젯의 너비
	height	number	0	위젯의 높이
	layoutType	string	headless	위젯 레이아웃 종류. 유효값: "window", "headless"
	title	string	""	위젯의 타이틀
	contentHtml	string	""	컨텐츠 영역에 설정할 html 문자열
	contentUrl	string	""	컨텐츠 영역에 설정할 html 문자열 url
	type	string	-	컨텐츠 영역에 설정할 타입. 유효값: "chart", "grid", "mapchart", "html", "text", "iframe", "video", "youtube", "image", "sound", "clock"
	options	object		위젯의 옵션. 자세한 사항은 아래 options 표를 참조해주세요.
	componentOptions	object		위젯 컨텐츠에 관련된 옵션. 자세한 사항은 아래 componentOptions 표를 참조해주세요.

#### options

속성명	type	기본값	설명
border	Boolean	false	위젯의 보더
borderColor	string	border가 true면 #EEEEEE border가 false면, #FFFFFF	위젯 보더의 색상
borderWidth	string	border가 true면 1px border가 false면 0px	위젯 보더의 굵기
resizable	Boolean	true	위젯의 리사이징 가능 여부0

# componentOptions

위젯	속성명	type	기본값	설명
ChartWidget	vars	string	""	차트 생성시 설정할 환경변수
GridWidget	vars	string	""	그리드 생성시 설정할 환경변수
MapChartWidget	vars	string	""	맵차트 생성시 설정할 환경변수
VideoWidget	autoplay	Boolean	false	<Video>의 자동재생 여부
	controls	Boolean	true	<Video>의 컨트롤 속성
	loop	Boolean	false	<Video>의 반복재생 속성
	muted	Boolean	false	<Video>의 음소거 속성
	poster	string	""	<Video>의 poster 속성
	preload	string		페이지가 로드될 때 비디오가 로드 되는지 여부. <video> 유효값과 같음.
	sources	array		비디오 source 배열.
	tracks	array		비디오 track 배열
SoundWidget	autoplay	Boolean	false	<audio>의 자동재생 여부
	controls	Boolean	true	<audio>의 컨트롤 속성
	loop	Boolean	false	<audio>의 반복재생 속성
	muted	Boolean	false	<audio>의 음소거 속성
	sources	array		오디오 source 배열.
TextWidget	text	string	""	TextWidget의 속성 TextWidget에 들어갈 문자열
	fontSize	string	"15px"	TextWidget의 속성 TextWidget 텍스트의 폰트의 사이즈
IFrameWidget	url	string	""	IFrameWidget의 속성 iframe의 src 속성. <iframe>으로 삽입할 특정 url
	name	string	""	IFrameWidget의 속성 iframe의 name 속성. 특정 이름.
	sandbox	string	""	IFrameWidget의 속성 iframe의 sandbox 속성. 유효값은 <iframe>의 sandbox와 동일.
	srcdoc	string	""	IFrameWidget의 속성 삽입하는 iframe에 보일 특정 HTML 콘텐츠
	frameBorder	Boolean	false	IFrameWidget의 속성 <iframe>의 보더 여부
YoutubeWidget	videoid	string	""	Youtube 비디오 아이디
	playerVars	object	{}	Youtube 플레이어 매개변수

				(자세한 사항은 Youtube IFrame Player API를 참조하세요)
ImageWidget	maintainAspectRatio	boolean	true	이미지 비율 유지 여부입니다. true면 비율 유지, false면 비율을 유지하지 않습니다.
	src	string	""	이미지 경로입니다
	alt	string	""	이미지가 없을 시 대체할 이미지 경로입니다.
	alpha	number	1	투명도입니다. 값이 작을수록 투명해집니다. 유효값 0 ~ 1

#### source

위젯	속성명	type	기본값	설명
VideoWidget	src	string	undefined	미디어 파일의 url
	type	string	undefined	리소스의 MIME-type
SoundWidget	src	string	undefined	미디어 파일의 url
	type	string	undefined	리소스의 MIME-type

#### track

위젯	속성명	type	기본값	설명
VideoWidget	src	string	undefined	트랙 파일의 url
	kind	string	undefined	텍스트 트랙 종류
	srclang	string	undefined	텍스트 트랙의 타이틀
	label	string	undefined	필수. 트랙 파일의 URL
	default	Boolean	false	기본 트랙 설정

6. 위젯에 들어갈 콘텐츠 내용과 핸들러를 작성합니다.

```
<h1>Hello World</h1>

<script>
  widget.componentReadyHandler = function() {
    console.log("widget ready");
  }
  widget.closeHandler = function() {
    return confirm("위젯을 닫으시겠습니까?");
  }
  widget.resizeHandler = function(w, h){
    console.log("resize");
  }
  widget.maximizeHandler = function(type, w, h){
    console.log(type);
  }
</script>
```

다음은 대시보드를 작성한 예제입니다.

(본 예제는 제품의 Samples/Dashboard 위치 기준으로 작성한 예제입니다.)

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Script-Type" content="text/javascript"/>
<meta http-equiv="Content-Style-Type" content="text/css"/>
<meta http-equiv="X-UA-Compatible" content="IE=edge"/>
<title>BIX5 Dashboard (BIXsoft)</title>
<!-- BIX5 에서 사용하는 스타일 -->
<link rel="stylesheet" type="text/css" href="../../BIX5/Assets/BIX5.css"/>
<link rel="stylesheet" type="text/css" href="../../BIX5/Assets/BIX5Dashboard.css"/>

<!-- IE7, 8 에서 차트 생성하고자 하는 경우 -->
<!--[if IE]><script language="javascript" type="text/javascript"
src="../../BIX5/JS/excanvas.js"></script><![endif]-->

<!-- BIX5 라이선스 -->
<script language="javascript" type="text/javascript"
src="../../LicenseKey/BIX5License.js"></script>

<!-- 실제적인 BIX5 라이브러리 -->
<script language="javascript" type="text/javascript"
src="../../BIX5/JS/BIX5.js"></script>

<script type="text/javascript">

// -----대시보드 설정 시작-----
BIX5.setAssetsPath("../../BIX5/Assets/");
// BIX5 Dashboard을 생성합니다.
// 파라미터 (순서대로)
// 대시보드 생성을 위한 파라미터는 다음과 같습니다.
// 1. (string) 대시보드 id
// 2. (string) 대시보드가 위치할 div의 id
// 3. (function) 대시보드 readyHandler
// 4. (number) 대시보드의 가로 사이즈
// 5. (number) 대시보드의 세로 사이즈
// 6. (object) options.
// 6-1. (boolean) enabled 대시보드 수정 가능 여부
// 6-2. (boolean) gridlines 그리드라인 보이기 여부
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [
        {slideName:"slide1",
        widgets:[
            {id:"title",x:10,y:0,width:250,height:40,contentUrl:"./Widgets/Html/title.html",type:"
html",title:"Untitled",options:{border:false,borderWidth:"0px",borderColor:"#ffffff"}}
            ,{id:"iframe",x:720,y:50,width:550,height:640, type:"iframe",title:"header
title",options:{border:true,borderColor:"#000",borderWidth:"1px"},
            componentOptions:{url:"http://www.bixsoft.net"}}
            ,{id:"Image_single_ratio",x:10,y:50,width:700,height:240,contentUrl:"./Widgets/Chart/I
mage_single_ratio.html",type:"chart",title:"Image_single_ratio",options:{border:true,b
orderColor:"#EEEEEE",borderWidth:"1px"}}
            ,{id:"DesignMap_World",x:10,y:300,width:700,height:190,contentUrl:"./Widgets/MapChart/
DesignMap_World.html",type:"mapchart",title:"DesignMap_World",options:{border:true,bor
derColor:"#EEEEEE",borderWidth:"1px"}}
```



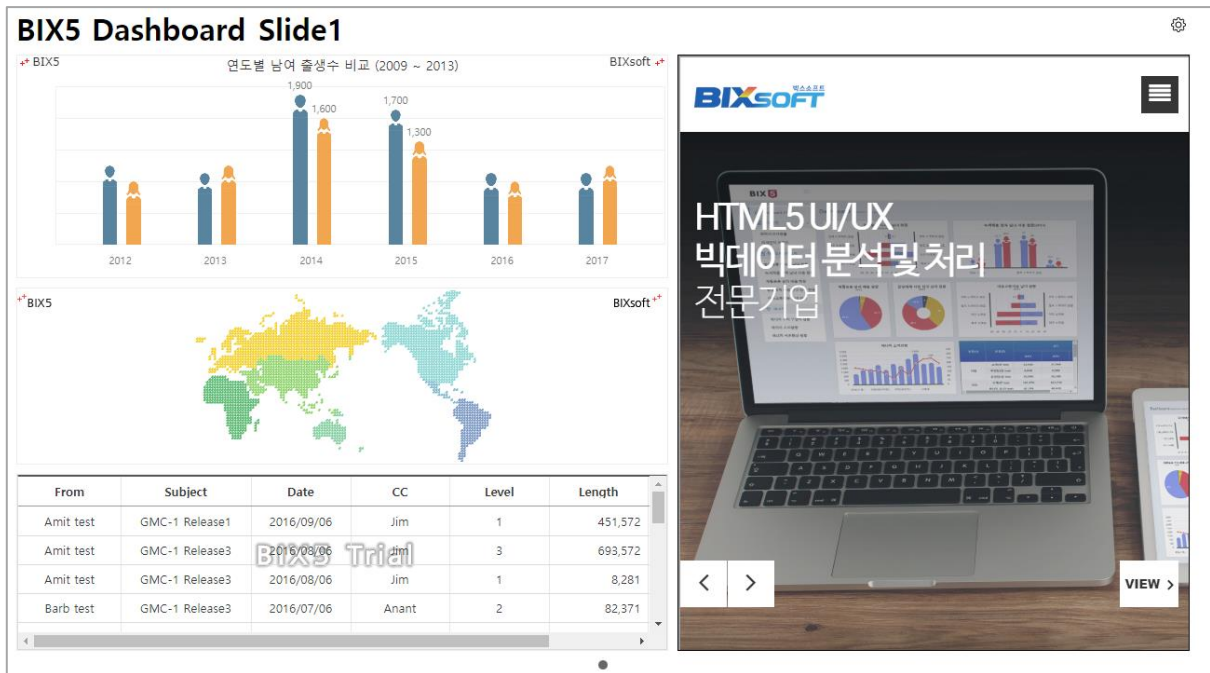
```
, {id: "BasicGrid", x: 10, y: 500, width: 700, height: 190, contentUrl: "../Widgets/Grid/BasicGrid.html", type: "grid", title: "Basic Grid", options: {border: true, borderColor: "#EEEEEE", borderWidth: "1px"}}
, {x: 270, y: 0, width: 90, height: 40, type: "html", title: "Untitled", contentHtml: "<h1>Slide1</h1>", options: {border: false}}
]
}
];

dashboard.setData(slides);
}, "1280", "720", {enabled: false, gridlines: false});

</script>
<!-- 샘플 작동을 위한 CSS -->
<link rel="stylesheet" type="text/css" href="../Web/sample.css"/>
</head>
<body>
    <div class="wrapper">
        <div class="header">
            <div class="headerTitle">대시보드 예제</div>
        </div>
        <div id="content">
            <!-- 대시보드가 삽입될 DIV -->
            <div id="dashboardHolder" style="width:1280px; height:720px;"></div>
        </div>
    </div>
</body>
</html>
```

<예제 2 HTML 페이지에 대시보드 삽입하기>

위와 같이 작성한 후 실행하면 다음과 같은 화면을 볼 수 있습니다.



<그림 12 HTML 페이지에 대시보드 삽입 성공 화면>

사용자가 웹 페이지를 작성할 때 지켜야 할 사항은 다음과 같습니다.

1. HTML 의 doctype 을 지정하십시오. (다른 doctype 이어도 무방합니다)

```
<!DOCTYPE html >
```

2. BIX5License.js 와 BIX5.js 를 포함시켜 주십시오.

```
<script language="javascript" type="text/javascript"
src=" ../LicenseKey/BIX5License.js"> </script>
<script language="javascript" type="text/javascript" src=" ../BIX5/JS/BIX5.js"> </script>
```

3. 필요에 따라 대시보드에서 사용할 이미지의 CSS 를 포함시켜 주십시오.

```
<link rel="stylesheet" type="text/css" href=" ../BIX5/Assets/BIX5.css"/>
<link rel="stylesheet" type="text/css" href=" ../BIX5/Assets/BIX5Dashboard.css"/>
```

4. 대시보드가 생성될 Div 태그를 지정하여 영역을 확보해 주십시오.

```
<div id="dashboardHolder"> </div>
```

5. 대시보드의 데이터를 작성하여 주십시오.

6. BIX5.dashboard.create() 함수 호출로 대시보드를 생성합니다.

```
// BIX5 Dashboard을 생성합니다.
// 파라미터 (순서대로)
// 대시보드 생성을 위한 파라미터는 다음과 같습니다.
// 1. (string) 대시보드 id
// 2. (string) 대시보드가 위치할 div의 id
// 3. (function) 대시보드 readyHandler
// 4. (number) 대시보드의 가로 사이즈
// 5. (number) 대시보드의 세로 사이즈
// 6. (object) options.
// 6-1. (boolean) enabled 대시보드 수정 가능 여부
// 6-2. (boolean) gridlines 슬라이드 격자선 보이기 여부

BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{slideName:"slide1",
        widgets:[ {id:"helloWorld",
            x:10,
            y:10,
            width:120,
            height:50,
            title:"Hello World",
            type: "html",
            contentUrl:"./Widgets/Html/helloWorld.html"
        }]}];

    dashboard.setData(slides);
}, "1000", "640", {enabled:true});
```

만일 html 문서 인코딩을 utf-8이 아닌 다른 문자셋을 사용할 경우에는 다음과 같이 자바스크립트를 포함시키는 문장에 charset을 utf-8로 설정해 주시기 바랍니다. (자바스크립트는 utf-8로 인코딩 되어 있음)

```
<script type="text/javascript" src="./BIX5/JS/BIX5.js" charset="utf-8"> </script>
```

## 5. 대시보드의 속성, 함수 및 이벤트

### 5.1. BIX5.dashboard

자바스크립트에서 대시보드를 생성하거나 기본값을 설정하는 컴포넌트로 대시보드 인스턴스가 생성되기 전에 만들어지는 객체입니다.

- 함수명 및 유효 값 설명.

함수명	파라미터	설명
<b>static</b> void <b>create</b> (id, target, readyHandler, width, height, options)	id : string 생성될 대시보드의 id target : string 대시보드를 생성할 div의 id readyHandler : function 대시보드가 사용할 준비가 되면 호출되는 함수 width : string 생성할 대시보드의 넓이. 예) "1000" height : string 생성할 대시보드의 높이. 예) "600" options : object 대시보드 옵션	전달된 id 를 가진 해당 target div 를 찾아 대시보드를 생성합니다. 만약 target div 가 없으면 예외가 발생합니다.  웹브라우저의 모든 구성요소에 대한 로드가 끝났을 때(DOM readyState 가 "complete"이면), target div 를 찾아 대시보드를 생성하게 됩니다.
<b>static</b> object <b>createEvent</b> (type)	type: string 생성할 이벤트명	위젯 간 이벤트 연동시 사용되는 이벤트를 생성합니다.
<b>static</b> object <b>get</b> (id)	id : string 대시보드의 id	대시보드 인스턴스를 얻습니다. 해당 인스턴스로 대시보드를 제어할 수 있습니다.
<b>static</b> void <b>resize</b> ()		각 제품에 대한 리사이즈를 실행합니다.

※ 대시보드 객체는 비동기적으로 생성하므로 대시보드가 준비된 이후에 대시보드 객체를 사용할 수 있습니다. BIX5.dashboard.create 의 readyHandler 에 등록된 콜백함수가 대시보드가 준비가 되면 호출됩니다.

● options 설명

옵션명	타입	기본값	설명
enabled	boolean	false	대시보드의 사용 여부를 설정합니다. 예를 들어 enabled : true 이면 대시보드의 위젯을 드래그, 리사이징, 삭제를 수행할 수 있습니다.
gridlines	boolean	false	대시보드의 격자선 보이기 여부를 설정합니다.
slideNavigator	boolean	true	슬라이드 네비게이터 보이기 유무를 설정합니다.
sideBarBtn	boolean	true	사이드바 버튼 보이기 유무를 설정합니다

## 5.2. Dashboard

Widget 을 생성, 관리하는 컨테이너입니다.

- 속성 및 유효 값 설명

속성명	타입	유효값(기본값)	설명
enabled	boolean	false	대시보드 편집 여부
gap	number	10	편집모드 시 위젯의 간격
selectedIndex	number	-1	현재 선택된 슬라이드 인덱스
slideHeight	number	720	슬라이드 높이
slideWidth	number	1280	슬라이드 너비

- Setter, Getter

함수명	파라미터	설명
boolean <b>getEnabled</b> ()	-	현재 대시보드 편집 여부를 불러옵니다.
number <b>getGap</b> ()	-	편집모드 시 위젯간격을 불러옵니다.
number <b>getSelectedIndex</b> ()	-	현재 선택된 슬라이드 인덱스를 불러옵니다.
number <b>getSlideHeight</b> ()	-	슬라이드 높이를 불러옵니다.
number <b>getSlideWidth</b> ()	-	슬라이드 너비를 불러옵니다.
void <b>setEnabled</b> (value)	value : boolean 편집모드 여부	대시보드 편집 여부를 설정합니다. dashboard.create의 enabled옵션보다 우선시 됩니다. ※ 파라미터 타입이 다른 경우 해당 설정을 실행하지 않습니다.
void <b>setGap</b> (value)	value : number 간격	편집모드 시 위젯간격을 설정합니다. ※ 파라미터 타입이 다른 경우 혹은 파라미터가 음수인 경우 해당 설정을 실행하지 않습니다.
void <b>setSelectedIndex</b> (value)	value : number 설정하려는 슬라이드 인덱스	현재 보여질 슬라이드 인덱스를 설정합니다.
void <b>setSlideHeight</b> (value)	value : number 슬라이드 높이	슬라이드 높이를 설정합니다.
void <b>setSlideWidth</b> (value)	value : number	슬라이드 너비를 설정합니다.

	슬라이드 너비	
--	---------	--

- 함수명 및 유효 값 설명.

함수명	파라미터	설명
void <b>addSlide</b> (slide)	slide : object 슬라이드 정보를 가진 객체	대시보드에 슬라이드를 추가 기존에 슬라이드의 끝에 추가합니다.
void <b>addSlideAt</b> (slide, index)	slide : object 슬라이드 정보를 가진 객체  Index : number 슬라이드를 추가할 인덱스 위치	대시보드에 슬라이드를 해당 인덱스에 추가합니다.
void <b>addSlides</b> (slides)	slides : array 슬라이드 정보를 가진 객체의 배열	대시보드에 여러 장의 슬라이드를 추가합니다.
object <b>getData</b> ()		대시보드의 현재 설정된 전반적인 데이터를 얻습니다.
number <b>getNumSlides</b> ()		대시보드의 슬라이드 개수를 얻습니다.
object <b>getSelectedSlide</b> ()		현재 선택된 슬라이드를 얻습니다.
object <b>getSideBar</b> ()		사이드바 객체를 얻습니다.
object <b>getSlideAt</b> (index)	index : number 슬라이드 인덱스	인덱스에 해당하는 슬라이드를 얻습니다.
array <b>getTemplateData</b> ()		대시보드의 위젯 레이아웃 데이터(x, y, width, height)를 얻습니다.
array <b>getWidgetPool</b> ()	id : string 위젯 아이디	사이드바에 설정한 위젯풀(사용자가 만든 위젯들)을 얻습니다.
void <b>moveFirst</b> ()		첫번째 슬라이드로 이동합니다.
void <b>moveLast</b> ()		마지막 슬라이드로 이동합니다.
void <b>moveNext</b> ()		다음 슬라이드로 이동합니다.
void <b>movePrev</b> ()		이전 슬라이드로 이동합니다.
void <b>removeSlideAt</b> (index)	index : number 슬라이드 인덱스	인덱스에 해당하는 슬라이드를 삭제합니다.
void <b>setBorder</b> (borderWidth, borderStyle, borderColor)	borderWidth : string border 굵기 borderStyle : string border 스타일	대시보드의 테두리를 설정합니 다.

	borderColor : string border 색상	
void <b>setData</b> (slides)	slides : array 대시보드 설정을 위한 대시보드 메타데이터	대시보드 설정
void <b>setSlides</b> (slides)	slides: array 슬라이드 데이터	슬라이드를 설정합니다.
void <b>setWidgetPool</b> (widgets)	widgets : array 위젯풀에 설정할 위젯리스트	사이드바의 위젯풀을 설정합니다.
void <b>showGridLine</b> (val)	val : boolean 격자선 보이기 유무	격자선 보이기 유무를 설정합니다.
void <b>showSideBarBtn</b> (val)	val : boolean 사이드바 버튼 보이기 유무	사이드바 버튼 보이기 유무를 설정합니다.
void <b>showSlideNavigator</b> (val)	val : boolean 슬라이드 네비게이터 보이기 유무	슬라이드 네비게이터 보이기 유무를 설정합니다.

※ 파라미터 타입이 다른 경우 해당 설정을 실행하지 않습니다.



### 5.3. Slide

- 함수명 및 유효값 설명.

함수명	파라미터	설명
void <b>addWidget</b> (widget)	widget : object 생성할 위젯 데이터	슬라이드에 위젯을 추가합니다.
void <b>addWidgetes</b> (widgets)	widgets : array 생성할 위젯 데이터 배열	슬라이드에 복수의 위젯을 추가합니다.
array <b>getData</b> ()		슬라이드의 위젯데이터를 얻습니다.
object <b>getWidget</b> (widgetId)	widgetId : string 인스턴스로 얻으려는 위젯 아이디	위젯 아이디로 해당 위젯을 얻습니다.
array <b>getWidgetAll</b> ()		해당 슬라이드의 모든 위젯을 배열로 얻습니다.
void <b>removeWidget</b> (widgetId)	widgetId : string 삭제하려는 위젯 아이디	위젯 아이디로 해당 위젯을 삭제합니다.
void <b>removeWidgetAll</b> ()		슬라이드의 모든 위젯을 삭제합니다
void <b>setData</b> ()		슬라이드에 위젯데이터를 설정합니다
void <b>reload</b> ()		슬라이드를 새로고침합니다.

※ 파라미터 타입이 다른 경우 해당 설정을 실행하지 않습니다.

## 5.4. Widget

- 속성 및 유효 값 설명

속성 명	타입	유효값(기본값)	설명
width	number	0	위젯의 너비
height	number	0	위젯의 높이
x	number	0	위젯의 x 좌표
y	number	0	위젯의 y 좌표
minHeight	number	10	위젯 최소 높이
minWidth	number	10	위젯 최소 너비

- Setter, Getter

함수명	파라미터	설명
number <b>getHeight()</b>	-	위젯의 높이를 불러옵니다.
number <b>getMinHeight()</b>	-	위젯의 최소 높이를 불러옵니다.
number <b>getMinWidth()</b>	-	위젯의 최소 너비를 불러옵니다.
number <b>getWidth()</b>	-	위젯의 너비를 불러옵니다.
number <b>getX()</b>	-	위젯의 x 좌표를 불러옵니다.
number <b>getY()</b>	-	위젯의 y 좌표를 불러옵니다.
void <b>setHeight(value)</b>	value : number 위젯의 높이	위젯의 높이를 설정합니다. 설정값이 위젯의 최소 높이보다 작아도 설정합니다. ※ 파라미터타입이 다르거나 음수이면 스킵합니다.
void <b>setMinHeight(value)</b>	value : number 위젯의 최소 높이	UI로 리사이징 할 때 위젯의 최소 높이를 설정합니다. ※ 파라미터타입이 다르거나 음수이면 해당 설정을 실행하지 않습니다.
void <b>setMinWidth(value)</b>	value : number 위젯의 최소 너비	UI로 리사이징 할 때 위젯의 최소 너비를 설정합니다. ※ 파라미터타입이 다르거나 음수이면 스킵합니다.
void <b>setWidth(value)</b>	value : number 위젯의 너비	위젯의 너비를 설정합니다. 설정값이 위젯의 최소 너비보다 작아도 설정합니다. ※ 파라미터타입이 다르거나 음수이면 해당 설정을 실행하지 않습니다.
void <b>setX(value)</b>	value : number 위젯의 x 좌표	위젯의 x 좌표를 설정합니다.

void <b>setY</b> (value)	value : number 위젯의 y 좌표	위젯의 y 좌표를 설정합니다.
--------------------------	----------------------------	------------------

※ 파라미터 타입이 다른 경우 해당 설정을 실행하지 않습니다.

- 함수명 및 유효값 설명.

함수명	파라미터	설명
void <b>destroy</b> ()		widget 삭제시 이벤트 리스너 제거 및 초기화하는 함수입니다.
object <b>getSlide</b> ()		해당 위젯이 속해있는 슬라이드의 인스턴스를 얻습니다.
void <b>setBorderColor</b> (rgb)	rgb : string rgb 또는 hex	위젯의 Border 색상을 설정합니다
void <b>setBorderWidth</b> (border)	border : string 굵기(px단위)	위젯의 Border 굵기를 설정합니다
void <b>showMaximized</b> ()		위젯의 크기를 데스크톱 사이즈로 확장합니다.
void <b>showNormal</b> ()		위젯의 크기를 이전 크기로 되돌립니다.
void <b>reload</b> ()		위젯을 새로고침 합니다.

※ 파라미터 타입이 다른 경우 해당 설정을 실행하지 않습니다.

- 이벤트 설명

이벤트 핸들러명	파라미터	설명
void <b>activeHandler</b> ()	-	위젯이 소속된 슬라이드가 활성화되면 호출됩니다.
boolean <b>closeHandler</b> ()		위젯의 닫기 버튼 클릭시 호출됩니다. true가 리턴되면 위젯이 닫힙니다. 예) widget.closeHandler(){ ... }
void <b>componentReadyHandler</b> (componentId)	componentId : string 컴포넌트 아이디	component를 사용할 준비가 되면 호출합니다.
void <b>componentRemoveHandler</b>	-	component가 삭제되기 전 호출됩니다.
void <b>contentReadyHandler</b>	-	content를 사용할 준비가 되면 호출합니다.
void <b>contentRemoveHandler</b>	-	content가 삭제되기 전 호출됩니다.

void <b>deactiveHandler</b> ()	-	위젯이 소속된 슬라이드가 비활성화되면 호출됩니다.
void <b>maximizeHandler</b> (type, contentWidth, contentHeight)	type : string 이벤트 종류. 유효값 : "maximize", "normal"  contentWidth : number content영역의 너비  contentHeight : number content영역의 높이	위젯의 최대화 버튼 클릭시 호출됩니다. 예) widget.maximizeHandler(type, contentWidth, contentHeight){ ... }
void <b>resizeHandler</b> (width, height)	width : number 위젯의 너비 height : number 위젯의 높이	위젯의 리사이징시 호출됩니다. 예) widget.resizeHandler(width, height){ ... }

※ 이벤트 사용 시 'widget'은 정해진 명칭이기 때문에 변경이 불가능합니다.

## 5.5. SideBar

- 함수명 및 유효값 설명.

함수명	파라미터	설명
void <b>setSlideLayouts</b> (arr)	arr : 정해진 슬라이드 레이아웃 데이터	사이드바의 슬라이드 레이아웃 리스트를 설정합니다.

※ 슬라이드 레이아웃, 위젯 템플릿 설정 방법

- /BIX5/JS/slidelayouts.js

위 자바스크립트 파일을 <head> 태그 안에 삽입하시면 됩니다.

위젯 템플릿은 생성 시점 문제로 인하여 defer 옵션을 추가해야 합니다.

```
<html>
<head>
    ...
    <!--슬라이드 레이아웃 삽입-->
    <script src="/BIX5/JS/slidelayouts.js" language="javascript"></script>

    <!-- 사용자 정의 설정 시작 -->
    <script language="JavaScript" type="text/javascript">
    BIX5.dashboard.create("dashboard1", "dashboardHolder", function (dashboard) {
        dashboard.getSideBar().setSlideLayouts(slidelayouts);
    }, "100%", "100%");
    ...
</script>
```

<예제 3 BIX5 슬라이드 삽입 예제>

※ 슬라이드 레이아웃에 대한 설명

```
예시
var slidelayouts = [
{
    "slideName": "2단(1+3) 레이아웃",
    "thumbnail": "thumbnail/layout_13.png",
    "meta": {
        "cols": 3,
        "rows": 2,
        "items": [
            { "x": 0, "y": 0, "w": 3, "h": 1 },
            { "x": 0, "y": 1, "w": 1, "h": 1 },
            { "x": 1, "y": 1, "w": 1, "h": 1 },
            { "x": 2, "y": 1, "w": 1, "h": 1 }
        ]
    }
}]
```

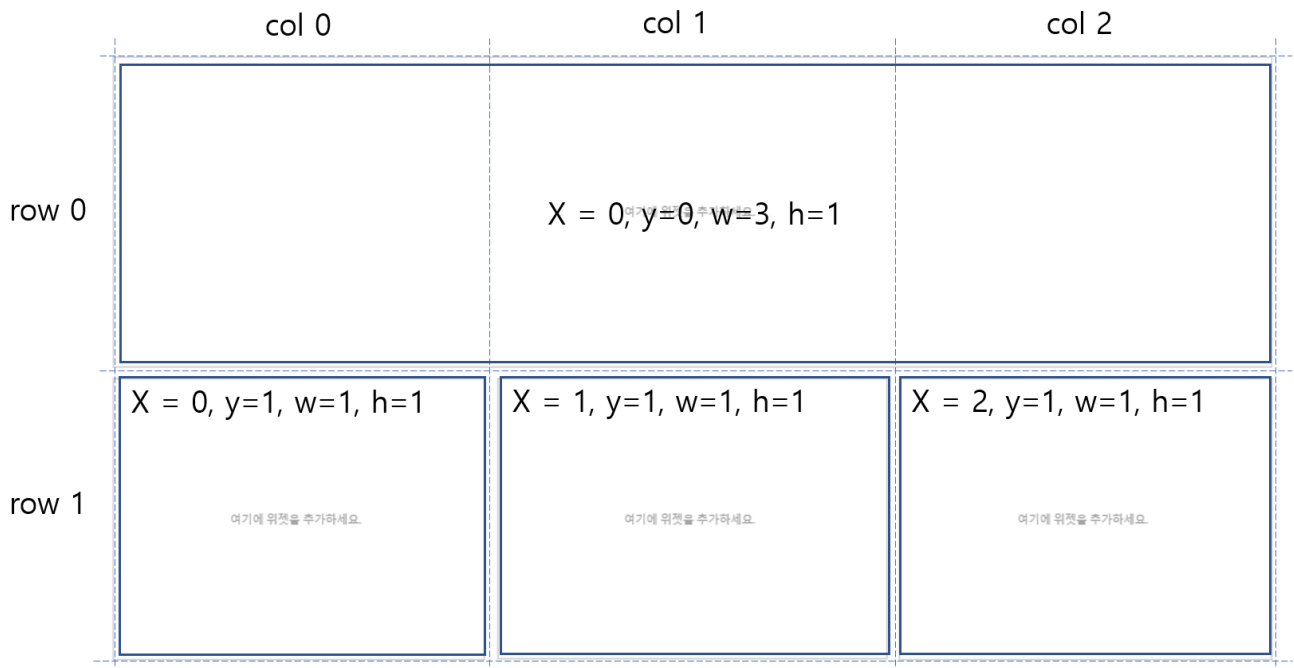
```

    }
  },
  ...
];

```

#### 프로퍼티 설명

프로퍼티명		설명
slideName		슬라이드 레이아웃에 노출되는 이름
thumbnail		슬라이드 레이아웃 목록에 마우스 오버시 나오는 이미지.
meta		슬라이드 레이아웃에 관한 정보입니다.
	cols	열의 개수. 해당 값으로 열을 생성합니다.
	rows	행의 개수. 해당 값으로 행을 생성합니다.
	items	아이템에 대한 데이터입니다. x : 아이템의 x좌표. col 기준입니다. y : 아이템의 y좌표. row 기준입니다. w : 아이템의 너비. col칸 기준입니다. h : 아이템의 높이. row칸 기준입니다.



<그림 13 slidelayouts 에 대한 설명>

## 6. 위젯 설정 방법

### 6.1. Text

1. 위젯 설정 배열의 type 으로 "text"를 입력합니다. Id 는 옵션입니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function (dashboard){
    var slides = [{slideName : "slide1",
                    widgets:[{
                        x:10,
                        y:10,
                        width:250,
                        height:30,
                        type:"text",
                        text: "BIX5 Dashboard",
                        fontSize: "20px"}]
                    }];
}, "1280", "720", {enabled:true});
```

<예제 4 위젯 설정 배열 작성>

2. 대시보드에 위젯을 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function (dashboard){
    var slides = [{...}];

    dashboard.setSlides(slides);
}, "1280", "720", {enabled:true});
```

<예제 5 위젯 설정>

## 6.2. Chart

1. 위젯 컨텐트로 설정하고자 하는 차트 레이아웃과 데이터를 생성합니다.  
(차트 생성 관련하여 상세한 내용은 BIX5 차트의 사용자 매뉴얼을 참고 바랍니다.  
위치 : 제품의 Docs/Chart/PDF)

차트 레이아웃은 다음과 같습니다.

```
<BIX5 backgroundColor="#FFFFFF" borderStyle="none">
  <Options>
    <Caption text="World Top 10 - Fastest Growing Economies (2017)" />
    <SubCaption text="GDP Growth (In %)" textAlign="center" />
  </Options>
  <Column2DChart showDataTips="true"
itemClickJsFunction="@widget.chartItemClickHandler" >
    <horizontalAxis>
      <CategoryAxis categoryField="Country"/>
    </horizontalAxis>
    <verticalAxis>
      <LinearAxis maximum="100" interval="10"/>
    </verticalAxis>
    <series>
      <Column2DSeries labelPosition="outside" yField="GDP" displayName="GDP
Growth (In %)">
      </Column2DSeries>
    </series>
  </Column2DChart>
</BIX5>
```

<예제 6 Samples/Dashboard/Widgets/Chart/layout/Column\_2D.xml>

차트이벤트는 레이아웃 작성시 @widget.이벤트핸들러명으로 작성합니다.

```
<Column2DChart showDataTips="true" itemClickJsFunction="@widget.chartItemClickHandler">
```

<예제 7 차트 이벤트 핸들러 설정>



차트 데이터는 다음과 같습니다.

```
[{
  "Country" : "South<br>Sudan",
  "GDP" : 20
},{
  "Country" : "Libya",
  "GDP" : 30
},{
  "Country" : "Sierra<br>Leone",
  "GDP" : 51.2
},{
  "Country" : "Mongolia",
  "GDP" : 44.5
},{
  "Country" : "Paraguay",
  "GDP" : 62.35
},{
  "Country" : "Timor<br>Leste",
  "GDP" : 84.46
},{
  "Country" : "Iraq",
  "GDP" : 48.9
},{
  "Country" : "Panama",
  "GDP" : 38
}]
```

<예제 8 Samples/Dashboard/Widgets/Chart/data/Column\_2D.json>

2. widget 객체에 componentReadyHandler(위젯 컴포넌트 사용 준비가 완료된 이후 호출) 메소드를 정의합니다. widget은 시스템 내부에서 생성되는 객체입니다. 이벤트 핸들러 사용을 원하신다면 아래와 같이 사용하면 됩니다.

(위젯은 비동기적으로 생성하므로 위젯 컴포넌트 사용 준비가 완료된 이후에 위젯 객체를 사용할 수 있습니다.)

```
<script type="text/javascript">
  widget.componentReadyHandler = function(componentId){}
</script>
```

<예제 9 Samples/Dashboard/Widgets/Chart/Column\_2D.html>

3. componentReadyHandler 내부에서 차트 layout 과 차트 Data 의 URL 을 설정합니다.

```
<script type="text/javascript">
  widget.componentReadyHandler = function(componentId){
    document.getElementById(componentId).setLayoutURL("./Widgets/Chart/layout/Column_2D.xml");
    document.getElementById(componentId).setDataType("json");
    document.getElementById(componentId).setDataURL("./Widgets/Chart/data/Column_2D.json");
  }
</script>
```

<예제 10 Column\_2D.html componentReadyHandler 작성>

4. widget 객체에 차트 레이아웃에서 설정한 이벤트 핸들러를 정의합니다.

```
<script type="text/javascript">

    widget.chartItemClickHandler = function(seriesId, displayText, index, data,
        values){ ... }

</script>
```

<예제 11 Column\_2D.html componentReadyHandler>

5. 위젯 설정 배열에 type 과 contentUrl 을 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{slideName : "slide1",
        widgets:[{id:"Column_2D",
            x:10,
            y:10,
            width:500,
            height:290,
            title:"Column 2D",
            type:"chart",
            contentUrl:"./Widgets/Chart/Column_2D.html"}]}
    ]};
}, "1280", "720", {enabled:true});
```

<예제 12 위젯 설정 배열 작성>

6. 대시보드에 슬라이드를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{...}];
    dashboard.setSlides(slides);
}, "1280", "720", {enabled:true});
```

<예제 13 위젯 설정>

### 6.3. Grid

1. 위젯 콘텐츠로 설정하고자 하는 그리드 레이아웃과 데이터를 생성합니다.

(그리드 생성 관련하여 상세한 내용은 BIX5 그리드의 사용자 매뉴얼을 참고 바랍니다.)

위치 : 제품의 Docs/Grid/PDF)

그리드 레이아웃은 다음과 같습니다.

```
<?xml version="1.0" encoding="utf-8"?>
  <BIX5>
    <NumberFormatter id="numfmt" useThousandsSeparator="true"/>
    <DataGrid id="dg1" horizontalScrollPolicy="auto" verticalAlign="middle"
      textAlign="center">
      <columns>
        <DataGridColumn dataField="From" width="110"
labelJsFunction="@widget.convertFrom"/>
        <DataGridColumn dataField="Subject" width="140"/>
        <DataGridColumn dataField="ReceiveDate" headerText="Date"/>
        <DataGridColumn dataField="CC"/>
        <DataGridColumn dataField="Level"/>
        <DataGridColumn dataField="Length" textAlign="right"
formatter="{numfmt}"/>
        <DataGridColumn dataField="AttachCount" textAlign="right"/>
      </columns>
    </DataGrid>
  </BIX5>
```

<예제 14 Samples/Dashboard/Widgets/Grid/layout/BasicGrid.xml>

그리드 이벤트는 레이아웃 작성시 @widget.이벤트핸들러명으로 작성합니다.

```
<DataGridColumn dataField="From" width="110" labelJsFunction="@widget.convertFrom"/>
```

<예제 15 그리드 이벤트 핸들러 설정 >

그리드 데이터는 다음과 같습니다.

```
[{"From":"Amit", "Subject":"GMC-1 Release1", "ReceiveDate":"2016/09/06",
"SendDate":"2016/09/06", "CC":"Jim", "Length":451572, "Level":1, "AttachCount":0},
{"From":"Amit", "Subject":"GMC-1 Release3", "ReceiveDate":"2016/08/06",
"SendDate":"2016/08/05", "CC":"Jim", "Length":693572, "Level":3, "AttachCount":1},
{"From":"Amit", "Subject":"GMC-1 Release3", "ReceiveDate":"2016/08/06",
"SendDate":"2016/08/06", "CC":"Jim", "Length":8281, "Level":1, "AttachCount":12},
{"From":"Barb", "Subject":"GMC-1 Release3", "ReceiveDate":"2016/07/06",
"SendDate":"2016/07/06", "CC":"Anant", "Length":82371, "Level":2,
"AttachCount":3},
{"From":"Barb", "Subject":"GMC-1 Release4", "ReceiveDate":"2016/08/06",
"SendDate":"2016/08/05", "CC":"Anant", "Length":9286, "Level":5, "AttachCount":6}]
```

<예제 16 Samples/Dashboard/Widgets/Grid/data/BasicGrid.json>

2. widget 객체에 componentReadyHandler(위젯 컴포넌트 사용 준비가 완료된 이후 호출) 메소드를 정의합니다. widget 은 시스템 내부에서 생성되는 객체입니다. 이벤트 핸들러 사용을 원하신다면 아래와 같이 사용하면 됩니다.

(위젯은 비동기적으로 생성하므로 위젯 컴포넌트 사용 준비가 완료된 이후에 위젯 객체를 사용할 수 있습니다.)

```
<script type="text/javascript">
    widget.componentReadyHandler = function(componentId){}
</script>
```

<예제/ 17 Samples/Dashboard/Widgets/Grid/BasicGrid.html>

3. componentReadyHandler 내부에서 그리드 layout 과 그리드 Data 의 URL 을 설정합니다.

```
<script type="text/javascript">
    widget.componentReadyHandler = function(componentId){
        document.getElementById(componentId).setLayoutURLEx("./Widgets/Grid/
        layout/basicGrid.xml");
        document.getElementById(componentId).setDataType("json");
        document.getElementById(componentId).setDataURLEx("./Widgets/Grid/da
        ta/basicGrid.json");
    }
</script>
```

<예제/ 18 BasicGrid.html componentReadyHandler 작성>

4. widget 객체에 그리드 레이아웃에서 설정한 이벤트 핸들러를 정의합니다.

```
<script type="text/javascript">
    widget.convertFrom = function(item, value, column){ ... }
</script>
```

<예제/ 19 BasicGrid.html convertFrom handler>

5. 위젯 설정 배열에 type 과 contentUrl 을 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{slideName:"slide1"
        , widgets: [{id:" BasicGrid",
            x:10,
            y:10,
            width:980,
            height:290,
            title:"Grid",
            type:"grid",
            contentUrl:"./Widgets/Grid/ BasicGrid.html"}]};
    }, "1280", "720", {enabled:true});
```

<예제/ 20 위젯 설정 배열 작성>

6. 대시보드에 슬라이드를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function (dashboard) {  
    var slides = [{...}];  
    dashboard.setData(slides);  
}, "1280", "720", {enabled:true});
```

<예제 21 위젯 설정>

## 6.4. MapChart

1. 위젯 컨텐트로 설정하고자 하는 맵차트 레이아웃을 생성합니다.

(맵차트 생성 관련하여 상세한 내용은 BIX5 맵차트의 사용자 매뉴얼을 참고 바랍니다.)

위치 : 제품의 Docs/MapChart/PDF)

맵차트 레이아웃은 다음과 같습니다.

```
<?xml version="1.0" encoding="utf-8"?>
<BIX5>
  <MapChart id="mainMap1" showDataTips="true" dataTipType="Type3">
    <series>
      <MapSeries id="mapseries" interactive="true" selectionMarking="Line"
        color="#555555" labelPosition="none" displayName="Map" rollOverFill="transparent"
        transparentValue="0" selectionStrokeWeight="1" selectionStrokeSize="small"
        localFillJsFunction="@widget.colorFunction">
        <stroke>
          <Stroke color="#FFFFFF" weight="0.5" alpha="1"/>
        </stroke>
        <rollOverStroke>
          <Stroke color="#FFFFFF" weight="0.5" alpha="1"/>
        </rollOverStroke>
      </MapSeries>
    </series>
  </MapChart>
</BIX5>
```

<예제 22 worldCountry.xml>

맵차트 이벤트는 레이아웃 작성시 @widget.이벤트핸들러명으로 작성합니다.

```
<MapSeries id="mapseries" localFillJsFunction="@widget.colorFunction">
```

<예제 23 맵차트 이벤트 핸들러 설정>

2. widget 객체에 componentReadyHandler(위젯 컴포넌트 사용 준비가 완료된 이후 호출) 메소드를 정의합니다. widget은 시스템 내부에서 생성되는 객체입니다. 이벤트 핸들러 사용을 원하신다면 아래와 같이 사용하면 됩니다.

(위젯은 비동기적으로 생성하므로 위젯 컴포넌트 사용 준비가 완료된 이후에 위젯 객체를 사용할 수 있습니다.)

```
<script type="text/javascript">
  widget.componentReadyHandler = function(componentId){}
</script>
```

<예제 24 WorldCountry.html componentReadyHandler>

3. componentReadyHandler 내부에서 맵차트 layout 과 맵차트 Data 의 URL 을 설정합니다.

```
<script type="text/javascript">

widget.componentReadyHandler = function(componentId){
    var mapDataBaseURL = "../MapChart/MapDataBaseXml/WorldCountry.xml";
    var sourceURL = "../MapChart/MapSource/WorldCountry.svg";
    var layoutURL = "../Widgets/MapChart/layout/worldCountry.xml";

    document.getElementById(componentId).setLayoutURLEx(layoutURL);
    document.getElementById(componentId).setMapDataBaseURLEx(mapDataBaseURL);
    document.getElementById(componentId).setSourceURLEx(sourceURL);
}
};
</script>
```

<예제 25 WorldCountry.html componentReadyHandler 작성>

4. widget 객체에 맵차트 레이아웃에서 설정한 이벤트 핸들러를 정의합니다.

```
<script type="text/javascript">

    widget.colorFunction = function(code, label, data){ ... }

</script>
```

<예제 26 WorldCountry.html colorFunction handler>

5. 위젯 설정 배열에 type 과 contentUrl 을 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{slideName:"slide1"
        , widgets : [{id:" WorldCountry",
            x:10,
            y:10,
            width:980,
            height:290,
            title:" World Country",
            type:"mapchart",
            contentUrl:"../Widgets/MapChart/WorldCountry.html"}]
        }]
    }, "1280", "720", {enabled:true});
```

<예제 27 위젯 설정 배열 작성>

6. 대시보드에 슬라이드를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{...}];
    dashboard.setData(slides);
    }, "1280", "720", {enabled:true});
```

<예제 28 위젯 설정>

## 6.5. Html

1. 위젯 콘텐츠로 설정할 html 과 필요에 따라 핸들러를 작성합니다.

```
<h1>Hello World</h1>

<script>
widget.componentReadyHandler = function() {
    console.log("widget ready");
}

widget.closeHandler = function() {
    return confirm("close");
}

widget.resizeHandler = function(w, h){
    console.log("resize");
}

widget.maximizeHandler = function(type, w, h){
    console.log("maximize");
}
</script>
```

<예제 29 helloWorld.html>

2. 위젯 설정 배열에 type 과 content 를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{slideName:"slide1"
        , widgets : [{id:"helloWorld",
            x:10,
            y:10,
            width:500,
            height:290,
            title:"hello world",
            type:"html",
            contentUrl: "../Widgets/Html/helloWorld.html"}]
    }], "1280", "720",{enabled:true});
```

<예제 30 위젯 설정 배열 작성>

3. 대시보드에 슬라이드를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [...];
    dashboard.setData(slides);

}, "1280", "720", {enabled:true});
```

<예제 31 위젯 설정>



## 6.6. Iframe

1. 위젯 설정 배열의 type 으로 "iframe"을 입력합니다. Id 는 옵션입니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function (dashboard){
    var slides = [{slideName : "slide1",
        widgets:[{
            x:10,
            y:10,
            width:710,
            height:670,
            type:"iframe",
            componentOptions:{url: "http://www.bixsoft.net"}}]}
    ], "1280", "720", {enabled:true});
```

<예제 32 위젯 설정 배열 작성>

2. 대시보드에 슬라이드를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function (dashboard){
    var slides = [{...}];
    dashboard.setData(slides);
    }, "1280", "720", {enabled:true});
```

<예제 33 위젯 설정>

## 6.7. Video

1. 슬라이드 데이터에 위젯의 위치, 크기정보 및 비디오 정보를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{slideName : "slide1",
                    widgets:[{
                        x:10,
                        y:10,
                        width:640,
                        height:360,
                        type:"video",
                        componentOptions:{autoplay:true,
                                         controls:true,
                                         loop:true,
                                         muted:true,
                                         sources:[
                                             {src:"../BIX5/Assets/Video/Blue.mp4",
                                              type:"video/mp4"}
                                         ]
                                         }
                    }]}];
}, "1280", "720", {enabled:true});
```

<예제 34 위젯 설정 배열 작성2>

2. 대시보드에 슬라이드를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{...}];

    dashboard.setData(slides);
}, "1280", "720", {enabled:true});
```

<예제 35 위젯 설정>

## 6.8. Youtube

1. Youtube IFrame API 를 사용을 위해 <head>에 다음 스크립트를 넣어줍니다.

```
<head>
  <script src="https://www.youtube.com/iframe_api"></script>
  ...
</head>
```

<예제 36 위젯 설정 배열 작성>

2. 슬라이드 데이터에 위젯의 위치, 크기정보 및 비디오 정보를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
  var slides = [{slideName : "slide1",
    widgets:[{
      x:10,
      y:10,
      width:710,
      height:670,
      type:"youtube",
      componentOptions:{
        videoId: "U_sYIKWhJvk",
        playerVars:{autoplay:1}
      }
    }]}
  ]};
}, "1280", "720", {enabled:true});
```

<예제 37 위젯 설정 배열 작성>

3. 대시보드에 슬라이드를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
  var slides = [{...}];

  dashboard.setData(slides);
}, "1280", "720", {enabled:true});
```

<예제 38 위젯 설정>

## 6.9. Clock

1. 슬라이드 데이터에 위젯의 위치, 크기정보 및 타입을 clock 으로 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function (dashboard){  
    var slides = [{slideName: "slide1",  
                    widgets: [{  
                        x:590,  
                        y:10,  
                        width:250,  
                        height:30,  
                        type:"clock"  
                    }]  
                }];  
}, "1280", "720", {enabled:true});
```

<예제 39 위젯 설정 배열 작성>

2. 대시보드에 슬라이드를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function (dashboard){  
    var slides = [{...}];  
  
    dashboard.setData(slides);  
}, "1280", "720", {enabled:true});
```

<예제 40 위젯 설정>

## 6.10. Image

1. 슬라이드 데이터에 위젯의 위치, 크기정보 및 이미지 정보를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{slideName : "slide1",
        widgets: [{
            x:590,
            y:10,
            width:250,
            height:30,
            type:"image",
            componentOptions: {
                maintainAspectRatio: false,
                src:"./Web/Images/bix5_log.png",
                alt:"BIX5 Logo",
                alpha:1
            }
        }]}
    ]};
}, "1280", "720", {enabled:true});
```

<예제 41 위젯 설정 배열 작성>

2. 대시보드에 슬라이드를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{...}];

    dashboard.setData(slides);
}, "1280", "720", {enabled:true});
```

<예제 42 위젯 설정>

## 6.11. Sound

1. 슬라이드 데이터에 위젯의 위치, 크기정보 및 사운드 정보를 설정합니다.

```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{slideName : "slide1",
        widgets: [{
            x:590,
            y:10,
            width:250,
            height:30,
            type:"sound",
            componentOptions: {
                autoplay: false,
                controls: true,
                loop: false,
                muted: false,
                sources: [
                    {src: "이곳에 오디오 파일 경로를 작성",
                     type: "audio/mpeg"
                    }
                ]
            }
        }]}
    ]};
}, "1280", "720", {enabled:true});
```

<예제 43 위젯 설정 배열 작성>

2. 대시보드에 슬라이드를 설정합니다.

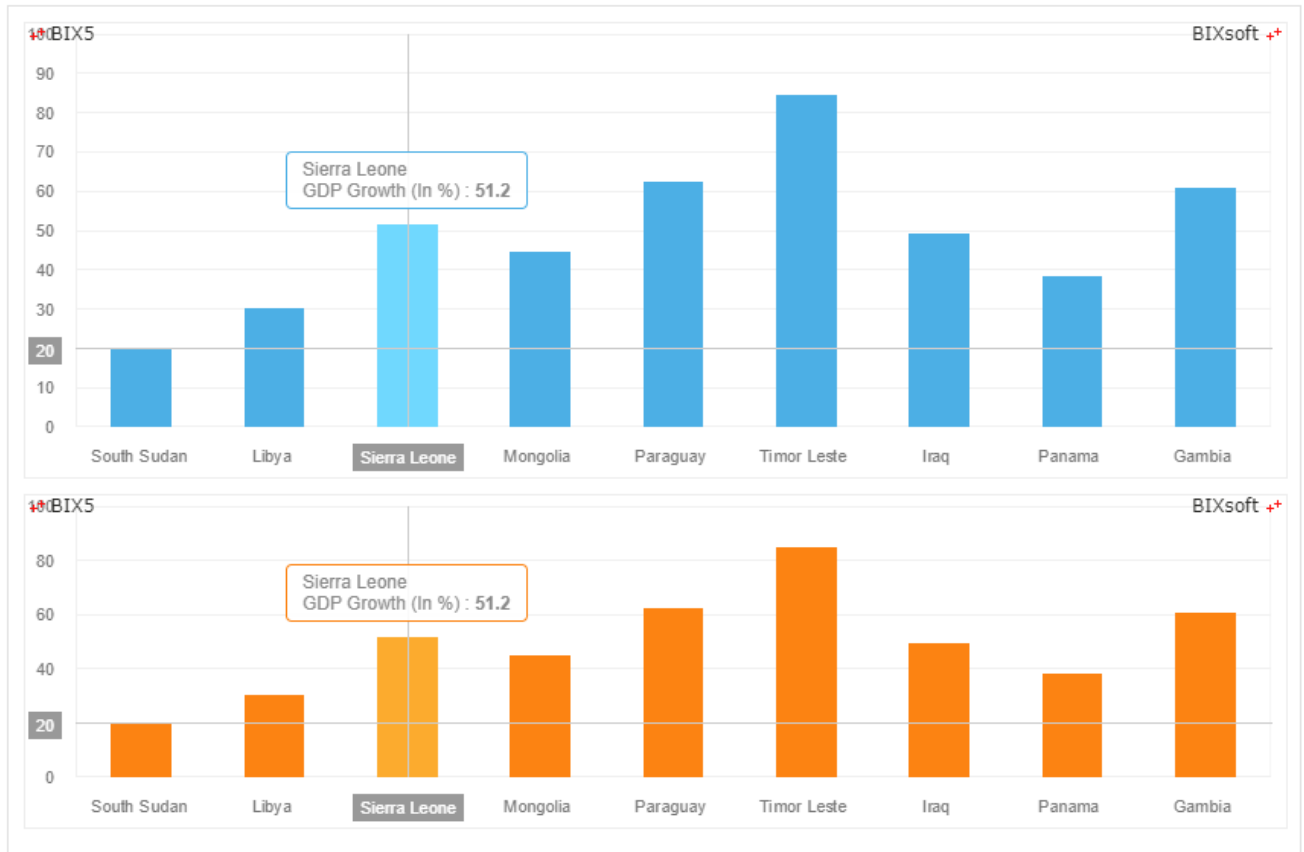
```
BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
    var slides = [{...}];

    dashboard.setData(slides);
}, "1280", "720", {enabled:true});
```

<예제 44 위젯 설정>

## 7. 위젯 간 연동 방법

위젯 간 연동은 BIX5.dashboard.createEvent 함수로 사용자가 이벤트를 생성하여 위젯 간 메시지를 주고받을 수 있습니다. 아래의 예제는 다른 차트에 십자선 위치가 동기화 되는 예제입니다.



<그림 14 위젯 간 연동>

다음은 위젯 간 연동을 설명하기 위한 예제 소스입니다. 아래는 대시보드를 생성하는 소스의 일부입니다. 차트 위젯이 2 개가 있고 [interaction\\_chart.html](#), [interaction\\_chart2.html](#) 이 `contentUrl` 로 각 위젯에 설정되어 있습니다.

widget\_interaction.html

```
<!DOCTYPE html>
<html>
<head>
...
<script>
    BIX5.dashboard.create("dashboard1", "dashboardHolder", function(dashboard){
        var slides = [{slideName:"slide1"
                        , widgets:[

                            {id:"w01",x:10,y:10,width:830,height:300,type:"chart",
                              contentUrl :"/Widgets/Chart/interaction_chart.html", options:{border:true}},

                            {id:"w02",x:10,y:320,width:830,height:220,type:"chart",
                              contentUrl :"/Widgets/Chart/interaction_chart2.html", options:{border:true}}
```

```

    }
  ];

  ...
}, "100%", "100%", {enabled:false, gridlines:false});
</script>
<body>
  ...
</body>
</html>

```

※ 옵션에 enabled 를 false 로 줘야 이벤트 연동 결과를 확인하실 수 있습니다

다음은 각 위젯 모듈의 소스를 통해 위젯간 이벤트를 주고받는 방식을 알아보겠습니다. 순서는 아래와 같습니다.

- ① 메시지를 보낼 위젯에서 BIX5.dashboard.createEvent 함수로 이벤트를 생성합니다.
- ② 앞서 생성한 이벤트의 detail 이라는 속성을 통해 원하는 데이터를 전달할 수 있습니다.  
detail 의 타입은 object 입니다.
- ③ 메시지를 보낼 위젯에서 슬라이드의 dispatchEvent 함수를 통해 이벤트를 송출합니다.
- ④ 메시지를 받을 위젯에서 해당 이벤트 리스너를 구현합니다.
- ⑤ 메시지를 받을 위젯에서 슬라이드의 addEventListener 함수를 통해 ④의 이벤트 리스너를 등록합니다. 해당 예제에서는 componentReady 때 이벤트 리스너를 등록했습니다.
- ⑥ 위젯 제거시 등록된 이벤트 리스너도 제거합니다. 해당 예제에서는 componentRemove 때 이벤트 리스너를 제거하였습니다.

해당 설명에서 나오는 함수 API 입니다.

클래스	함수명	파라미터	설명
BIX5.dashboard	<b>static</b> object <b>createEvent</b> (type, detail)	type: string 생성할 이벤트명 detail : object 이벤트로 전달할 세부정보	위젯 간 이벤트 연동시 사용되는 이벤트를 생성합니다.



interactionChart1.html

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

    widget.rangeFunc = function(type, x, y){
        //CrossRangeZoomer의 마우스 이벤트가 발생할 경우 호출됩니다.
        //getSlide() 메서드를 통해 위젯이 소속된 슬라이드를 얻을 수 있습니다.
        var slide = widget.getSlide();

        //사용자 정의 이벤트를 만들 수 있습니다.
        var evt = BIX5.dashboard.createEvent("rangeUpdate");

        var detail = {};

        detail.widgetId = widget.id;
        detail.componentId = widget.componentId;
        detail.type = type;
        detail.x = x;
        detail.y = y;

        //이벤트 전달 시 detail 객체를 통해 필요한 데이터를 담아 보낼 수 있습니다.
        evt.detail = detail;

        //사용자 정의 이벤트를 슬라이드에 송출합니다.
        slide.dispatchEvent(evt);
    }
</script>
```

interactionChart2.html

```
<script>
    var rangeUpdate = "rangeUpdate";
    widget.componentReadyHandler = function(componentId){
        widget.componentId = componentId;
        .....
        //전달 받을 이벤트명과 이벤트 리스너를 등록합니다.
        widget.getSlide().addEventListener(rangeUpdate, widget.rangeUpdateHandler);
    }

    widget.rangeUpdateHandler = function(event){
        //이벤트 리스너를 구현합니다.
        var type = event.detail.type;
        var x = event.detail.x;
        var y = event.detail.y;

        document.getElementById(widget.componentId).updateCrossRange(type, x, y);
    }

    widget.componentRemoveHandler = function(){
        //삭제 시 등록된 이벤트리스너를 제거합니다.
        var listener = widget.rangeUpdateHandler;

        widget.getSlide().removeEventListener(rangeUpdate, listener);
    }
</script>
```

## 8. 제품 유지보수

- **제품 유지보수 제공 기간**

전화/이메일을 통한 기술지원 : 기한 1 년

무상 하자 보수지원(제품 패치 포함): 기한 1 년

※ 방문지원은 별도의 비용이 부과됩니다.

- **기술 지원의 범위**

기술 지원 범위	제공하는 서비스
소프트웨어 라이선스	라이선스에 대한 문제 해결방안에 대해 알려드립니다.
제품 설치 및 구성	고객의 시스템 관리자가 컴퓨터에 소프트웨어를 설치하고 구성하는 것을 도와드립니다.
제품의 사용법 안내	소프트웨어와 함께 제공되는 매뉴얼에 묘사된 특정 기능의 사용법을 고객이 더 잘 이해하도록 돕습니다.
발견된 제품 결함의 해결	빅스소프트는 발견된 제품의 결함을 해결하는 패치를 비롯하여 임시해결방안을 제공합니다. (빅스소프트는 제품의 결함이 고객이 작업을 수행하는데 심각한 에러를 발생한다고 판단될 경우는 소프트웨어 새로운 버전 출시 이전이라도 패치를 제공합니다.)

## 9. BIX5 Dashboard 의 기타 유의사항 설명

### 9.1. BIX5 Dashboard 에서 지원하지 않는 기능

- 대시보드의 사이즈는 퍼센트를 지원하지만 이것이 곧 반응형을 의미하는 것은 아닙니다. 위젯의 위치와 크기는 고정으로 지원하고 있습니다.  
예를 들어 대시보드 홀더 너비가 1000px 이고 대시보드의 너비를 50%로 설정하면 대시보드의 너비는 500px 로 생성됩니다. 그러나 이에 따라 위젯의 위치, 크기 정보가 변경되는 것은 아니므로 설정에 따라 잘려 나올 수 있습니다.

### 9.2. 대시보드 사용시 유의사항

- 대시보드 생성 전 제품의 자원 경로를 꼭 지정해야 합니다. 설정 방법은 사용설명서 [4. BIX5 대시보드 생성하기](#)의 3 번을 참조바랍니다.
- 대시보드 객체는 비동기적으로 생성하므로 대시보드가 준비된 이후에 대시보드 객체를 사용할 수 있습니다. BIX5.dashboard.create 의 readyHandler 에 등록된 콜백함수가 대시보드가 준비가 되면 호출됩니다.

### 9.3. 위젯 아이디 설정 시 유의사항

- 변수명은 문자, 밑줄(\_), 달러 기호(\$)로 시작하고 그 뒤에 숫자(0~9)도 올 수 있습니다.
- 문자는 대문자 A~Z 와 소문자 a~z 를 사용할 수 있고 대소문자를 구별합니다.
- 자바스크립트에서 정한 예약어는 사용할 수 없습니다.
- 한글도 사용 가능하지만 영문자를 권장합니다.

### 9.4. 위젯 콘텐츠 설정 시 유의사항

- 위젯 콘텐츠 작성시 웹문서와 충돌되지 않게 작성하셔야 합니다.
- Html 타입 위젯에 CSS 를 적용하려면 선택자를 명확히 지정해야 합니다.

<감사합니다.>