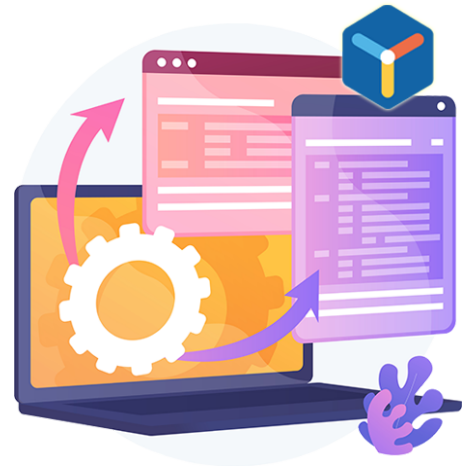




Y[wai]

Web Application Innovator



VSCode User Manual

INDEX

1. 순수 웹 개발 플랫폼, Y[wai]	
A. Y[wai] 소개	5
B. Y[wai] 구조 및 특징	6
2. Y[wai] 설치 및 환경설정	
A. VSCode Extension 설치	9
B. Wai Designer 열기	11
3. Y[wai] IDE	
A. WYSIWYG Editor	
a. About WYSIWYG Editor	15
b. Design Mode	21
c. Property Editor	31
d. GuideFrame	60
4. Y[wai] Component 및 HTML, HTML5	
A. 기본 설명 및 사용 방법	64
B. Component 소개	66
5. Y[wai] Component 도움말	
A. Grid	77
B. CodeCombo	85
C. Edit_mask	91
D. Edit_currency	95
E. Edit_date	98
F. Edit_datetime	101

G. Edit_regex	104
H. Edit_decimal	107
I. Edit_integer	110
J. DateEdit	112
K. TabControl	115
L. Calendar	117
M. Chart	119
N. TreeView	129
O. cTree	138
P. Bar Progress	146
Q. Circle Progress	148
R. External	150
S. SubFrame	152
T. Search Edit	153
U. Image button	156

6. Y[wai] Component 예제

A. Grid	158
B. CodeCombo	252
C. Edit	272
D. DateEdit	284
E. TabControl	293
F. Calendar	294
G. Chart	300
H. Treeview	343
I. cTree	375
J. Bar Progress	399
K. Circle Progress	405
L. External	413
M. SubFrame	419
N. Search Edit	420
O. Image Button	427

7. waiServer 연동 기능

A. DataService Mode	432
B. SubFrame	439
C. Template	441
D. 라이브러리 호출	443

8. 고객 지원 및 제품 공급자

A. 고객 지원 정보	445
B. 제품 공급자 정보	446

1. 순수 웹 개발 플랫폼, Y[wai]

A. Y[wai] 소개

Y[wai]는 순수 웹을 기반으로 하는 웹 개발 플랫폼입니다. 당사의 X-Internet 솔루션인 Winvader를 잇는 차세대 개발 플랫폼으로서 Winvader의 장점을 이어나가고 있습니다.

Y[wai]는 표준 웹 시스템을 지원합니다. Y[wai]로 개발되는 웹 시스템과 Y[wai] 모두 표준 웹 방식으로 개발되었습니다. Y[wai]는 Web Application Innovator의 머리글로 웹 개발 및 유지보수 과정을 쉽고 빠르게 혁신한다는 의미를 가지고 있습니다.



Y[wai]는 웹 기반 비즈니스 시스템의 화면(Front-end)단을 구축합니다. 개발 시점부터 실행 시점까지 표준 HTML과 JavaScript를 이용하는 유일한 개발 플랫폼입니다.

따라서 변환 과정 없이 HTML, CSS, JavaScript를 직접 사용하여 개발할 수 있으며, 다양한 유형의 통합 개발 환경과 HTML 태그 방식의 컴포넌트를 제공합니다.

1. 순수 웹 개발 플랫폼, Y[wai]

B. Y[wai] 구조 및 특징

Y[wai]는 항상 표준 웹 방식을 사용합니다. 웹 화면과 Script를 전용 개발 도구로 변환하는 타 솔루션들과 달리 개발 시점부터 실행 시점까지 표준 웹 방식을 사용합니다.

개발자원의 빠른 적응

개발 도구에 대한 교육이 필요 없거나 상대적으로 적습니다.
일반적인 웹 개발 방식과 비슷하여 쉽게 적응할 수 있습니다.

- ✓ 웹 디자이너가 일반적인 방식으로 디자인하고 적용 가능
- ✓ 웹 개발자가 쉽게 적응하고 응용 가능
- ✓ 개발자는 이미 개발된 코드 및 화면 구성을 쉽게 파악 가능

성능 및 안정성

HTML과 Script로 직접 개발하여 변환된 HTML과 JavaScript보다 적은 객체와 코드를 사용합니다.
따라서 메모리 사용량은 적고 로딩 속도는 빠릅니다.

- ✓ 시스템 메모리를 적게 사용
- ✓ 화면 로딩 속도가 상대적으로 빠름
- ✓ 네트워크를 상대적으로 적게 사용

유연한 개발

개발 시 공개된 웹 Library나 컴포넌트를 병행하여 사용할 수 있습니다.
개발 시간을 효율적으로 사용하면서 다양한 Needs를 수용할 수 있습니다.

- ✓ 다양한 Runtime Library를 병행하여 사용 가능
- ✓ 다양한 외부 컴포넌트를 병행하여 사용 가능

Y[wai]는 다양한 환경과 개발 방식 및 표준화를 제공하고 창의력을 발휘하게 합니다. 여러 개발 환경과 운영 환경에 적합하도록 개발 도구 및 기능을 제공합니다.

다양한 개발 환경

설치가 필요 없는 웹 기반의 통합 개발 환경 제공 ✓
Eclipse 기반의 통합 개발 환경 제공 ✓
Microsoft의 Visual Studio Code 기반의 통합 개발 환경 제공 ✓

개발과 운영, 유지보수 등 다양한 유형과
개발자의 개발 범위에 따라서
적합한 개발 도구를
선택하여 사용할 수 있습니다.

선택적 표준화

데이터 정의, 처리, 통신 표준화 ✓
다국어 및 사전 표준화 ✓
공통 모듈, 공통 화면, 공용 화면 각각의 표준화 ✓
프로젝트 사용 가능 3rd party Library 표준화 ✓
Project Theme(테마) 표준화 ✓

개발 환경과 요건에 따라서
개발의 각 부분을 선택하여
표준화할 수 있습니다.
개발 도구에 의한 강제적인
표준화가 아닌 상황에 맞춰 0~100%
개발 표준화를 선택할 수 있습니다.

Extended Web

waiBrowser를 이용한 확장된 기능 사용 ✓
(사용자 PC 설치 필요)
waiClient를 이용한 확장된 기능 사용 ✓
(사용자 PC 설치 필요)

waiBrowser와 waiClient를 사용하면
일반적인 웹 브라우저로
처리가 어려운 기능을 수행할 수 있습니다.

Y[wai]는 Front-End 단의 개발을 위한 모든 부분을 제공합니다.

IDE : 통합 개발 환경

HTML을 WYSIWYG 방식으로 편집하고
Script를 개발할 수 있는
통합 개발 환경을 제공합니다.
IDE 유형에 따라서
최적의 코드 Editor를 사용합니다.

- ✓ Web IDE
설치가 필요 없는 웹 기반의 통합 개발 환경 제공
- ✓ waiEclipse IDE
Eclipse 기반의 통합 개발 환경 제공
- ✓ VSCode IDE
Microsoft 의 Visual Studio Code 기반의 통합 개발 환경 제공

waiServer : 개발 및 운영 서버

waiServer를 사용하면
팀 단위의 개발과 표준적인 개발을
진행할 수 있습니다.
waiBrowser의 자동 업데이트 기능을
제공하며 사용자별 로그 조회 및
원격 지원 등 운영 기능을
사용할 수 있습니다.

- ✓ 개발자 및 운영자 관리 및 권한 관리
- ✓ 소스 버전 관리 및 배포 기능
- ✓ 모든 표준화 개발 및 개발 리소스 공유 관리
- ✓ waiBrowser 버전 관리 및 자동 업데이트
- ✓ 사용자별 로그 조회 및 원격 지원

waiBrowser

waiBrowser를 사용하면
일반적인 웹 브라우저에서 처리가 어려운
기능을 수행할 수 있습니다.
개발자가 표준 C# DLL 형태로 쉽게
기능을 추가하고 사용할 수 있도록
자동 Interface 기능이 내장됩니다.
waiBrowser의 모든 기능은
웹 화면의 JavaScript로 Interface됩니다.

- ✓ DLL Interface
- ✓ File Control
- ✓ Shell Processing
- ✓ TCP/IP - WebSocket Server
- ✓ Zip / Unzip
- ✓ Clipboard Control
- ✓ System Status Monitoring
- ✓ Remote Control Client
- ✓ Screen Capture
- ✓ QR / Barcode Processing

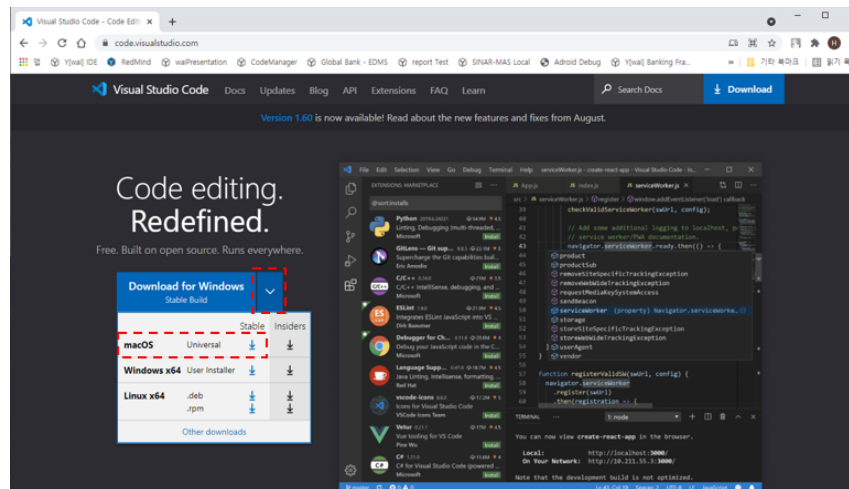
2. Y[wai] 설치 및 환경설정

A. VSCode Extension 설치

■ Microsoft Visual Studio Code 설치

Y[wai] is a purely web-based web building tool. Y[wai]는 순수 웹 기반의 웹 구축 도구로 순수 웹 기반의 컴포넌트와 WYSIWYG HTML Designer(IDE)를 제공합니다. waiVSCode는 Visual Studio Code 환경에서 Extension으로 제공되기 때문에 waiVSCode를 사용하려면 Visual Studio Code가 설치되어 있어야 합니다.

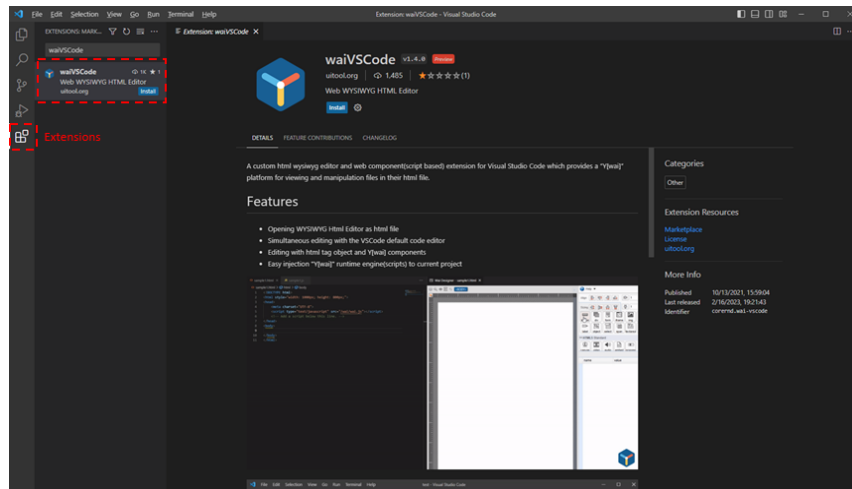
Visual Studio Code(이하 : VSCode)는 <https://code.visualstudio.com/>에서 다운로드하여 설치할 수 있습니다.



■ waiVSCode Extension 설치

VSCode를 실행한 후 Activity Bar에서 Extensions tab을 선택합니다.

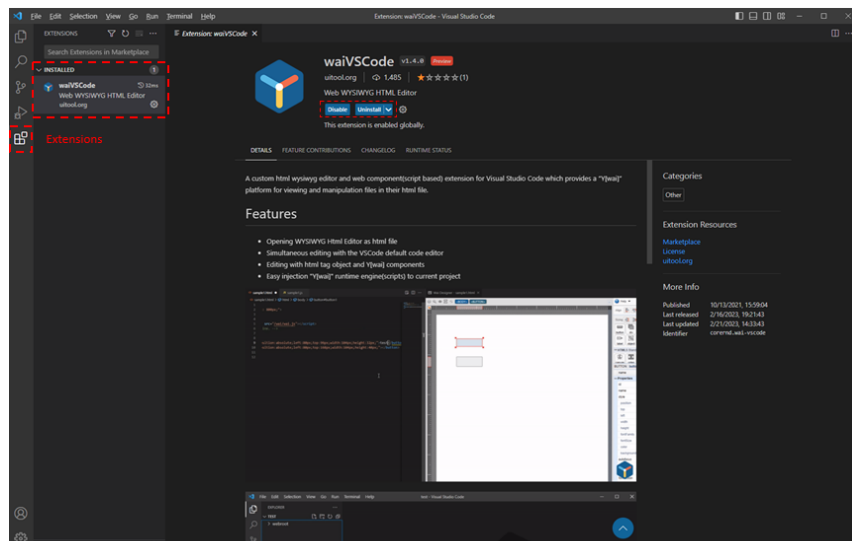
Side Bar의 검색창에 'waiVSCode' 또는 'wysiwyg'이라고 검색하면 waiVSCode Extension이 검색됩니다. 검색된 waiVSCode extension을 설치한 다음 VSCode를 재시작합니다.



■ waiVSCode Extension 설치 확인

VSCode를 실행한 후 Extensions Tab을 보면 waiVSCode가 설치된 것을 확인할 수 있습니다.

'Disable' 버튼으로 waiVSCode를 비활성화하거나 'Uninstall' 버튼으로 삭제할 수 있습니다.



2. Y[wai] 설치 및 환경설정

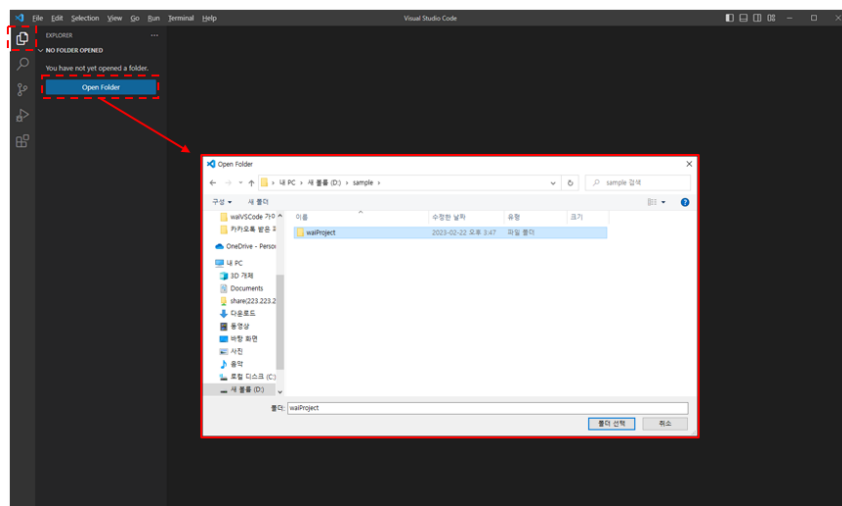
B. Wai Designer 열기

■ 폴더 선택

먼저 프로젝트에서 사용할 폴더를 정합니다.

좌측 사이드 바에서 Explorer Tab을 선택한 후 'Open Folder' 버튼을 클릭합니다.

파일 선택창이 열리면 프로젝트에서 사용할 폴더를 선택합니다.

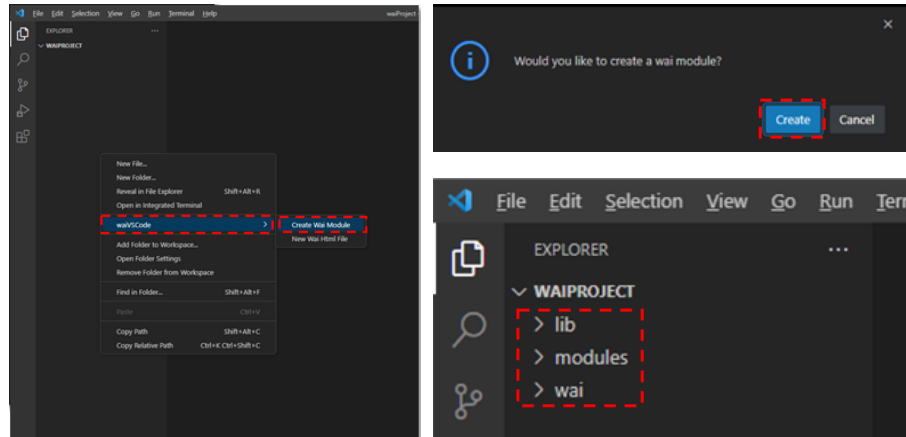


■ Wai Module 생성

Explorer Tab은 선택한 프로젝트 폴더를 기준으로 구성됩니다.

폴더 내부 영역을 선택하면 파란색 테두리가 생기는 것을 확인할 수 있습니다. 해당 영역에서 우클릭을 하고 'waiVSCode' 메뉴 하위의 'Create Wai Module' 메뉴를 선택합니다. 확인 창이 나타나면 'Create' 버튼을 클릭하여 Wai Module을 생성합니다.

생성된 폴더 및 파일(Wai Module)은 Y[wai] 컴포넌트 및 Y[wai] Runtime Engine입니다.

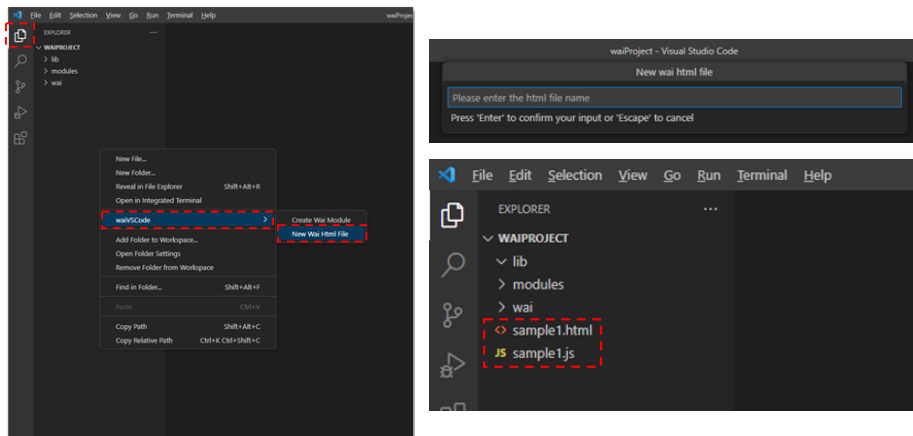


■ New HTML File 생성

Explorer Tab에서 HTML 파일을 생성하고 싶은 경로를 우클릭합니다.

'waiVSCode' 메뉴 하위의 'New Wai Html File' 버튼을 클릭하고 VSCode 상단 영역에 'New Wai Html File' 입력 창이 나타나면 만들고자 하는 HTML 파일명과 엔터 키를 누릅니다.

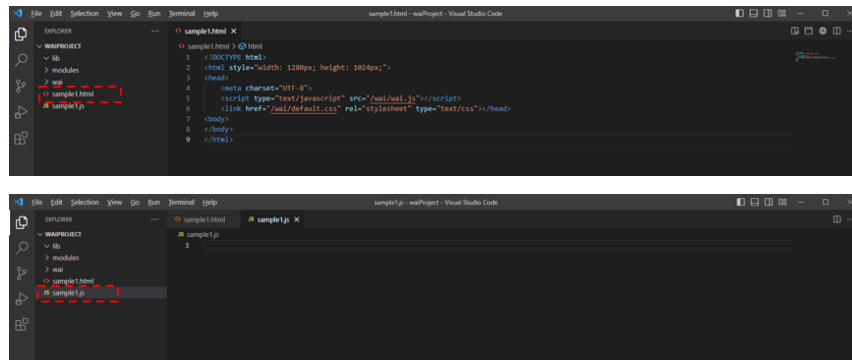
입력한 파일명의 HTML과 Javascript 파일이 각각 생성됩니다.



■ New HTML File 확인

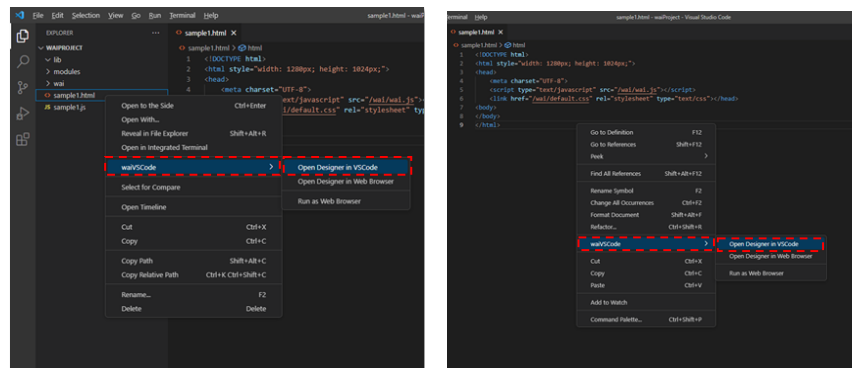
생성한 HTML 파일을 열면 기본 템플릿이 작성된 소스를 확인할 수 있습니다. 앞서 프로젝트 폴더에 생성한 Wai Module을 포함하고 있으며, HTML 파일의 이름과 같은 Javascript 파일이 자동 생성됩니다.

HTML 작동 시 HTML 안에서 Script 연결을 하지 않아도 Wai Module에 의해 자동으로 해당 HTML과 같은 이름의 Javascript 파일을 연결(로드)합니다. Y[wai]는 HTML과 Script를 분리하여 개발하는 것을 원칙으로 합니다.



■ VSCode에서 Wai Designer 열기

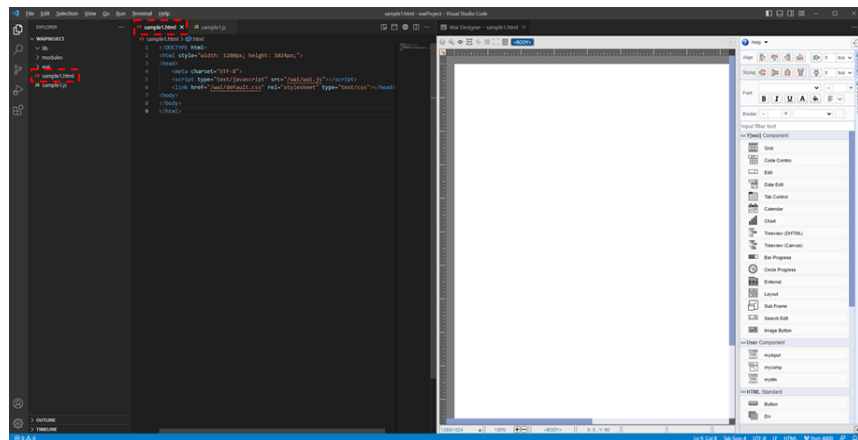
Explorer Tab에서 HTML 파일을 우클릭하고 'waiVSCode' 메뉴 하위의 'Open Designer in VSCode' 메뉴를 클릭합니다. 또는 HTML Editor 영역에서 우클릭하고 'waiVSCode' 메뉴 하위의 'Open Designer in VSCode' 메뉴를 클릭합니다.



■ Wai Designer 확인

VSCode 내부에 Wai Designer가 열립니다.

HTML Editor에서 소스 코드를 직접 작성할 수 있으며, Wai Designer에서 Drag&Drop으로 HTML을 작성할 수 있습니다.



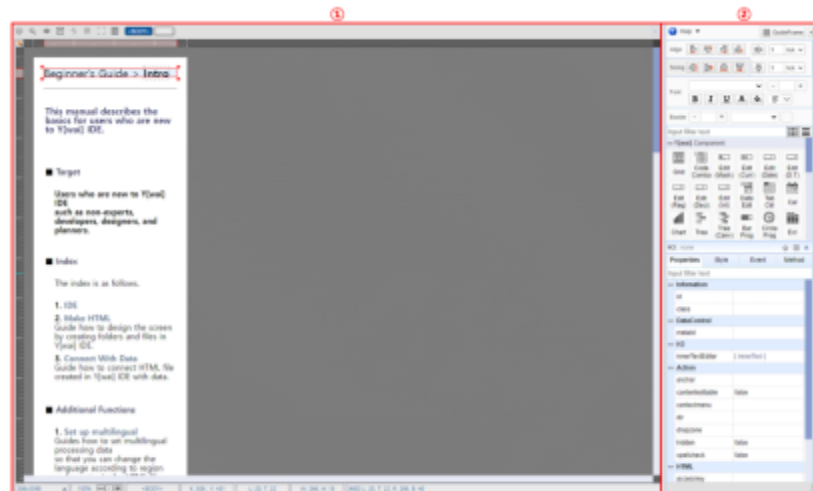
3. Y[wai] IDE

A. WYSIWYG Editor

a. About WYSIWYG Editor

HTML 태그 및 Y[wai] Component를 사용하여 다양한 객체의 디자인과 속성을 쉽게 생성하고 웹 화면을 편집할 수 있습니다. 또한 HTML Page를 효율적으로 생성하고 편집할 수 있는 여러 기능을 제공합니다.

■ 메인 화면



메인 화면

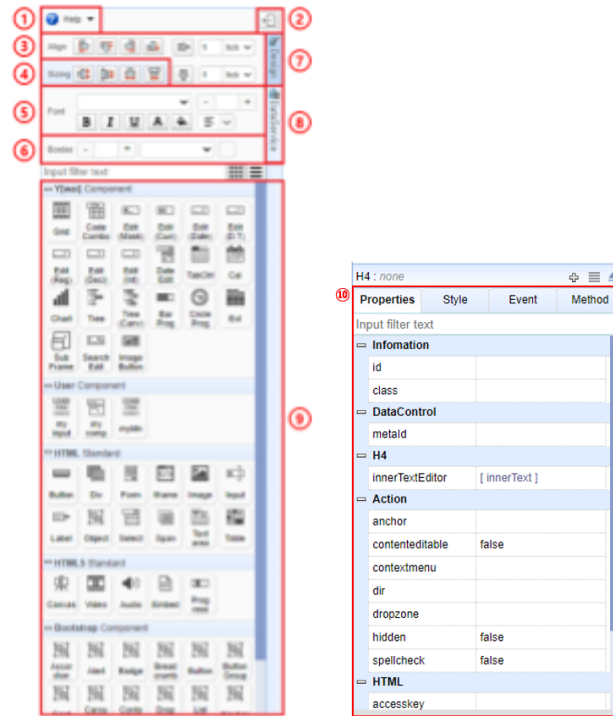
① WYSIWYG 화면

객체의 디자인을 편집할 수 있습니다.

② 도구 모음 창

WYSIWYG 화면에서 사용할 수 있는 도구들의 모음입니다.

■ 도구 모음 화면



도구 모음 화면

- ① Help
도움말 페이지를 호출합니다.
- ② Change Position (Left/Right)
도구 모음 창을 왼쪽이나 오른쪽에 배치합니다.
- ③ Align (Design Tools)
선택된 컴포넌트의 위치를 일괄적으로 정렬하고 간격을 조절합니다.
- ④ Sizing (Design Tools)
선택된 컴포넌트의 사이즈를 일괄적으로 변경합니다.
- ⑤ Font
선택된 컴포넌트의 글꼴을 설정합니다.

⑥ Border

선택된 컴포넌트의 테두리를 설정합니다.

⑦ Design Mode

HTML 코드를 변환하여 WYSIWYG Editor 모드로 보여줍니다.

⑧ DataService Mode

데이터를 처리하기 위한 기능을 제공합니다.

(waiServer와 연동된 경우에만 사용 가능)

⑨ Component Palette

Y[wai] Component, HTML, HTML5의 태그를 사용할 수 있습니다.

⑩ Property Editor

선택된 객체의 속성을 편집할 수 있습니다.

■ Design Tools

여러 개의 객체를 선택하고 정렬, 높이, 넓이, 간격을 한 번에 조정할 수 있습니다.

여러 개의 객체를 한 번에 설정할 경우 해당 버튼에 마우스 커서를 대면 빨간 점선으로 실행 결과를 미리 볼 수 있으며 오류를 최소화할 수 있습니다.



① Align Control

다중 선택된 객체를 정렬합니다.

② Sizing Control

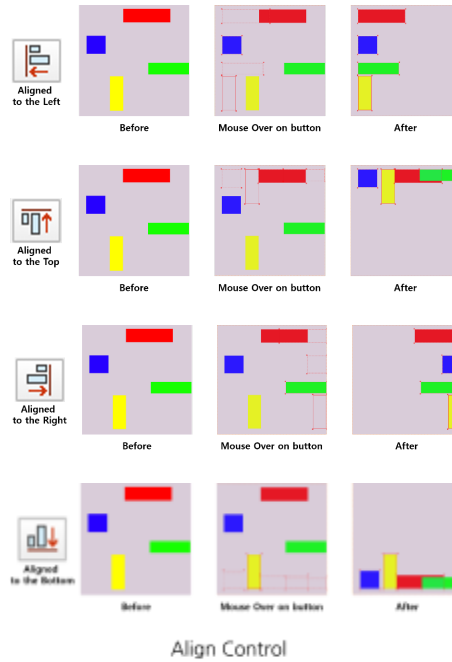
다중 선택된 객체의 넓이와 높이를 설정합니다.

③ Separation Space Control

객체 간의 좌우 및 상하 간격을 설정합니다.

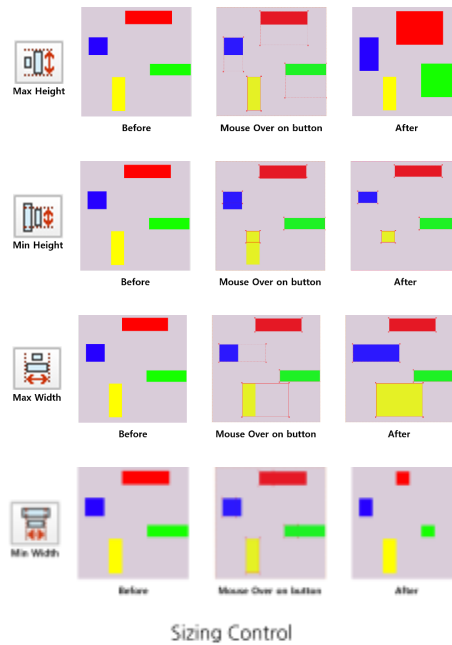
■ Align Control

가장 끝에 있는 객체의 위치에 맞춰서 정렬합니다.



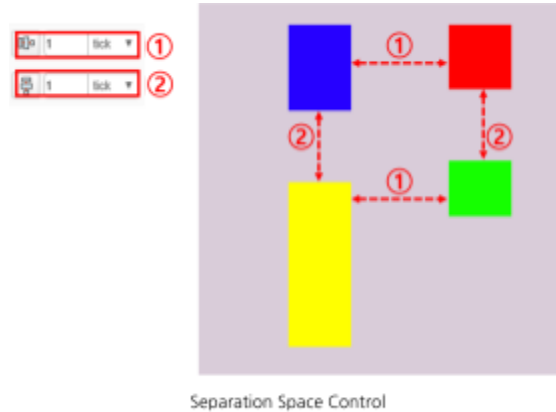
■ Sizing Control

최대 높이 및 최소 높이, 최대 넓이 및 최소 넓이에 맞춰서 객체의 높이와 넓이를 설정합니다.



■ Separation Space Control

다중 선택된 객체의 가로와 세로 간격을 px 혹은 설정한 snap(tick) 단위에 맞춰서 정렬합니다.



① Horizontal Separation Space

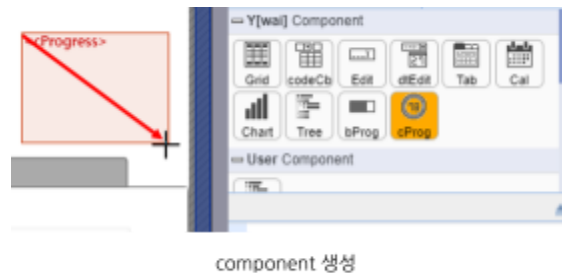
선택된 객체 사이의 가로 간격을 설정합니다.

② Vertical Separation Space

선택된 객체 사이의 세로 간격을 설정합니다.

■ Component Palette

다양한 컴포넌트를 Component Palette에서 선택하여 Drag로 생성할 수 있습니다. ESC를 누르거나 선택한 Component를 다시 클릭하면 선택이 해제됩니다.

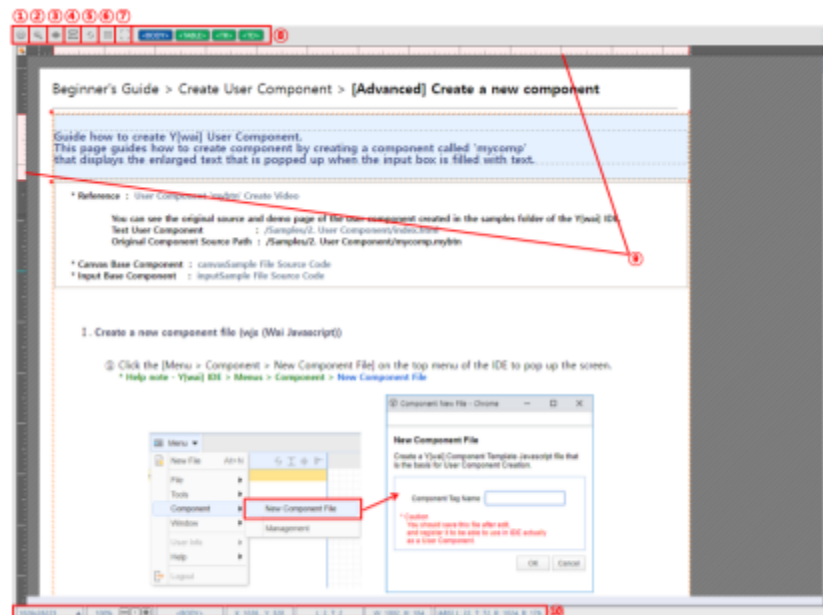


3. Y[wai] IDE

A. WYSIWYG Editor

b. Design Mode

■ 메인 화면



메인 화면

① Editing Options

각 모드별로 사용할 수 있는 옵션을 설정합니다.

② Search Object

컴포넌트 객체를 검색합니다.

③ View Size

화면의 사이즈를 설정합니다.

④ Hidden Object Viewer

숨겨진 컴포넌트 객체를 표시합니다.

⑤ Reload Page

작업 중인 페이지를 새로고침합니다.

⑥ Show Guideline

디자이너 배경에 가이드라인을 표시합니다.

⑦ Show Outline

모든 컴포넌트의 아웃라인을 표시합니다.

⑧ Depth Selector

선택한 객체의 부모 객체를 표시하고 선택합니다.

⑨ Ruler

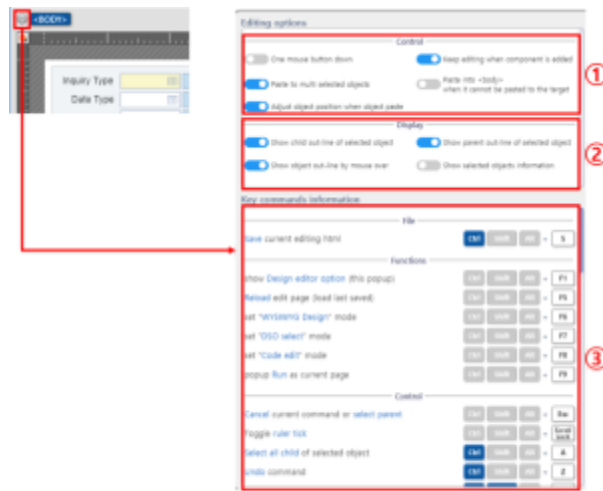
객체를 이동하거나 위치를 설정합니다.

⑩ Position / Size

선택된 객체의 위치 및 크기 등을 표시합니다.

HTML 페이지 사이즈 관련 옵션을 수정합니다.

■ Editing Options



Editing Options

① Control

마우스 버튼 이벤트, 복사, 붙여넣기 옵션 등을 설정합니다.

- ✓ One mouse button down : Click & Drag 한 번으로 이동
- ✓ Keep editing when component is added : 연속 컴포넌트 추가
- ✓ Paste to multi selected objects : 멀티 선택 붙여넣기
- ✓ Paste into <body> when it cannot be pasted to the target :
붙여 넣을 대상이 없을 경우 body에 붙여넣기

② Display

화면에 표시되는 효과를 설정합니다.

- ✓ Show child out-line of selected object : 선택된 객체의 자식 객체 표시
- ✓ Show parent out-line of selected object : 선택된 객체의 부모 객체 표시
- ✓ Show object out-line by mouse over : 클릭할 시 선택될 객체 미리 표시
- ✓ Show selected objects information : 선택된 객체의 정보 표시

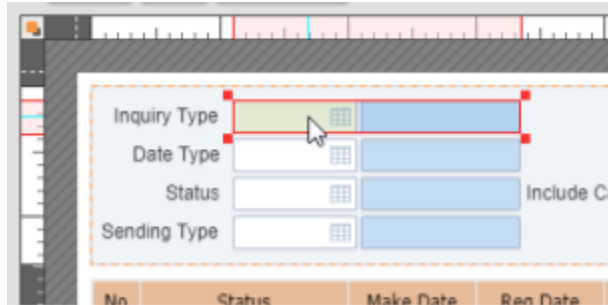
③ Key commands information

단축키 설정 모음

■ Ruler

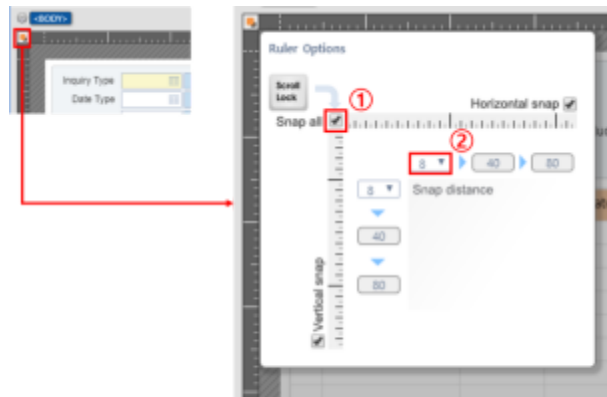
객체의 이동하거나 위치를 설정합니다.

선택한 객체는 빨간색 선으로 표시하며, 부모 객체가 있을 경우 하얀색 바탕으로 표시합니다.



객체를 선택할 경우 Ruler

좌측 상단 버튼을 클릭하면 Ruler 옵션을 설정할 수 있습니다.



Ruler Options

① Snap On / Off

간격 설정을 활성화하거나 비활성화합니다.

ScrollLock 키를 이용해 쉽게 사용할 수 있으며, 가로와 세로를 별도로 설정할 수 있습니다.

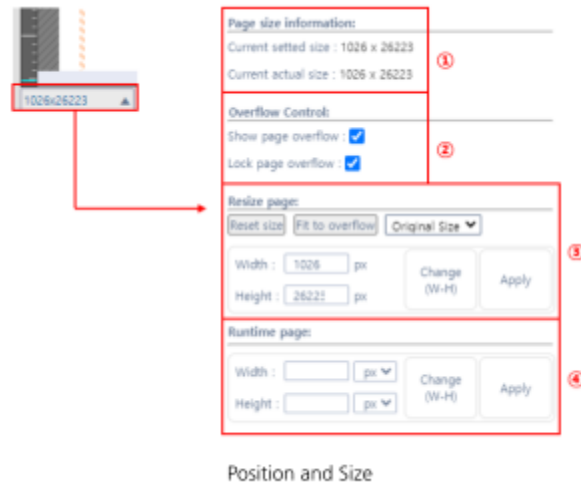
② Snap Distance

가로와 세로의 간격을 조절합니다.

■ Position / Size

현재 디자인하고 있는 HTML의 크기와 디자인 화면에서 마우스의 위치를 표시합니다.

객체를 선택하면 선택된 객체의 크기를 표시합니다. HTML 페이지 사이즈를 클릭하면 화면 조정 옵션을 설정할 수 있습니다.



① Page Size Information

현재 설정된 페이지 크기와 실제 페이지 크기를 비교하여 overflow 여부를 알려줍니다.

② Overflow Control

설정된 크기를 초과하는 영역에 관해 설정합니다.

- ✓ Show page overflow : overflow된 영역을 보이거나 가리기
- ✓ Lock page overflow : overflow된 영역으로 객체를 이동하거나 이동하지 않음

③ Resize Page

페이지 크기를 조절합니다. 특정 디바이스의 화면 크기로 재조정할 수 있습니다.

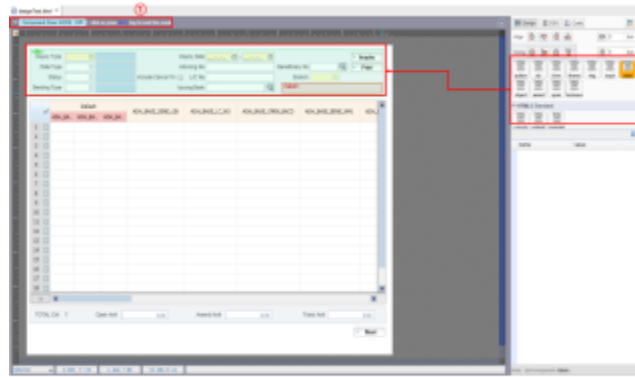
④ Runtime Page

화면 실행(Runtime)시 화면 사이즈를 설정합니다.

■ Component Draw Mode

도구 모음에서 component를 선택할 시 활성화됩니다.

component를 연속해서 추가할 수 있으며, 부모 객체가 있는 경우 초록색으로 표시됩니다.



Component Draw Mode

① Component Draw Mode 종료

ESC를 누르거나 해당 컴포넌트를 클릭합니다.

■ Mouse Control

기본적인 마우스 컨트롤과 키보드 단축키로 다양한 기능을 사용할 수 있습니다.

① 객체 생성

Component Palette에서 component를 선택한 후에 생성합니다.

- ✓ Left Click + Drag : 객체 생성
- ✓ Alt + Left Click + Drag : 생성된 객체를 무시하고 <body> 태그에 객체 생성

② 객체 선택 / 해제

- ✓ Left Click : 마우스 왼쪽 클릭으로 객체를 선택하거나 해제



- ✓ Shift + Left Click : 여러 개의 객체를 추가로 선택하거나 해제



- ✓ Drag : Drag를 하여 객체의 영역을 선택하거나 해제



③ 이동 (선택 상태)

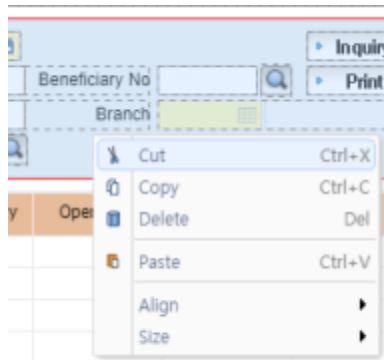
- ✓ Drag : 선택한 객체를 이동

④ 크기 조정 (선택 상태)

- ✓ Drag : snap 단위로 크기 조정
- ✓ Ctrl + Drag : snap 단위를 무시하고 크기 조정

⑤ 메뉴 창

- ✓ Right Click : 메뉴 창을 보여줌
- ✓ 선택 + Right Click : 선택된 객체의 메뉴 창을 보여줌



선택 객체 메뉴 창

■ Key Control

① Base

- ✓ Ctrl + S : 편집 화면 저장
- ✓ F1 : Editing Options 열기
- ✓ F5 : 편집 화면 새로고침 (최근 저장 내용 로드)
- ✓ F6 : Design Mode 실행
- ✓ F9 : 편집 화면 미리 보기

② Basic Control

- ✓ ESC : 명령을 취소하거나 부모 객체를 선택
- ✓ ScrollLock : Ruler를 활성화거나 비활성화
- ✓ Ctrl + A : 선택된 객체의 모든 자식 객체 선택 (미선택할 시 <body> 안의 모든 객체 선택)
- ✓ Ctrl + Z : 되돌리기
- ✓ Ctrl + Y : 되돌리기 취소

③ Object Default Control

- ✓ Delete : 객체 삭제
- ✓ Ctrl + C : 객체 복사
- ✓ Ctrl + X : 객체 잘라내기
- ✓ Ctrl + V : 객체 붙여 넣기

④ Object Move Control

- ✓ ← : Ruler 단위(snap) 만큼 왼쪽으로 이동
- ✓ ↑ : Ruler 단위(snap) 만큼 위쪽으로 이동
- ✓ → : Ruler 단위(snap) 만큼 오른쪽으로 이동
- ✓ ↓ : Ruler 단위(snap) 만큼 아래쪽으로 이동
- ✓ Ctrl + ← : 1px씩 왼쪽으로 이동
- ✓ Ctrl + ↑ : 1px씩 위쪽으로 이동
- ✓ Ctrl + → : 1px씩 오른쪽으로 이동
- ✓ Ctrl + ↓ : 1px씩 아래쪽으로 이동

⑤ Object Size Control

- ✓ Shift + ← : Ruler 단위(snap) 만큼 객체 넓이 줄이기
- ✓ Shift + ↑ : Ruler 단위(snap) 만큼 객체 높이 줄이기
- ✓ Shift + → : Ruler 단위(snap) 만큼 객체 넓이 늘리기
- ✓ Shift + ↓ : Ruler 단위(snap) 만큼 객체 높이 늘리기
- ✓ Ctrl + Shift + ← : 1px씩 객체 넓이 줄이기
- ✓ Ctrl + Shift + ↑ : 1px씩 객체 높이 줄이기
- ✓ Ctrl + Shift + → : 1px씩 객체 넓이 늘리기
- ✓ Ctrl + Shift + ↓ : 1px씩 객체 높이 늘리기

⑥ Multi-Selected Objects Align Control

- ✓ Ctrl + Alt + ← : 다중 선택된 객체를 왼쪽으로 맞춤 정렬
- ✓ Ctrl + Alt + ↑ : 다중 선택된 객체를 위쪽으로 맞춤 정렬
- ✓ Ctrl + Alt + → : 다중 선택된 객체를 오른쪽으로 맞춤 정렬
- ✓ Ctrl + Alt + ↓ : 다중 선택된 객체를 아래쪽으로 맞춤 정렬

⑦ Multi-Selected Objects Size Control

- ✓ Ctrl + Shift + Alt + ← : 다중 선택된 객체의 넓이를 최소 넓이로 설정
- ✓ Ctrl + Shift + Alt + ↑ : 다중 선택된 객체의 높이를 최소 높이로 설정
- ✓ Ctrl + Shift + Alt + → : 다중 선택된 객체의 넓이를 최대 넓이로 설정
- ✓ Ctrl + Shift + Alt + ↓ : 다중 선택된 객체의 높이를 최대 높이로 설정

⑧ Component Draw Mode에서 객체 선택

한글 입력 모드에서는 작동하지 않습니다.

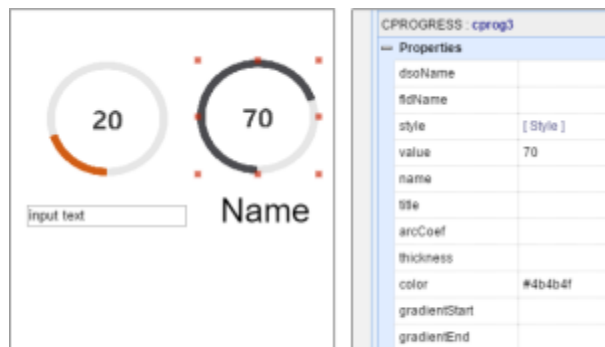
- | | |
|------------------------------|-------------------------------|
| ✓ B : button component 선택 | ✓ L : label component 선택 |
| ✓ C : codecombo component 선택 | ✓ M : img component 선택 |
| ✓ D : dateedit component 선택 | ✓ S : select component 선택 |
| ✓ E : edit component 선택 | ✓ T : tabcontrol component 선택 |
| ✓ G : grid component 선택 | ✓ V : div component 선택 |
| ✓ I : input component 선택 | ✓ X : textarea component 선택 |

3. Y[wai] IDE

A. WYSIWYG Editor

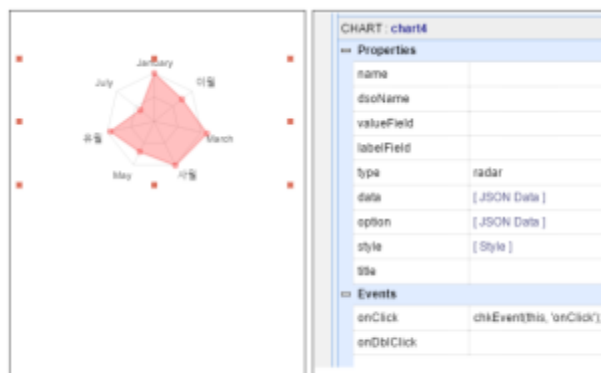
c. Property Editor

WYSIWYG Editor에서는 객체의 속성을 효율적으로 수정하고 제어할 수 있도록 Property Editor를 제공합니다. 사용자는 객체의 속성이나 이벤트를 알지 못해도 Property Editor를 이용하여 객체를 빠르고 쉽게 customizing 할 수 있습니다. 또한 다양한 속성 창을 지원하며 편리한 사용자 인터페이스를 제공합니다.



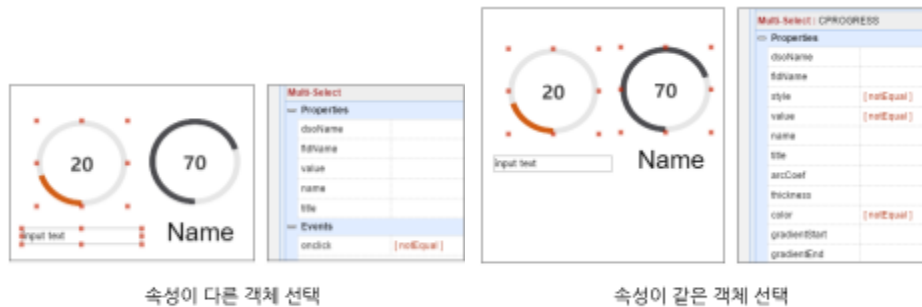
WYSIWYG Editor가 제공하는 Property Editor

Property Editor는 WYSIWYG Editor에서 Design Mode일 때 객체를 선택하면 나타납니다. 선택된 객체의 속성과 이벤트 목록 및 코드에 입력된 속성값들이 Property Editor에 표시됩니다.



Property Editor 구성

WYSIWYG Editor에서 객체를 선택하면 화면 왼쪽에 있는 Property Editor에 객체의 속성과 이벤트가 표시됩니다. 상단에 [컴포넌트 명 : 객체 ID]가 표시되며, 선택된 객체의 Properties와 Events 목록을 표시합니다.



속성이 다른 객체가 선택되면 상단에 [Multi-Select]로 표시되며 공통 속성만 표시됩니다. 속성이 같은 객체가 선택되면 상단에 [Multi-Select : 컴포넌트]로 표시됩니다. 선택된 객체의 속성값이 같을 경우 값을 그대로 표시하고 다를 경우 [notEqual]로 표시됩니다.

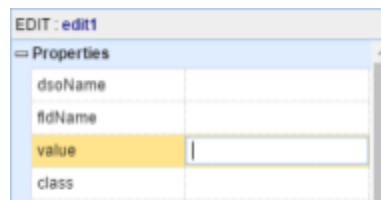
여러 객체의 속성값을 한 번에 편집할 수 있습니다.

사용자는 객체의 Property를 편집할 수 있는 Property 항목에서 객체의 속성을 쉽고 빠르게 편집할 수 있습니다. 객체의 Property는 객체의 특성에 따라 속성값을 제어합니다.

■ Text

일반적인 텍스트 값을 입력합니다.

객체의 속성에 따라 String 혹은 Number 형태로 입력할 수 있습니다.

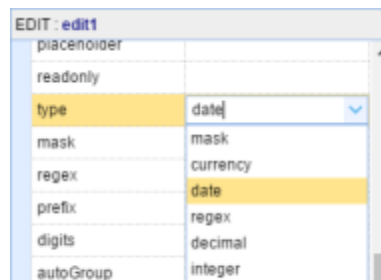


Text Field

■ Select

정해진 속성값이 Select Box로 표시됩니다.

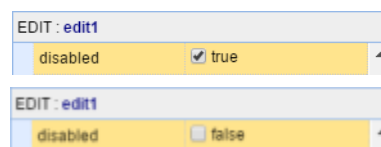
속성값을 고를 수 있습니다.



Select Field

■ Boolean

Boolean 속성값을 true 혹은 false로 체크하여 나타냅니다.



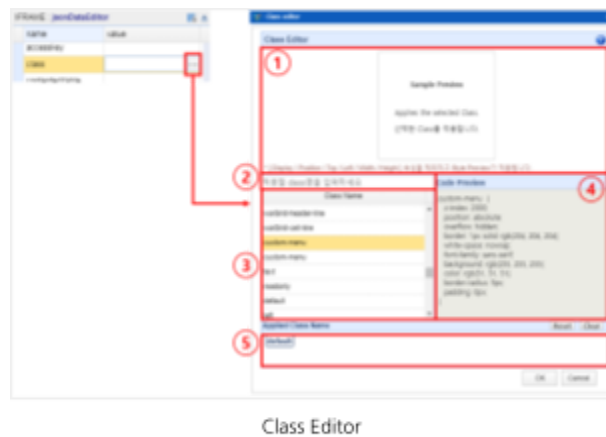
Boolean Field

또한 사용자가 속성을 효율적으로 편집할 수 있도록 Sub Editor를 지원합니다.

■ Class Editor

편집 중인 페이지에 로드된 CSS 파일의 Class 혹은 페이지에 직접 코딩한 Class 소스를 연결합니다.

연결할 수 있는 Class Name 목록이 표시되면 원하는 Class Name을 선택하고 오른쪽에서 소스를 미리 볼 수 있습니다. 선택된 Class 속성이 적용된 모습은 상단에서 미리 볼 수 있으며, 하단에는 추가할 Class Name이 표시됩니다. Display, Position, Top, Left, Width, Height 속성이 제외된 Style이 적용됩니다.



Class Editor

① Sample Preview

선택된 Class가 적용된 모습을 미리 볼 수 있습니다.

② 검색

원하는 Class를 검색하여 찾을 수 있습니다.

③ 목록

현재 페이지에 연결할 수 있는 Class Name이나 검색된 Class Name을 목록으로 보여줍니다.

④ Code Preview

선택된 Class의 코드를 보여줍니다.

⑤ Class List

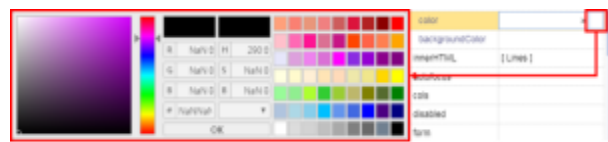
목록에서 적용할 Class를 더블클릭하면 Class List에서 버튼 형식으로 추가됩니다.

버튼을 선택하면 해당 Class를 List에서 삭제할 수 있습니다.

■ Color Editor

원하는 색상을 쉽게 선택할 수 있도록 도와줍니다.

객체의 Property Editor에서 color 부분의 버튼을 클릭하면 Color Editor가 팝업됩니다.



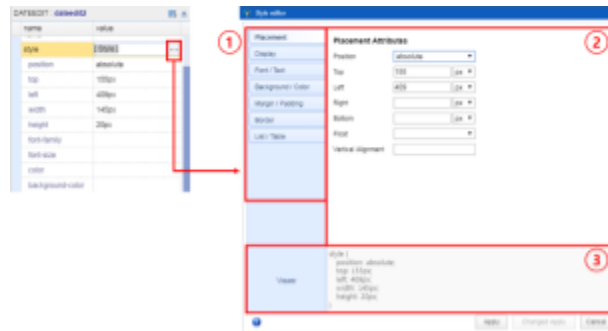
Color Editor

RGB 또는 HSB 값을 입력하여 원하는 색상을 설정할 수 있습니다. 기본으로 제공되는 색상을 이용하면 색상을 더욱 쉽게 선택할 수 있습니다.

■ Style Editor

선택한 객체의 Style 속성을 편집하고 적용합니다.

Style은 Properties에서 일괄적으로 [Style]로 표시되며, Property Editor의 Style 속성에 연결됩니다.



Style Editor

① Style Attribute Tabs

Style을 속성별로 묶어 Tab으로 제공합니다.

- ✓ Placement : Position 및 객체의 위치 스타일 설정
- ✓ Display : 객체의 Display 관련 속성 설정
- ✓ Font/Text : 폰트 및 글꼴 설정
- ✓ Background/Color : Background 및 Color 속성 설정
- ✓ Margin/Padding : 여백 속성 설정
- ✓ Border : 테두리 속성 설정
- ✓ List/Table : List 및 Table 속성 설정
- ✓ Others : 기타 속성 설정

② Attributes Detail

Tab에 해당되는 Style 속성을 수정할 수 있습니다.

③ CSS Code Viewer

수정하거나 적용한 속성을 code로 미리 볼 수 있습니다.

■ Chart Option Editor

Chart Component의 option을 쉽게 설정할 수 있도록 도와줍니다.

Bar, Line, Radar, Pie, PolarArea, Doughnut Chart에 해당하는 Chart Option Editor를 이용하여 option을 편리하게 선택할 수 있습니다. Chart Component의 Property Editor에서 option 부분의 버튼을 클릭하면 Chart Option Editor가 팝업됩니다.

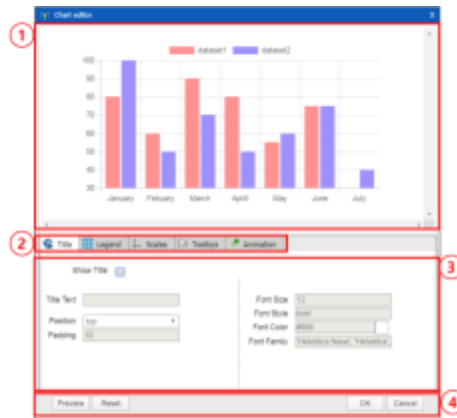


Chart Option Editor

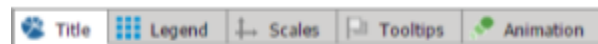
① Preview

Chart의 option을 변경한 모습을 실제 크기로 미리 볼 수 있습니다.

② Tab

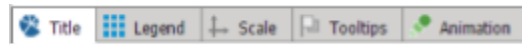
Chart의 기능별 Option Tab입니다.

Chart의 유형에 따라 각각 다른 Tab과 Option이 표시됩니다.



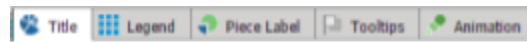
Bar / Line

- ✓ Title : Chart의 타이틀 설정
- ✓ Legend : Chart의 범례 설정
- ✓ Scale : Chart의 x, y축 option 설정
- ✓ Tooltips : Chart에 마우스 오버 시 데이터 정보가 표시되는 툴팁 설정 (Chart 공통)
- ✓ Animation : Chart에 애니메이션 설정 (Chart 공통)



Radar / PolarArea

- ✓ Scale : Chart의 축 option 설정



Pie / Doughnut

- ✓ Piece Label : Chart의 데이터 세그먼트의 Label option 설정

③ Option 편집

Tab별로 option을 설정합니다.

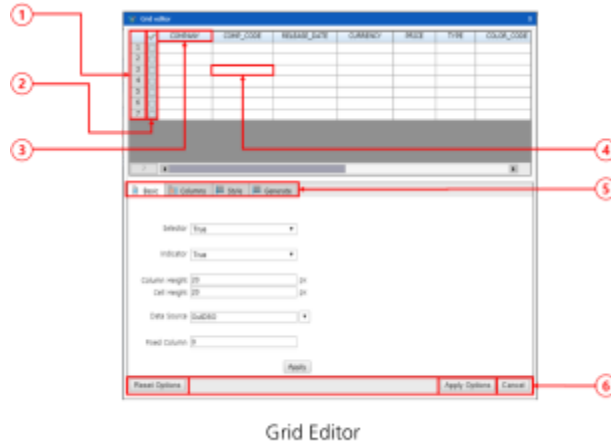
④ Button

- ✓ Preview : Chart Preview에서 변경한 option이 적용된 모습을 미리 보기
- ✓ Reset : Chart Option Editor 초기화
- ✓ Apply Options : 변경한 option 적용
- ✓ Cancel : Chart Option Editor 닫기

■ Grid Editor

Grid를 직접 디자인합니다.

Grid 객체의 Property Editor에서 options 부분의 버튼을 클릭하면 Grid Editor가 팝업됩니다.



① Indicator

데이터의 행 번호를 표시합니다.

② Selector

특정 행을 선택합니다.

선택된 특정 행은 일괄적으로 복사하거나 삭제할 수 있습니다.

③ Header

컬럼명 및 컬럼 그룹명 등을 표시합니다.

④ Cell

데이터의 항목값을 표시합니다.

⑤ Editing Item

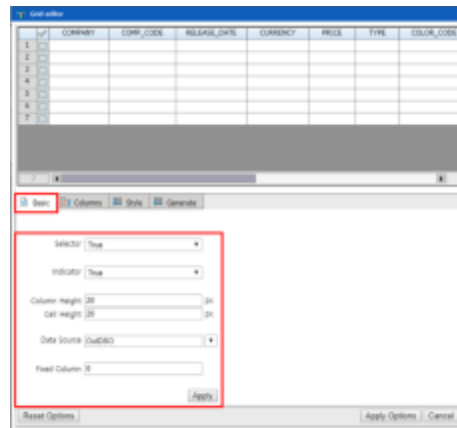
- ✓ Basic : Grid의 일반적인 속성 편집
- ✓ Columns : 선택된 컬럼의 개별 속성 편집
- ✓ Style : 컬럼에 적용할 수 있는 스타일을 생성, 편집, 삭제
- ✓ Generate : DSO와 연동하여 컬럼을 자동 생성

⑥ Option Button

- ✓ Reset Option : 편집하기 전 option으로 되돌리기
- ✓ Apply Option : 편집한 option 적용
- ✓ Cancel : 편집한 option 취소

■ Grid Editor / Basic Tab

일반적인 속성을 편집합니다.



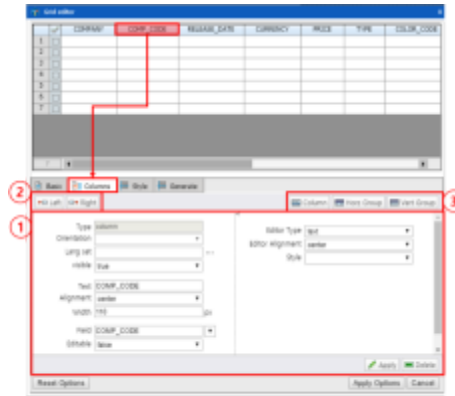
Basic Tab

- ✓ Selector : Selector 표시 여부 설정 (True/False)
- ✓ Indicator : Indicator 표시 여부 설정 (True/False)
- ✓ Column Height : Header의 높이 설정
- ✓ Cell Height : Cell의 높이 설정
- ✓ Data Source : Grid와 연동할 DSO 선택
- ✓ Fixed Column : Grid 컬럼 중 왼쪽에서부터 고정시킬 컬럼의 수 설정
- ✓ Apply : 설정한 내용을 적용

■ Grid Editor / Columns Tab

column의 개별 속성을 편집합니다.

Grid Editor에서 편집하고 싶은 column을 선택하면 Columns 편집 창으로 이동합니다.



Columns Tab

① Editing Item

선택된 column을 세부 편집합니다.

- ✓ Type : column이 group인지 일반 column인지 표시
- ✓ Orientation : column이 group일 경우 horizontal과 vertical 중 선택
- ✓ Text : column에 표시되는 text 설정
- ✓ Alignment : column에 표시되는 text를 정렬 (left/center/right)
- ✓ Width : column의 넓이 설정
- ✓ Field : column과 연동되는 DSO 객체의 Field명 선택
- ✓ Editor Type : column의 Type 설정 (Text/lookup/calcEdit/Currency/Date)
- ✓ Editor Alignment : column에 해당하는 Cell을 정렬 (left/center/right)
- ✓ Style : column에 적용할 Style 설정 (Style 탭에서 해당 Style 확인 및 수정 가능)
- ✓ Apply : 설정한 내용을 적용
- ✓ Delete : 선택된 컬럼 삭제

② Move Column

선택된 column을 이동합니다.

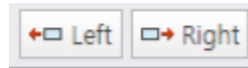
③ Add / Group Column

column을 생성하거나 선택된 column을 group화합니다.

[Column 위치 변경]

① Button

버튼을 클릭하여 한 칸씩 이동합니다.



Button

② Drag & Drop

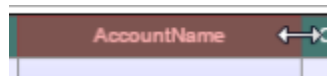
마우스 왼쪽 버튼을 누르고 원하는 위치로 Drag&Drop합니다.



Drag & Drop

[Column 크기 변경]

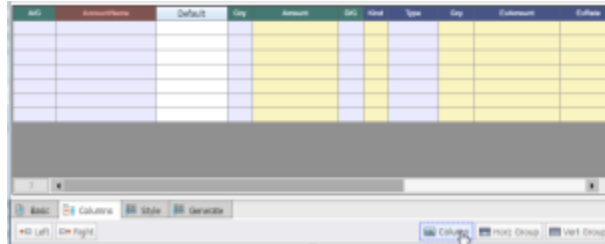
Width에서 변경하거나 마우스 왼쪽 버튼을 누르고 커서를 움직여 변경할 수 있습니다.



마우스를 움직여서 Size 변경

[Column 생성]

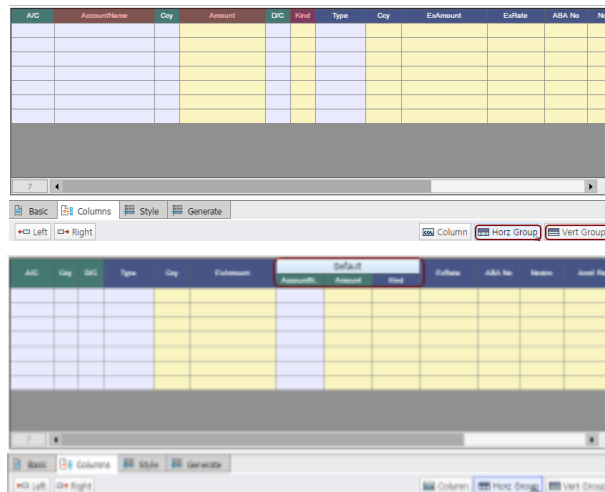
column을 생성하고 싶은 위치에 왼쪽 column을 선택하고 Column 버튼을 클릭하면 기본 column이 생성됩니다.



Column 생성

[Group 생성]

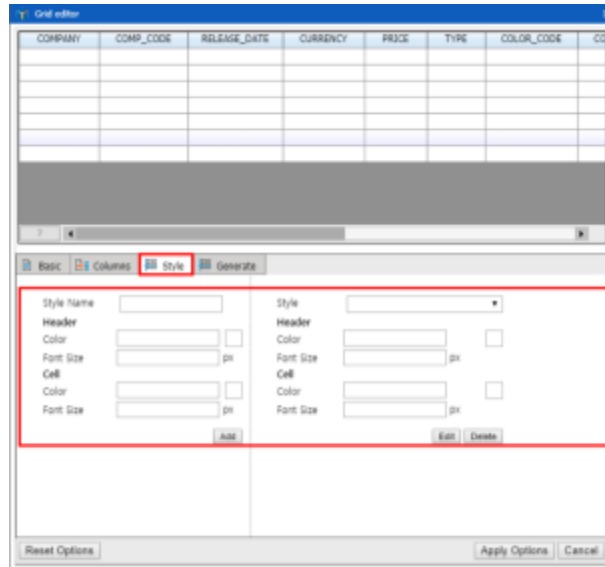
group으로 묶고 싶은 column을 Ctrl 혹은 Shift를 누르고 하나씩 선택합니다.
group 형태에 따라 Horz Group 혹은 Vert Group을 누르면 기본 group이 생성됩니다.
새로 생성된 group을 선택하고 이름 및 Style을 변경합니다.



group 생성

■ Grid Editor / Style Tab

column의 Style을 편집합니다.

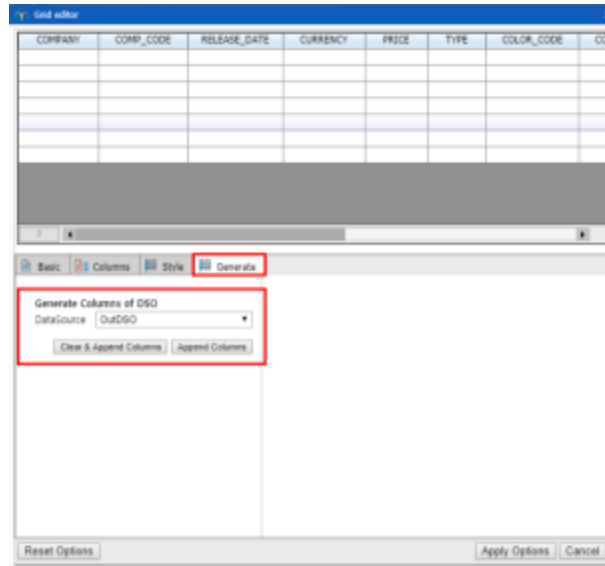


Style Tab

- ✓ Style Name : Style의 이름 설정
- ✓ Header : Header의 스타일
- ✓ Color : Style의 색상
- ✓ Font Size : Header의 글꼴 크기
- ✓ Cell : Cell의 스타일
- ✓ Color : Cell의 색상
- ✓ Font Size : Cell의 글꼴 크기
- ✓ Add : 새로운 Style을 추가
- ✓ Edit : 선택된 이름의 Style 수정
- ✓ Delete : 선택된 이름의 Style 삭제

■ Grid Editor / Generate Tab

DSO와 연동하여 column을 자동 생성합니다.



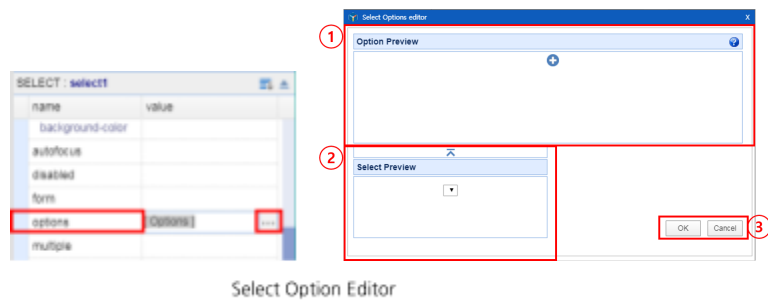
Generate Tab

- ✓ DataSource : column을 자동 생성할 시 사용할 DSO 선택
- ✓ Clear & Append Columns : 기존의 column을 모두 지우고 DSO 내의 column을 생성
- ✓ Append Columns : 기존의 column에 DSO 내의 column을 추가

■ Select Option Editor

Select Tag의 Option tag를 쉽게 입력할 수 있도록 도와줍니다.

select 객체의 Property Editor에서 options 부분의 버튼을 클릭하면 Select Option Editor가 팝업됩니다.



① Option Preview

option tag를 생성, 수정, 삭제합니다.

② Select Preview

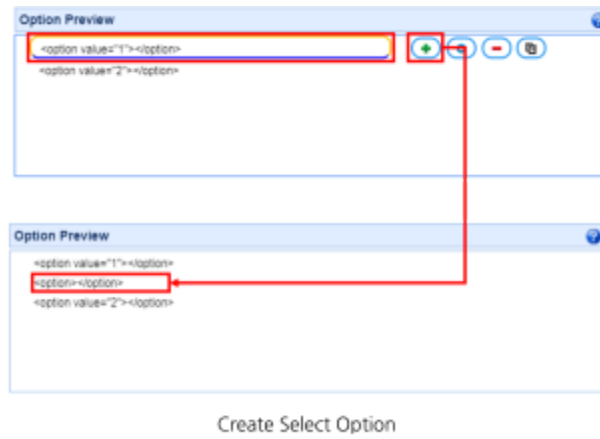
Option Preview에서 설정한 값에 맞춰 생성된 select tag를 보여줍니다.

③ Confirm Option

설정된 내용을 적용하거나 취소합니다.

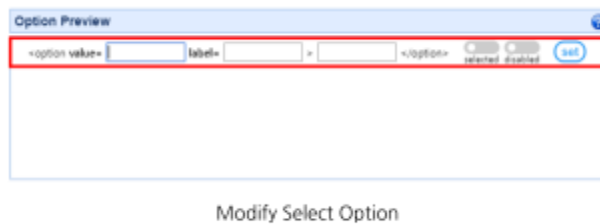
[Option Tag 생성]

- 버튼을 클릭하여 option tag를 생성합니다.
- option tag를 선택하고 생성 버튼을 클릭합니다.
- 해당 tag 아래에 생성될 위치가 파란색 선으로 표시됩니다.



[Option Tag 수정]

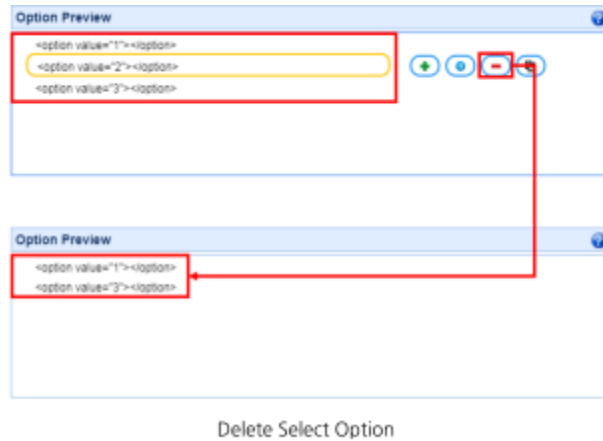
- 버튼을 클릭하여 option tag를 수정합니다.
- value, label, selected, disabled 등 다양한 값을 설정할 수 있습니다.



[Option Tag 삭제]

② 버튼을 클릭하여 option tag를 삭제합니다.

삭제하고 싶은 option tag를 선택하고 삭제 버튼을 클릭합니다.

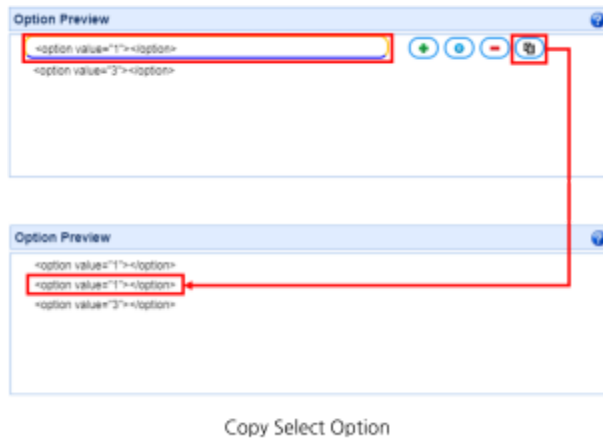


[Option Tag 복사]

③ 버튼을 클릭하여 option tag를 복사합니다.

복사하고 싶은 option tag를 선택하고 복사 버튼을 클릭합니다.

해당 tag 아래에 생성될 위치가 파란색 선으로 표시됩니다.



■ Chart Data Editor

Chart Component의 Data를 쉽게 입력할 수 있도록 도와줍니다.

Bar, Line, Radar, Pie, PolarArea, Doughnut chart에 해당하는 Chart Data Editor를 이용하여 Chart를 쉽게 그릴 수 있습니다. Chart Component의 Property Editor에서 data 부분의 버튼을 클릭하면 Chart Data Editor가 팝업됩니다.

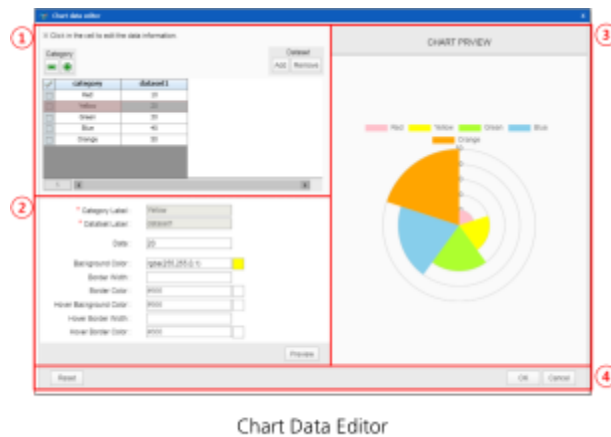


Chart Data Editor

① Data Grid

Chart의 data를 한 눈에 볼 수 있습니다.

상단 버튼을 이용하여 표를 편집하고, 셀을 클릭하여 데이터 정보 입력 영역을 띄웁니다.

② Data 편집

Chart의 data 정보를 변경할 수 있습니다.

③ Preview

Chart의 data를 변경하고 미리 볼 수 있습니다.

실제 적용되는 크기로 미리 보기 기능이 제공됩니다.

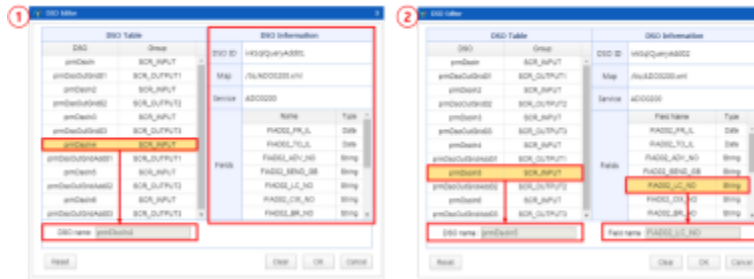
④ Button

- ✓ Preview : Chart Preview에서 변경한 data가 적용된 chart를 미리 보기
- ✓ Reset : Chart Data Editor 초기화
- ✓ Apply : 변경한 Data 적용
- ✓ Cancel : Chart Data Editor 닫기

■ DSO Editor

다양한 component와 Data Service의 연결을 도와줍니다.

DSO Editor를 이용하면 dsoName, fldName, valueField, labelField 속성을 쉽게 설정하고 적용할 수 있습니다. component의 Property Editor에서 위의 속성을 선택하면 DSO Editor가 팝업됩니다.



DSO Editor

① dsoName 설정

DSO Table에 해당 페이지와 연결된 Data Service의 정보가 출력됩니다.

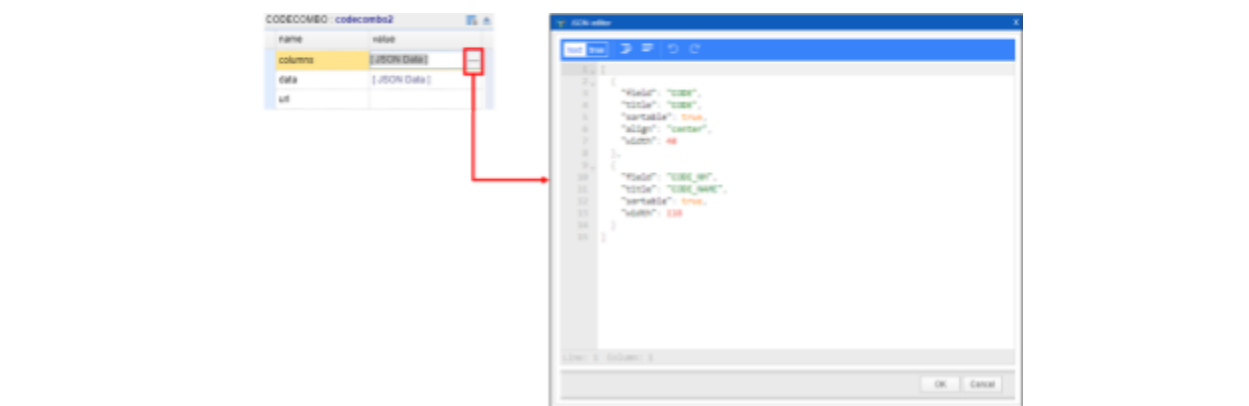
DSO를 클릭하면 해당 DSO의 정보가 표시되며, Field Name을 설정할 수 없습니다.

② fldName, valueField, labelField 설정

DSO Table에 해당 페이지와 연결된 Data Service의 정보가 출력됩니다.

Field Name이 활성화되어 DSO Name과 Field Name을 모두 설정할 수 있습니다.

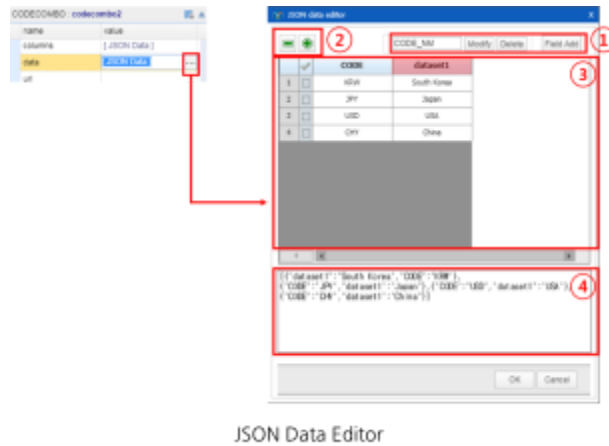
JSON 소스를 트리 형태로 보여주며 가독성을 높입니다. 또한 접기(collapse), 펴기(expand), Array 간 index 이동, 필드 삭제 등의 기능을 제공합니다. JSON Data는 Properties 창에서 [JSON Data] 라고 표시되며, Chart, CodeCombo, Grid 컴포넌트 등의 데이터 속성과 연결됩니다.



■ JSON Data Editor

객체의 JSON Data 편집을 지원합니다.

data를 표와 JSON 형태의 코드로 보여줍니다. JSON Data Editor를 이용하면 Grid와 CodeCombo 컴포넌트의 data를 쉽게 입력하고 설정할 수 있습니다.



① DataSet Control Button

Grid의 column을 생성하거나 삭제하고, Header 이름을 변경합니다.

② Row Control Button

Grid의 row를 추가하거나 삭제합니다.

③ Data Edit Area

data를 Grid 형식의 표로 보여줍니다.

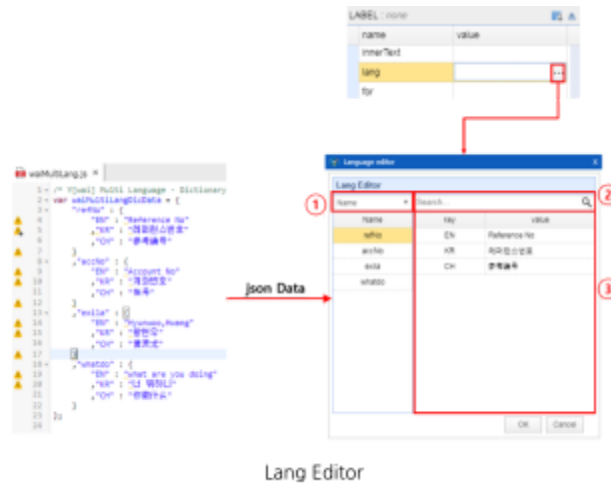
Cell과 Header를 클릭하여 data를 변경할 수 있습니다.

④ Preview Area

Data Edit Area에서 변경한 data를 JSON 형태의 코드로 변환하여 보여줍니다.

■ Lang Editor

객체의 다국어 처리하는 data를 Lang 속성과 연결합니다.



① 검색 조건

Lang data의 Name과 Name의 값(key)으로 검색 조건을 설정합니다.

② 검색

검색 기능을 제공합니다.

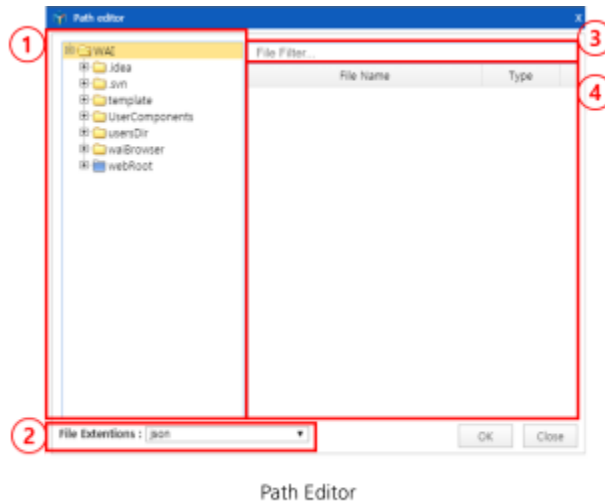
③ 다국어 데이터 목록

data의 값을 읽어 목록으로 보여주며 가독성을 높입니다.

■ Path Editor

URL 경로를 쉽게 입력할 수 있도록 도와줍니다.

객체의 Property Editor에서 url 부분의 버튼을 클릭하면 Path Editor가 팝업됩니다.



왼쪽 목록에서 디렉토리를 선택하면 오른쪽에 파일 목록이 나타납니다.
원하는 파일을 클릭하여 URL을 연결합니다.

① Directory List

프로젝트의 디렉토리 항목을 Tree 형식으로 보여줍니다.

② File Extensions

Property에 맞춰 연결할 확장자를 선택하여 볼 수 있습니다.

③ Search Input

검색값에 해당하는 File Name을 입력할 수 있습니다.

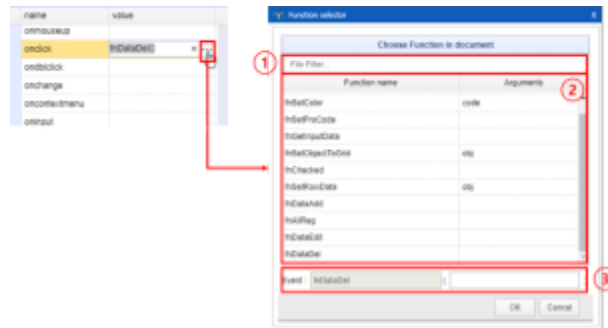
① File List

선택한 디렉토리 파일의 목록을 보여줍니다.

■ Script Editor

객체의 Event를 설정합니다.

기존에 script로 작성한 함수(function)를 목록으로 보여줍니다.



Script Editor

① 검색

함수명으로 검색할 수 있습니다.

② Function List

기존에 등록된 Event 함수를 목록으로 보여줍니다.

③ Event

Function List에서 선택한 함수를 보여줍니다. 인자값을 수정할 수 있습니다.

[함수명 생성 규칙]

함수의 Event명에서 on을 제외하고 tag(component)의 id 혹은 이름을 붙여 생성합니다.

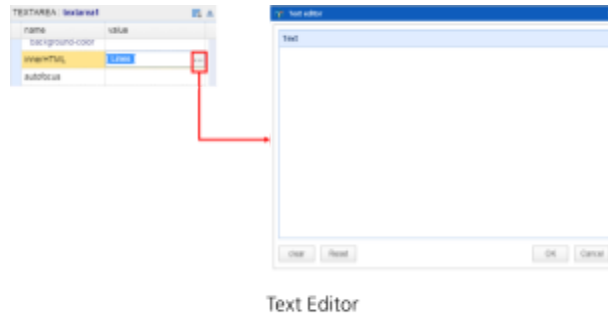
함수명이 중복할 시 _index를 붙여 구분합니다. (divBlur_1, divBlur_2...)

	Tag id가 있음	Tag id가 없음
onevent	id + Event	Tag명 + Event
함수명 중복 시	id(TagName) + Event + _number	

■ Text Editor

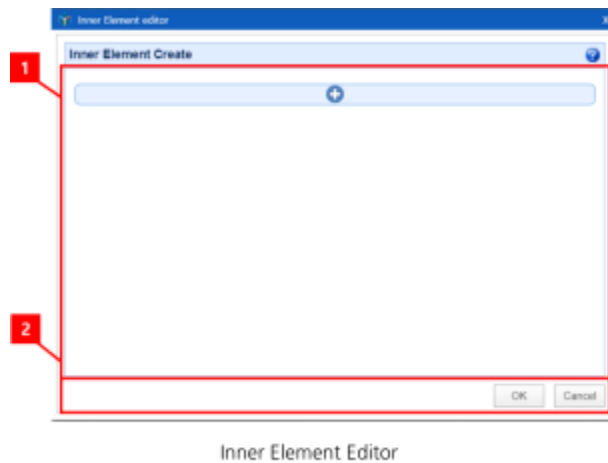
여러 줄의 Text를 입력하고 적용합니다.

<textArea> 같은 태그의 innerText 속성과 연결됩니다.



■ Inner Element Editor

video, audio, object, external Tag 등의 객체에 Inner Element를 쉽게 적용할 수 있도록 도와줍니다. 객체의 Property Editor에서 elements 부분의 버튼을 클릭하면 Inner Element Editor가 팝업됩니다.



① Edit Elements

elem 태그를 생성, 수정, 삭제합니다.

② Confirm Elements

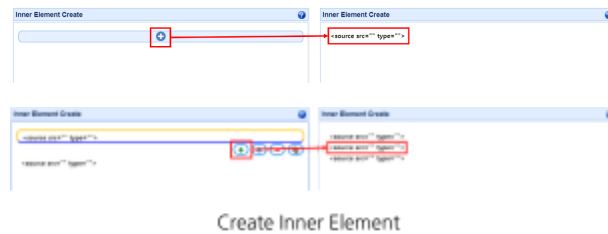
작성한 내용을 확인하거나 취소합니다.

[Element 생성]

 또는  버튼을 클릭하여 element를 생성합니다.

객체를 선택하고 생성 버튼을 클릭합니다.

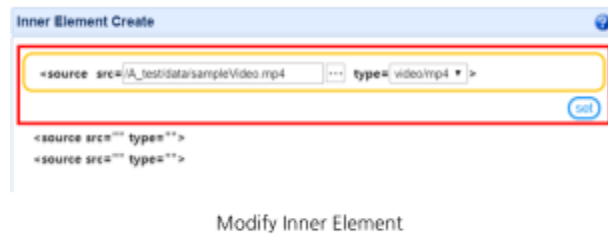
해당 element 아래에 생성될 위치가 파란색 선으로 표시됩니다.



[Element 수정]

 버튼을 클릭하여 element를 수정합니다.

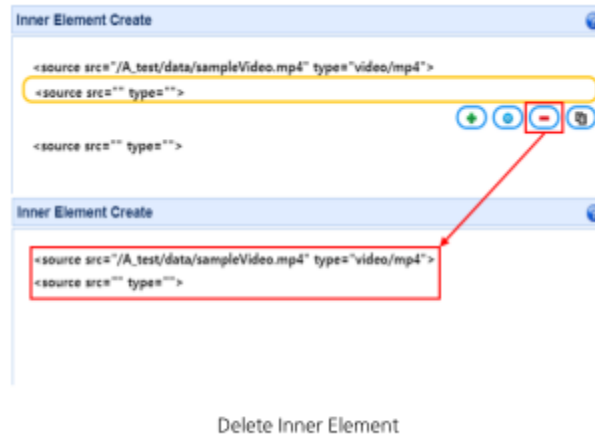
해당 객체의 속성을 설정할 수 있습니다.



[Element 삭제]

 버튼을 클릭하여 element를 삭제합니다.

삭제하고 싶은 element를 선택하고 삭제 버튼을 클릭합니다.



[Element 복사]

 버튼을 클릭하여 element를 복사합니다.

복사하고 싶은 element를 선택하고 복사 버튼을 클릭합니다.

해당 element 아래에 생성될 위치가 파란색 선으로 표시됩니다.

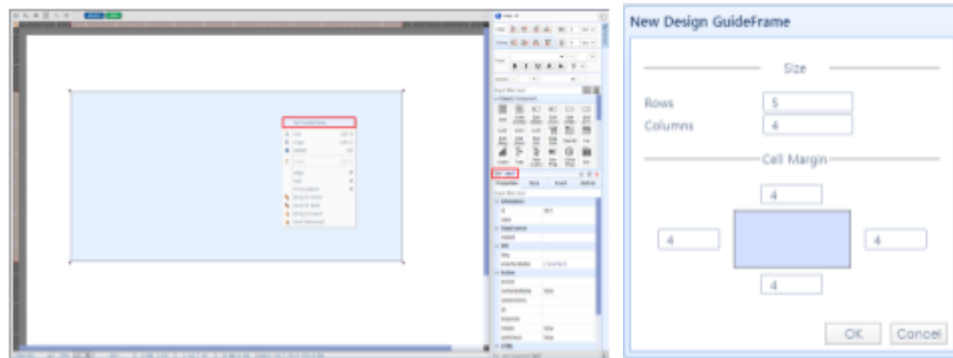


3. Y[wai] IDE

A. WYSIWYG Editor

d. GuideFrame

화면을 구성할 때 컴포넌트를 위치시킬 프레임을 미리 설정하여 화면을 쉽게 만들 수 있습니다.



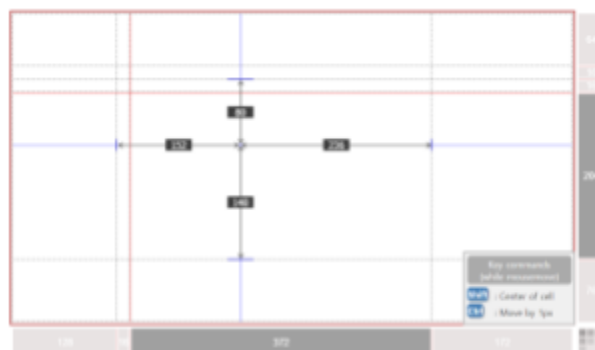
Set GuideFrame 선택

GuideFrame 디자인

GuideFrame은 div 컴포넌트에서 적용할 수 있습니다.

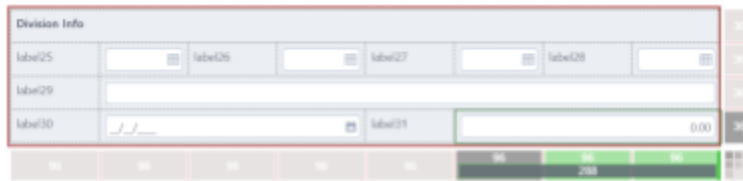
div 컴포넌트에서 마우스 우클릭을 하고 Set GuideFrame를 선택하면 GuideFrame 생성 창이 팝업됩니다.

행과 열, 여백 등의 구성 정보를 입력한 뒤 [OK] 버튼을 누르면 GuideFrame이 생성됩니다.



마우스로 행과 열 조정

GuideFrame이 생성되면 마우스로 셀과 행, 열을 조정할 수 있습니다.



GuideFrame의 컴포넌트 배치

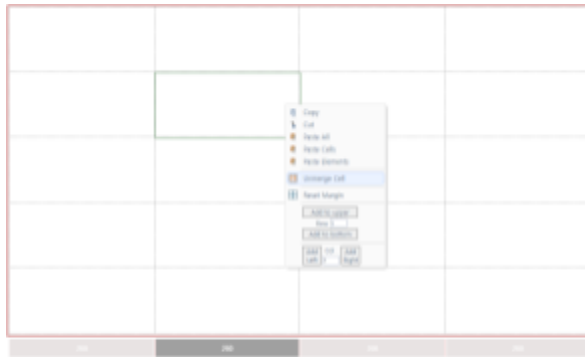
한 셀에 하나의 컴포넌트만 추가할 수 있습니다.

추가할 컴포넌트를 선택하고 GuideFrame의 원하는 셀에 위치시켜 컴포넌트를 추가합니다.

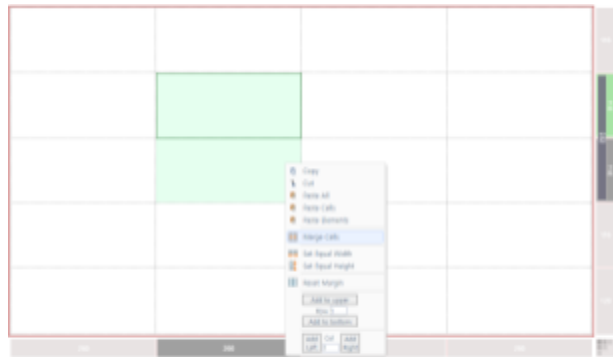
GuideFrame이 적용된 div 컴포넌트 혹은 GuideFrame의 셀에 추가한 컴포넌트는 더블클릭하여 선택할 수 있습니다.

GuideFrame을 이용하여 화면을 구성하고 저장하면 편집하고 있는 html 파일과 동일한 경로에 확장자가 gf인 파일이 동일한 이름으로 생성됩니다.

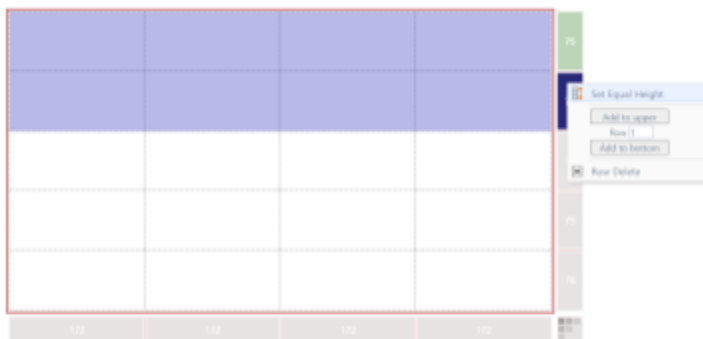
해당 gf 파일은 html 파일에 적용한 GuideFrame 정보를 저장하거나 불러오는 데 사용됩니다.



GuideFrame 편집
(셀을 한 개 선택할 경우)



GuideFrame 편집
(셀을 두 개 이상 선택할 경우)



GuideFrame 편집
(롤러를 통해 행을 선택할 경우)



GuideFrame 편집
(롤러를 통해 열을 선택할 경우)

- ① Copy
선택한 셀(내부 요소 포함)을 복사합니다.
- ② Cut
선택한 셀(내부 요소 포함)을 잘라냅니다.
- ③ Paste All
복사 또는 잘라낸 셀과 셀의 내부 요소를 붙여 넣습니다.
- ④ Paste Cells
복사 또는 잘라낸 셀을 붙여 넣습니다.
- ⑤ Paste Elements
복사 또는 잘라낸 셀의 내부 요소를 붙여 넣습니다.

⑥ Merge/Unmerge Cell

셀을 병합하거나 분리합니다.

⑦ Reset Margin

여백을 재설정합니다.

⑧ Set Equal Height/Width

선택한 셀의 넓이와 높이를 동일하게 설정합니다.

⑨ Row/Column Add

행과 열을 추가합니다.

⑩ Row/Column Delete

선택한 행과 열을 삭제합니다.

4. Y[wai] Component 및 HTML, HTML5

A. 기본 설명 및 사용 방법

Y[wai] Component는 HTML 태그 형태로 제공됩니다.

따라서 HTML 표현 방법과 JavaScript 처리 방법이 같습니다.

■ 기본 설명

Y[wai] Component는 HTML 태그와 같은 방식으로 HTML에 표현됩니다.

| 형식 | 예시 |
|--|---|
| <code><태그명 속성="속성값" ... ></태그명></code> | <code><edit id="edit1" type="currency"></edit></code> |

태그 내부에 데이터 혹은 옵션을 가질 수 있는 component는

아래와 같은 방식으로 HTML에서 표현됩니다.

| 형식 | 예시 |
|---|--|
| <code><태그명 속성="속성값" ... ></code>
{ Option : JSON Data }
<code></태그명></code> | <pre><codecombo id="combo1"> "columns" : [[{"field":"id","title":"key","width":80,"sortable":true,"align":"left"}, {"field":"text","title":"value","width":80}, {"field":"status","title":"비고","width":50}], "data" : [{"text":"FI-SW-01","id":"EST-1","status":"Y"}, {"text":"FL-DLH-02","id":"EST-16","status":"Y"}, {"text":"K9-DL-01","id":"EST-10","status":"Y"}]] </codecombo></pre> |

component 태그 내부에 JSON Data를 표현할 시 최초의 중괄호는 생략합니다. ({ })

component별로 내부 형식이 다를 수 있습니다. Component 도움말을 참고하거나 Y[wai] IDE의 속성 편집기에서 작성하기 바랍니다.

■ 사용 방법

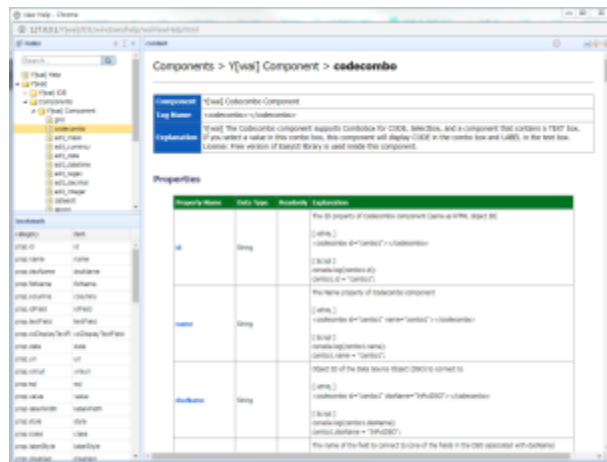
Y[wai] Component는 JavaScript에서 해당 component의 id를 이용하여 제어할 수 있습니다. HTML, HTML5 객체를 제어하는 방법과 같습니다. component별로 속성과 메소드, 이벤트가 다르기 때문에 Component 도움말을 참고하여 사용하기 바랍니다.

Y[wai] IDE를 사용하여 해당 component를 선택하면 Property Editor에 속성과 메소드, 이벤트가 표시되며 값을 입력하거나 수정할 수 있습니다.

형식	예시
객체ID.속성 = 속성값; 객체ID.메소드명(메소드 인자);	<pre><script type="text/javascript"> function btnClick() { <u>Edit1.value = '123-45-6789';</u> <u>Edit1.focus();</u> } </script></pre>

Component 예제에서 component별 다양한 예제를 참고할 수 있습니다.

또한 Y[wai] IDE에서 Help 메뉴를 클릭하면 웹에서 도움말 페이지가 열립니다. component별로 자세한 내용을 확인할 수 있습니다.



Help

4. Y[wai] Component 및 HTML, HTML5

B. Component 소개

■ Y[wai] Component

① Grid

	grid 형태로 데이터를 입출력합니다. option에 따라 Editing 기능을 제공합니다.
예시	<pre><grid id="grid1" dsoName="gridData"> { option : JSON Data } </grid></pre>

② CodeCombo

	Grid에서 특정 row를 선택하여 입력할 수 있습니다. option으로 제공되는 읽기 전용 입출력 컨트롤을 이용해 코드 설명을 표시할 수 있습니다.
예시	<pre><codecombo id="cc1" xmlurl="/xmlldb/F829.xml"> { option : JSON Data } </codecombo></pre>

③ Edit

	Type 속성에 따라서 Mask, Currency, Date, DateTime, Regex, Decimal, Integer 형태로 데이터를 입출력합니다. option에 따라 입력이 자동으로 제한되며 입력값에 관한 오류가 발생하지 않습니다.
예시	<pre><edit id="edit1" type="mask" mask="999-99-9999"></edit></pre>

④ DateEdit

	Edit 컴포넌트의 Type="date"와 같은 입력 방식을 제공합니다. 달력을 팝업으로 표시하는 버튼을 가지고 있습니다. Lang 속성으로 달력의 언어를 선택할 수 있습니다.
예시	<code><dateedit id="de1" lang="Korean"></dateedit></code>

⑤ tabControl

	Tab type을 지원합니다.
예시	<code><tabControl id="tabControl1"> <tabPage id="tabPage0" style="display: block;"></tabPage> <tabPage id="tabPage1" style="display: none;"></tabPage> </tabControl></code>

⑥ Calendar

	달력을 지원합니다. 날짜를 선택하고 설정할 수 있습니다. Lang 속성으로 달력의 언어를 선택할 수 있습니다. Date를 설정하지 않으면 시스템 날짜로 설정됩니다.
예시	<code><calendar id="cal1" lang="Korean"></calendar></code>

⑦ Chart

	Type 속성에 따라서 Bar, Line, Radar, PolarArea, Pie, Doughnut 형태로 데이터를 표시합니다. option에 따라 Tooltip, Animation 등의 기능을 제공합니다.
예시	<pre><chart id="chart1" type="bar"> { option : JSON Data } </chart></pre>

⑧ TreeView

	Tree 구조의 리스트 서식을 지원하며, data를 Tree 구조로 표시합니다. JSON 형태의 url으로 Tree의 초기값을 설정할 수 있습니다. option에 따라 Animation, ContextMenu 등을 제공합니다.
예시	<pre><treeview id="tree1" url="testData/tree_data2.json"> { option : JSON Data } </treeview></pre>

⑨ cTree

	Tree 구조의 리스트 서식을 지원하며, data를 Tree 구조로 표시합니다. JSON 형태의 url으로 Tree의 초기값을 설정할 수 있습니다. TreeView와 다르게 Canvas 기반으로 개발되어 객체 생성 속도가 우수합니다.
예시	<pre><ctree id="tree1" url="testData/tree_data2.json"> { option : JSON Data } </ ctree ></pre>

⑩ BarProgress

	bar 형태의 Progress를 지원합니다. 0~100까지의 Value값을 Animation으로 표시합니다.
예시	<code>< bprogress id="bp1" value="65"></bprogress></code>

⑪ CircleProgress

	원 형태의 Progress를 지원합니다. 0~100까지의 Value값을 Animation으로 표시합니다.
예시	<code>< cprogress id="cp1" value="78"></cpprogress></code>

⑫ External

	3rd Party Library를 지원합니다. 객체를 정의하고 3rd Party Library를 적용시킵니다.
예시	<pre> <external id=" external1" createType="elements"> 두 가지 태그 방식으로 객체를 정의합니다. 1. <elem tag="table"></elem> <elem tag="div" id="plist47"></elem> 2. <custHtml> <table></table> <div id="plist47"></div> </custHtml> </external> </pre>

⑬ SubFrame

	다른 HTML 페이지를 호출해서 부분 화면으로 사용합니다.
예시	<code><subFrame id="subFrame1" screen="/biz/dp/DP10.html"></subFrame></code>

⑭ SearchEdit

	인풋박스(wai edit component)와 버튼이 결합된 컴포넌트입니다. 데이터를 입력, 출력, 수정할 수 있습니다.
예시	<code><searchEdit id="searchEdit1"></searchEdit></code>

⑮ ImageButton

	이미지 영역과 버튼이 결합된 컴포넌트입니다.
예시	<code><imgButton id="imgButton1"></imgButton></code>

■ HTML Standard

① Button

	button을 생성합니다. 이미지나 텍스트를 포함하여 유연하게 디자인할 수 있습니다.
예시	<code><button id="btn1" onclick="func();" ></button></code>

② Div

	문서의 영역을 지정합니다. id 속성 혹은 class 속성을 이용하여 식별자를 지정합니다. 같은 섹션에 포함된 여러 개의 문단을 묶어 주석을 달 때 유용하게 사용할 수 있습니다.
예시	<code><button id="btn1" onclick="func();" ></button></code>

③ Form

	웹 페이지가 상호작용할 수 있도록 form의 최상위 요소로 범위를 지정합니다.
예시	<code><form action="demo_form.asp" method="get"> First name: <input type="text" name="fname">
 Last name: <input type="text" name="lname">
 <input type="submit" value="Submit"> </form></code>

④ Iframe

	HTML 문서에서 다른 HTML 문서를 임의의 위치에 보여줍니다. 웹 문서 중간에 다른 웹 문서나 메모장 같은 내용을 원하는 크기로 불러들여 보여줍니다.
예시	<code><iframe src="http://www.corebank.co.kr"></iframe></code>

⑤ Img

	문서에 그림을 삽입합니다. 이미지에 다른 문서를 링크할 시에는 <a> 태그 안에서 사용할 수 있습니다.
예시	<code></code>

⑥ Input

	데이터를 입력할 수 있는 입력 field를 지정합니다. 입력 field는 Type에 따라 다양한 방식으로 변할 수 있습니다.
예시	<code><input id="in1" type="text" value="input01"></code>

⑦ Label

	사용자 인터페이스에 label을 부여합니다. label의 for 속성에 컨트롤의 id 속성과 동일한 값을 지정하여 label과 컨트롤을 동기화합니다. label의 범위에 텍스트와 컨트롤을 포함시킵니다.
예시	<code><label for="male">Male</label></code>

⑧ Object

	이미지, 오디오, 자바 애플릿, 플래시 같은 object를 표시합니다. data 속성이나 type 속성 중 하나는 반드시 존재해야 합니다.
예시	<code><object width="400" height="400" data="helloworld.swf"></object></code>

⑨ Select

	선택 목록(drop-down list)을 생성합니다. option을 한 개 이상 포함해야 합니다. <optgroup> 태그를 이용하여 항목을 그룹화할 수 있습니다.
예시	<code><select id="sel1"> <option value="volvo">Volvo</option> <option value="saab">Saab</option> </select></code>

⑩ Span

	인라인 요소로 사용됩니다. class, lang, dir 등의 전역 속성과 함께 사용하면 유용합니다.
예시	<code>blue</code>

⑪ TextArea

	여러 줄의 text를 편집할 수 있는 컨트롤을 생성합니다. 초기값의 text는 코드 상의 공백 문자나 줄 바꿈이 그대로 반영됩니다.
예시	<code><textarea rows="4" cols="50"></code> option : TEXT Data <code></textarea></code>

⑫ Table

	HTML Table을 정의합니다. <table> 요소와 하나 이상의 <tr>, <th>, <td> 요소로 구성됩니다. <tr>은 table 행, <th>는 table 헤더, <td>는 table 셀을 정의합니다.
예시	<code><table border="1"></code> <code><tr><th></th></tr></code> <code><tr><td></td></tr></code> <code></table></code>

■ HTML5 Standard

① Canvas

	스크립트와 해상도에 의존하는 비트맵 캔버스를 제공합니다. 그래프, 게임 그래픽, 비주얼 이미지를 실시간으로 렌더링합니다.
예시	<code><canvas id="myCanvas"></canvas></code>

② Video

	비디오와 미디어를 재생합니다. 플러그인 없이 브라우저 자체에서 동영상 기능을 지원합니다.
예시	<code><video id="video1" src="/A_test/data/sampleVideo.mp4"></video></code>

③ Audio

	사운드 파일을 웹 페이지에 삽입합니다. 음악 또는 기타 오디오를 웹 페이지에 추가할 때 유용합니다.
예시	<code><video id="video1" src="/A_test/data/sampleVideo.mp4"></video></code>

④ Embed

	브라우저에 설치되어 있고 플러그인이 필요한 외부 애플리케이션을 실행합니다.
예시	<code><embed src="helloworld.swf"></code>

⑤ Progress

	작업의 진행 정도를 표시합니다. 현재 값과 최대값을 인라인으로 포함시켜 진척도를 알 수 있습니다.
예시	<pre><table border="1"> <tr><th></th></tr> <tr><td></td></tr> </table></pre>

5. Y[wai] Component 도움말

A. Grid

Canvas(이미지) grid 형태로 데이터를 입출력합니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Grid Component의 id (HTML 객체 id와 동일)
dsoName	문자열		연결할 DSO(data source object)의 객체 id
style	문자열		component의 style 속성
option	가변형		grid에 설정할 JSON 형태의 option 속성
data	가변형		grid에 설정할 JSON 형태의 data
editorMode	가변형		component의 grid 편집 (true: 활성화 / false: 비활성화)
FormatString	문자열		grid column의 text 값 설정 ex) ^Ref No;^Cust No;<Customer Name
ColumnsWidth	문자열		grid column의 넓이 설정, 구분자 ex) 120px;55px;110px
FieldMapping	문자열		grid에 연결할 field 이름, 구분자 ex) REF_NO;BASE_CIX_NO;BASE_CUST_ENM

■ Methods

함수명	설명	아규먼트	리턴
exportExcel	현재 grid의 내용을 excel 파일로 저장	fileName (문자열) (excel 파일명)	파일 (변환이 완료되면 excel 파일 출력)
AddColumn	grid에 표시되는 column을 append	data (문자열) (추가할 column 값)	
DelColumn	grid에 표시되는 column을 delete	columnName (문자열) (삭제할 column명)	
getCheckeRows	grid 행이 선택되어 있는지 확인	column (정수형) (설정할 column의 index)	불린형 (선택 여부)
getCheckedRows	grid의 Select에서 선택된 행의 data를 가져옴		문자열 (선택된 행의 data)
setCheckAll	grid의 모든 행을 선택		
setUnCheckAll	grid의 모든 행을 선택 취소		
getSelection	grid에서 선택된 행의 data를 가져옴		문자열 (선택된 행의 data)
setColumnVisible	grid에 표시되는 column의 Visible 설정	column (정수형) (설정할 column의 index) visible (불린형) (visible 설정)	
setColumnOption	grid에 표시되는 column의 Editor Option 설정	column (정수형) (설정할 column의 index) option (문자열) (option 설정)	
SetRecordset	grid에 표시되는 data 설정	rs (문자열) (새로운 grid data)	
SyncToDSO	레코드 선택 위치를 DSO 기준으로 정렬		
SyncToGRID	레코드 선택 위치를 grid 기준으로 정렬		
getSelectIndex	현재 선택된 row의		정수형

	index를 가져옴		(선택된 row의 index)
SetValue	현재 선택된 row의 field에 값을 설정	fieldName (문자열) (설정할 field 이름) fieldValue (문자열) (변경할 값)	
GetValue	현재 선택된 row의 field에 값을 가져옴	fieldName (문자열) (설정할 field 이름)	문자열 (field의 값)
Refresh	grid를 다시 그림		
SelectRow	지정한 row를 선택	RowNo (정수형) (선택할 row의 index)	
EditPost	grid에서 수정한 내용을 DSO에 반영		
SetColumnEditing	지정한 column의 editable 설정	column (정수형) (설정할 column의 index) editable (불린형) (선택할 row의 index)	
GetData	grid의 data를 JSON 형태로 반환		문자열 (grid의 data)
SetData	grid의 data를 JSON 형태로 설정	fieldName (문자열) (설정할 field 이름)	
GetGrid	grid의 option과 data를 반환		문자열 (grid의 option과 data)
SetGrid	grid의 option과 data를 JSON 형태로 설정		
clear	grid data 삭제		
reset	grid 초기화		
GetLookupValue	xml에 넘긴 value와 일치하는 orgFldName의 newFldName을 가져옴	xmlid (문자열) (xml id) orgFldName (문자열) (찾을 field 이름) NewFldName (문자열) (가져올 field 이름)	문자열 (해당 field의 값)

		value (문자열) (찾을 값)	
moveNextEditColumn	현재 선택된 column 이후에 edit할 수 있는 column으로 이동		
AddRow	끝에 row를 추가		
AddPreRow	현재 선택된 row 이전에 새로운 row를 추가		
AddNextRow	현재 선택된 row 이후에 새로운 row를 추가		
RecordCount	grid의 RecordCount를 반환		정수형 (grid의 RecordCount)

■ Events

이벤트명	설명
onclick	특정 cell을 클릭하면 발생
ondblclick	특정 cell을 더블클릭하면 발생
oneditchanged	grid에서 특정 cell의 값이 변경되면 발생
oneditvaluechanged	grid에서 특정 cell의 값이 변경되면 발생
oncolumnheaderclick	grid에서 특정 column을 클릭하면 발생
onrecordchange	grid에서 이전과 다른 row를 클릭하면 발생
onfocusedrecordchanged	grid에서 이전과 다른 row를 클릭하면 발생
onrecordclick	grid에서 특정 row를 클릭하면 발생

■ Options

Grid Component 태그 내부에 설정되는 JSON 형태의 grid option입니다.

grid 설정은 다른 component보다 내용이 많기 때문에 태그 속성(Attribute) 형태가 아니라 태그 내부에 설정하는 형태로 개발되었습니다.

```
<grid id="grid1" dsoName="gridData">
  option : '옵션값',
  style : {
  },
  columns : [
    { 컬럼 설정 1 },
    { 컬럼 설정 2 },
    { 컬럼 설정 3 }
  ]
</grid>
```

option 항목과 style 항목은 생략할 수 있습니다. 이러한 경우에는 기본값으로 설정됩니다.

column은 반드시 설정해야 하며, 각 column은 object 형태입니다.

option	유형	설명
fixColumn	정수형	가로 스크롤 고정 column의 수 (좌측 기준) (default: 0) 가로 스크롤 고정 column이 없는 설정

■ Style

style은 JSON의 object 형태로 선언할 수 있습니다.

선언한 style을 특정 column의 style로 지정할 수 있으며, 한 style을 여러 column에서 style로 지정할 수 있습니다.

```
“style” : {  
  
  “custCol1” : {  
    “header” : {  
      “color” : [ '#EAEAFF', '#A0A0FF', '#4040AA' ],  
      “font” : “normal 12px arial blue”  
    },  
    “cell” : {  
      “color” : [ '#FFFFFF', '#A0FFA0', '#EAFFEA' ],  
      “font” : “normal 11px arial darkgrey”  
    }  
  }  
}
```

이처럼 선언된 style을 column 설정에서 “style” : “custCol1”로 지정할 수 있습니다.

grid / styles / default.css의 style을 수정하면 별도의 style을 선언하지 않아도 grid의 전체 style을 모두 변경할 수 있습니다.

■ Column

grid의 column을 설정합니다.

column이 선언된 순서대로 좌측부터 선언되며, 최초로 선언된 column이 가장 좌측에 생성됩니다.

각 column은 type 속성을 가집니다.

type이 group인 경우 하위에 child column을 가질 수 있습니다.

column 속성	유형	설명
type	문자열 (option)	group 또는 column의 값으로 설정 group으로 설정된 경우 하위에 child column 설정 가능 (default: column)
text	문자열	column의 head에 표시되는 text
width	정수형	column의 좌우 너비 (단위: pixel)
orientation	문자열 (option)	column의 type이 group인 경우 column을 묶는 방향 설정 vertical 또는 horizontal로 설정 가능 (default: horizontal)
field	문자열	column에 연결된 fldName (dsoName에 선언된 DSO의 field 이름 중 하나 지정)
visible	불린형	column의 표시 여부 설정 (default: true)
alignment	문자열	column의 head에 표시되는 text를 정렬 (default: left) (right, center 설정 가능)
editor	가변형	column의 cell의 화면 표시 및 편집 editor 설정 설정이 없는 경우 text editor로 지정됨

■ Column Editor

grid의 각 column은 화면 표시 여부와 편집 editor를 지정할 수 있습니다.

지정된 editor에 따라서 설정값이 다르며, 기본 속성은 아래와 같습니다.

editor 속성	유형	설명
type	문자열 (option)	editor 유형 선택 text, date, mask, currency, combo, calculator, LookupComboBox 중 선택 가능 (default: text)
alignment	문자열	column의 cell에 표시되는 data를 정렬 (default: left) (right, center 설정 가능)

```
columns: [
  {text:"1",field:"field1",width:30,"visible":true,"style":"custCol1"},
  {text:"2",field:"field2",width:60,"visible":true,"style":"custCol1"},
  {text:"3",field:"field3",width:50,"visible":true,"style":"custCol1"},
  {
    "type":"group",
    "text":"직원정보",
    "width":220,
    "orientation":"vertical",
    "columns":[
      {
        "type":"group",
        "text":"기본정보",
        "visible":true,
        "columns":[
          {text:"성명",field:"field1","alignment":"left","editor":{"type":"text"}},
          {text:"직급",field:"field2","visible":true,"editor":{"type":"text","alignment":"center"}},
          {text:"ID",field:"field3","alignment":"right","editor":{"type":"text","alignment":"right"}}
        ]
      },
      {text:"부서명",field:"field4"}
    ]
  },
  {text:"상태",field:"field5",width:60,"visible":true,"style":"custCol1"},
  {
    "type":"group",
    "text":"프로젝트",
    "width":360,
    "height":90,
    "columns":[
      {text:"프로젝트명",field:"field6","width":140},
      {text:"담당업무",field:"field7"},
      {
        "type":"group",
        "text":"작업진행",
        "columns":[
          {text:"부분",field:"field8"},
          {text:"전체",field:"field9","style":"custCol2"}
        ]
      }
    ]
  },
  {
    "type":"group",
    "text":"근무시간",
    "width":75,
    "orientation":"vertical",
    "columns":[
      {text:"아근",field:"field10","sortType":"number"},
      {text:"휴가Win연차",field:"field11"}
    ]
  }
]
```

Column Editor Example

5. Y[wai] Component 도움말

B. CodeCombo

Code용 Combobox와 Selectbox를 지원하며 Textbox를 포함합니다.

Combobox에서 값을 선택하면 Code는 Combobox 안에, Label은 Textbox 안에 표시됩니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		CodeCombo Component의 id (HTML 객체 id와 동일)
name	문자열		codecombo의 name 속성
dsoName	문자열		연결할 DSO의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
columns	가변형		combobox의 panel에 설정할 column
idField	문자열		combobox의 CODE가 되는 field (default: CODE)
textField	문자열		combobox의 LABEL이 되는 field (default: CODE_NM)
cbDisplayTextField	불린형		LABEL field가 없으면 TRUE TRUE일 경우 textField가 combobox에 표시됨
data	가변형		combobox의 panel에 설정할 JSON 형태의 data
url	문자열		combobox의 panel에 data를 설정하기 위한 JSON 파일의 URL
xmlurl	문자열		combobox의 panel에 data를 설정하기 위한 XML 파일의 URL
sql	문자열		combobox의 panel에 data를 설정하기 위한 SQL statement
value	문자열		codecombo의 현재 표시되는 값
labelWidth	문자열		textField가 표시되는 label textbox의 Width 고정 길이

style	문자열		codecombo의 현재 style
class	문자열		codecombo의 현재 class
labelStyle	문자열		textbox의 현재 style
disabled	불린형		codecombo의 disable (true: 비활성화 / false: 활성화)
readonly	불린형		codecombo의 readonly (true: 비활성화 / false: 활성화)
editable	불린형		CodeCombo Component의 combobox 편집 (true: 비활성화 / false: 활성화)
editType	문자열		codecombo의 combobox 편집 유형 (All, Number, Uppercase, Lowercase)
panelWidth	문자열		combobox의 panel 가로 길이 (default: component 길이)
panelHeight	문자열		combobox의 panel 세로 길이 (default: 150px)
showHeader	불린형		combobox의 grid header (true: 보임(default) / false: 보이지 않음)
fitColumns	불린형		combobox의 grid의 각 field 넓이를 panel 넓이에 맞춰 자동으로 조절 (default: false)
btnImg	문자열		codecombo의 combo button 이미지 URL
dropdownkeyCode	문자열		특정 Keycode로 combobox의 panel을 Dropdown
title	문자열		codecombo의 title 속성
tabindex	정수형		codecombo의 tabindex 속성
itemCount	정수형	읽기 전용	현재 combobox에 설정된 data의 수
codeId	문자열	읽기 전용	정보성 속성
codeName	문자열	읽기 전용	정보성 속성

■ Methods

함수명	설명	아규먼트	리턴
focus	codecombo로 focus 이동		
getValue	codecombo의 선택된 값(CODE)을 가져옴 다중 선택을 한 경우 모든 값을 문자열 형태로 가져옴		문자열 (codecombo의 선택된 값(CODE)을 반환)
getSelectIndex	codecombo의 선택된 값의 index를 정수 형태로 가져옴		정수형 (codecombo의 선택된 값의 index를 반환)
getSelected	codecombo의 선택된 row의 정보를 object 형태로 가져옴		가변형 (codecombo의 선택된 row의 정보를 반환)
getText	codecombo의 선택된 값(CODE)의 label 이름(CODE_NM)을 가져옴 다중 선택을 한 경우 모든 값의 label 이름을 문자열 형태로 가져옴		문자열 (codecombo의 선택된 값의 label 이름을 반환)
getData	codecombo combobox의 grid data를 가져옴		가변형 (codecombo combobox의 grid에 현재 설정된 data를 반환)
setValue	codecombo에 임의의 값(CODE)을 입력 다중 선택이 가능한 codecombo의 경우 모든 값을 문자열 형태로 입력	value (문자열) (입력할 임의의 값(CODE))	
setSelectIndex	codecombo에 임의의 index를 입력	index (문자열) (임의의 index number)	
setData	codecombo combobox에 새로운 grid data를 입력	data (가변형) (새로운 grid data)	
clear	codecombo combobox의 새로운 grid data 모두 삭제 url, xmlurl, sql,data에 입력된 값이 모두 삭제됨		
addItem	codecombo combobox에 새로운 data를 append	code (문자열) (추가할 CODE 값)	

		label (문자열) (추가할 CODE 값의 표시 내용)	
deleteItem	codecombo combobox에 임의의 data를 delete	index (정수형) (삭제할 data의 index number)	
itemIndex	codecombo combobox에 임의의 data의 index number를 가져옴	code (문자열) (임의의 CODE 값)	숫자형 (codecombo의 임의의 값의 index number를 반환)
itemValue	codecombo combobox에 index로 임의의 data의 CODE를 가져옴 fieldName은 필수가 아님 (default: CODE_NAME)	index (문자열) (임의의 index number) fieldName (문자열) (임의의 field name)	문자열 (codecombo의 해당 index의 값(CODE)을 반환)
sort	codecombo의 data 중 임의의 field를 sort (default: asc sorting) (정렬 순서(sortOrder)를 입력하지 않을 경우)	fieldName (문자열) (정렬할 field name) sortOrder (문자열) (정렬 순서: asc/desc)	
hideColumn	codecombo combobox의 grid panel에서 임의의 field를 숨김	fieldName (문자열) (숨길 field name)	
showColumn	codecombo combobox의 grid panel에서 숨긴 임의의 field를 보여줌	fieldName (문자열) (보여질 field name)	

■ Events

이벤트명	아규먼트	설명
onclick		객체를 클릭하면 발생
ondblclick		객체를 더블클릭하면 발생
onkeydown		키를 누르는 순간 발생
onkeypress		키를 누른 상태에서 발생
onkeyup		키를 해제하면 발생
onmousedown		마우스 버튼을 아래로 누르면 발생
onmouseup		마우스 버튼을 클릭 후 놓으면 발생
onmouseover		마우스 커서가 객체 위로 이동하면 발생
onblur		객체가 focus를 잃으면 발생
onfocus		객체가 focus를 얻으면 발생
oncontextmenu		codecombo를 우클릭하면 발생
onSelect	evt.args.index (경수형) (grid panel에서 선택된 row의 index number) evt.args.rowData (가변형) (선택된 row의 정보)	codecombo combobox의 grid panel에서 row를 선택하면 발생 선택한 row의 정보를 얻을 수 있음
onChange	evt.args.newValue (문자열) (변경된 value) evt.args.oldValue (문자열) (변경되기 전 value)	codecombo에 선택된 값이 바뀌면 발생 변경되기 전의 값과 변경된 값의 정보를 얻을 수 있음
onShowPanel		codecombo combobox의 grid panel이 열리면 발생
onHidePanel		codecombo combobox의 grid panel이 닫히면 발생
onLoadSuccess		codecombo의 Load가 성공하면 발생
onLoadError		codecombo의 Load에 오류가 생기면 발생
onSortColumn	evt.args.column (문자열) (정렬된 field name)	codecombo combobox의 grid를 sort하면 발생

	evt.args.order (문자열) (정렬 순서: asc/desc)	
--	--	--

5. Y[wai] Component 도움말

C. Edit_mask

원하는 마스킹 형태를 지정할 수 있습니다.

type 속성을 mask로 입력하고, mask 속성에 마스킹 형식을 설정하여 사용합니다.

mask 속성을 입력하지 않을 시 기본값으로 * 형식이 적용됩니다.

```
<edit type="mask" mask="*****"></edit>
```

mask 속성의 형식은 다음과 같은 기호로 사용합니다.

기호	설명
*	알파벳 (대소문자) 숫자 입력 (A-Z, a-z, 0-9)
&	알파벳 (대문자) 숫자 입력 (A-Z, 0-9)
a	알파벳 (대소문자) (A-Z, a-z)
A	알파벳 (대문자) (A-Z)
9	숫자만 입력 (0-9)
[]	대괄호 안의 mask 형식(같은 형식)의 입력값 생략 가능 ex) mask="aa[99][**]**"
	OR (또는) ex) "a 9" → a or 9, "(aaa) (999)" → aaa or 999
{n}	n만큼 반복 ex) mask="9{5}" 숫자 입력을 5번 하는 mask 형식
{n,m}	n에서 m까지 반복 ex) mask="9{3,5}" 숫자 입력을 3~5번 하는 mask 형식

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Edit Component의 id (HTML 객체 id와 동일)
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
name	문자열		edit의 name 속성
style	문자열		edit의 style 속성
value	문자열		edit의 value 속성
class	문자열		edit의 class 속성
tabIndex	정수형		edit의 tabIndex 속성
title	문자열		edit의 title 속성
autofocus	불린형		edit의 autofocus 설정
disabled	불린형		객체 비활성화 여부 (true: disabled / false: enabled(default))
readonly	불린형		읽기 전용 / 읽기 쓰기 가능 (true: readonly / false: readable and writable(default))
maxlength	정수형		text의 최대 길이
placeholder	문자열		edit의 placeholder 속성 (힌트 표시)
type	문자열 (option)		마스킹 형식 설정 (mask, currency, date, regex, decimal, integer)
mask	문자열		마스킹 형태
defaultValue	문자열		미입력한 마스킹 부분을 설정한 값으로 채움
dataLength	정수형	읽기 전용	value 속성의 값의 길이
displayMask	불린형		마스킹 미리 보기 표시 여부

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
unmaskedValue	마스킹을 제거한 value를 가져옴		문자열 (masking을 제거한 입력 객체의 value를 반환)
getValue	화면에 표시되는 객체의 현재 value를 가져옴		문자열 (입력 객체의 값을 반환)

■ Events

이벤트명	아규먼트	설명
onkeydown		키를 누르는 순간 발생
onkeypress		키를 누른 상태에서 발생
onkeyup		키를 해제하면 발생
onblur		객체가 focus를 잃으면 발생
onfocus		객체가 focus를 얻으면 발생
onclick		객체를 클릭하면 발생
ondblclick		객체를 더블클릭하면 발생
onmousedown		마우스 버튼을 아래로 누르면 발생
onmouseover		마우스 커서가 객체 위로 이동하면 발생
onmouseup		마우스 버튼을 클릭 후 놓으면 발생
onchange	evt.args.newValue (문자열) (변경된 value) evt.args.oldValue (문자열) (변경되기 전 value)	객체의 값이 바뀌면 발생 변경되기 전의 값과 변경된 값의 정보를 얻을 수 있음
oncontextmenu		객체를 우클릭하면 발생
oncomplete		입력 포맷에 맞게 입력이 완료되면 발생
onincomplete		입력이 미완료된 후 focus를 잃으면 발생
oncleared		입력 값을 모두 지우면 발생
onZeroMskComplete		type이 mask일 때 enter key를 누르면 발생 (mask 속성이 모두 같은 형식이고 defaultValue가 이 mask에 적용될 수 있는 값일 경우 masking 입력 값의 일부를 입력하거나 아무 값도 입력하지 않고 enter key를 누르면 미입력한 masking 부분을 defaultValue로 채워줌)

5. Y[wai] Component 도움말

D. Edit_currency

금액 입력 형태를 지원합니다.

type 속성을 currency로 입력하여 사용합니다.

```
<edit type='currency'></edit>
```

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Edit Component의 id (HTML 객체 id와 동일)
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
name	문자열		edit의 name 속성
style	문자열		edit의 style 속성
value	문자열		edit의 value 속성
class	문자열		edit의 class 속성
tabIndex	정수형		edit의 tabIndex 속성
title	문자열		edit의 title 속성
autofocus	불린형		edit의 autofocus 설정
disabled	불린형		객체 비활성화 여부 (true: disabled / false: enabled(default))
readonly	불린형		읽기 전용 / 읽기 쓰기 가능 (true: readonly / false: readable and writable(default))
placeholder	문자열		edit의 placeholder 속성 (힌트 표시)
type	문자열 (option)		마스킹 형식 설정 (mask, currency, date, regex, decimal, integer)
prefix	문자열		currency 기호

digits	정수형		소수점 이하 자릿수
autoGroup	불린형		정수부 자릿수 나누기 자동 설정 여부
groupSize	정수형		정수부 자릿수를 나눌 길이
groupSeparator	문자열		정수부 자리 나눔 기호
integerDigits	정수형		정수부 자릿수 총 길이
allowMinus	불린형		마이너스 기호 표시 여부

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
unmaskedValue	마스킹을 제거한 value를 가져옴		문자열 (masking을 제거한 입력 객체의 value를 반환)
getValue	화면에 표시되는 객체의 현재 value를 가져옴		문자열 (입력 객체의 value를 반환)

■ Events

이벤트명	아규먼트	설명
onkeydown		키를 누르는 순간 발생
onkeypress		키를 누른 상태에서 발생
onkeyup		키를 해제하면 발생
onblur		객체가 focus를 잃으면 발생
onfocus		객체가 focus를 얻으면 발생
onclick		객체를 클릭하면 발생
ondblclick		객체를 더블클릭하면 발생
onmousedown		마우스 버튼을 아래로 누르면 발생
onmouseover		마우스 커서가 객체 위로 이동하면 발생
onmouseup		마우스 버튼을 클릭 후 놓으면 발생
onchange	evt.args.newValue (문자열) (변경된 value) evt.args.oldValue (문자열) (변경되기 전 value)	객체의 값이 바뀌면 발생 변경되기 전의 값과 변경된 값의 정보를 얻을 수 있음
oncontextmenu		객체를 우클릭하면 발생
oncomplete		입력 포맷에 맞게 입력이 완료되면 발생
onincomplete		입력이 미완료된 후 focus를 잃으면 발생
oncleared		입력 값을 모두 지우면 발생

5. Y[wai] Component 도움말

E. Edit_date

날짜 입력 형태를 지원합니다.

type 속성을 date로 입력하여 사용합니다.

기본값으로 yyyy/mm/dd 마스크가 적용됩니다.

```
<edit type='date'></edit>
```

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Edit Component의 id (HTML 객체 id와 동일)
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
name	문자열		edit의 name 속성
style	문자열		edit의 style 속성
value	문자열		edit의 value 속성
class	문자열		edit의 class 속성
tabIndex	정수형		edit의 tabIndex 속성
title	문자열		edit의 title 속성
autofocus	불린형		edit의 autofocus 설정
disabled	불린형		객체 비활성화 여부 (true: disabled / false: enabled(default))
readonly	불린형		읽기 전용 / 읽기 쓰기 가능 (true: readonly / false: readable and writable(default))
maxlength	정수형		text의 최대 길이
placeholder	문자열		edit의 placeholder 속성 (힌트 표시)
type	문자열		마스킹 형식 설정

	(option)		(mask, currency, date, regex, decimal, integer)
dateOrder	문자열 (option)		year/month/day 표시 순서 (yyyymmdd(default) / ddmmYYYY / mmddYYYY)
dateSeparator	문자열 (option)		날짜 구분자 ([/] (default) / [-] / [.])

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
unmaskedValue	마스킹을 제거한 value를 가져옴		문자열 (masking을 제거한 입력 객체의 value를 반환)
getValue	화면에 표시되는 객체의 현재 value를 가져옴		문자열 (입력 객체의 value를 반환)

■ Events

이벤트명	아규먼트	설명
onkeydown		키를 누르는 순간 발생
onkeypress		키를 누른 상태에서 발생
onkeyup		키를 해제하면 발생
onblur		객체가 focus를 잃으면 발생
onfocus		객체가 focus를 얻으면 발생
onclick		객체를 클릭하면 발생
ondblclick		객체를 더블클릭하면 발생
onmousedown		마우스 버튼을 아래로 누르면 발생
onmouseover		마우스 커서가 객체 위로 이동하면 발생
onmouseup		마우스 버튼을 클릭 후 놓으면 발생
onchange	evt.args.newValue (문자열) (변경된 value) evt.args.oldValue (문자열) (변경되기 전 value)	객체의 값이 바뀌면 발생 변경되기 전의 값과 변경된 값의 정보를 얻을 수 있음
oncontextmenu		객체를 우클릭하면 발생
oncomplete		입력 포맷에 맞게 입력이 완료되면 발생
onincomplete		입력이 미완료된 후 focus를 잃으면 발생
oncleared		입력 값을 모두 지우면 발생

5. Y[wai] Component 도움말

F. Edit_datetime

시간 입력 type을 지원합니다.

type 속성을 datetime으로 입력하여 사용합니다.

기본값으로 hh:mm:ss 마스크가 적용됩니다.

```
<edit type="datetime"></edit>
```

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Edit Component의 id (HTML 객체 id와 동일)
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
name	문자열		edit의 name 속성
style	문자열		edit의 style 속성
value	문자열		edit의 value 속성
class	문자열		edit의 class 속성
tabIndex	정수형		edit의 tabIndex 속성
title	문자열		edit의 title 속성
autofocus	불린형		edit의 autofocus 설정
disabled	불린형		객체 비활성화 여부 (true: disabled / false: enabled(default))
readonly	불린형		읽기 전용 / 읽기 쓰기 가능 (true: readonly / false: readable and writable(default))
maxlength	정수형		text의 최대 길이
placeholder	문자열		edit의 placeholder 속성 (힌트 표시)
type	문자열		마스킹 형식 설정

	(option)		(mask, currency, date, regex, decimal, integer)
--	----------	--	---

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
unmaskedValue	마스킹을 제거한 value를 가져옴		문자열 (masking을 제거한 입력 객체의 value를 반환)
getValue	화면에 표시되는 객체의 현재 value를 가져옴		문자열 (입력 객체의 value를 반환)

■ Events

이벤트명	아규먼트	설명
onkeydown		키를 누르는 순간 발생
onkeypress		키를 누른 상태에서 발생
onkeyup		키를 해제하면 발생
onblur		객체가 focus를 잃으면 발생
onfocus		객체가 focus를 얻으면 발생
onclick		객체를 클릭하면 발생
ondblclick		객체를 더블클릭하면 발생
onmousedown		마우스 버튼을 아래로 누르면 발생
onmouseover		마우스 커서가 객체 위로 이동하면 발생
onmouseup		마우스 버튼을 클릭 후 놓으면 발생
onchange	evt.args.newValue (문자열) (변경된 value) evt.args.oldValue (문자열) (변경되기 전 value)	객체의 값이 바뀌면 발생 변경되기 전의 값과 변경된 값의 정보를 얻을 수 있음
oncontextmenu		객체를 우클릭하면 발생
oncomplete		입력 포맷에 맞게 입력이 완료되면 발생
onincomplete		입력이 미완료된 후 focus를 잃으면 발생
oncleared		입력 값을 모두 지우면 발생

5. Y[wai] Component 도움말

G. Edit_regex

정규 표현식 입력을 지원합니다.

type 속성을 regex로 입력하고, regex 속성에 형식을 입력하여 사용합니다.

regex 속성을 입력하지 않을 시

기본값으로 (010|011|016)-?[0-9]{3,4}-?[0-9]{4} 형식이 적용됩니다.

```
<edit type='regex' regex='(010|011|016)-[0-9]{3,4}-[0-9]{4}'></edit>
```

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Edit Component의 id (HTML 객체 id와 동일)
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
name	문자열		edit의 name 속성
style	문자열		edit의 style 속성
value	문자열		edit의 value 속성
class	문자열		edit의 class 속성
tabIndex	정수형		edit의 tabIndex 속성
title	문자열		edit의 title 속성
autofocus	불린형		edit의 autofocus 설정
disabled	불린형		객체 비활성화 여부 (true: disabled / false: enabled(default))
readonly	불린형		읽기 전용 / 읽기 쓰기 가능 (true: readonly / false: readable and writable(default))
maxlength	정수형		text의 최대 길이
placeholder	문자열		edit의 placeholder 속성 (힌트 표시)

type	문자열 (option)		마스킹 형식 설정 (mask, currency, date, regex, decimal, integer)
regex	문자열		정규 표현식 형태 입력

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
unmaskedValue	마스킹을 제거한 value를 가져옴		문자열 (masking을 제거한 입력 객체의 value를 반환)
getValue	화면에 표시되는 객체의 현재 value를 가져옴		문자열 (입력 객체의 value를 반환)

■ Events

이벤트명	아규먼트	설명
onkeydown		키를 누르는 순간 발생
onkeypress		키를 누른 상태에서 발생
onkeyup		키를 해제하면 발생
onblur		객체가 focus를 잃으면 발생
onfocus		객체가 focus를 얻으면 발생
onclick		객체를 클릭하면 발생
ondblclick		객체를 더블클릭하면 발생
onmousedown		마우스 버튼을 아래로 누르면 발생
onmouseover		마우스 커서가 객체 위로 이동하면 발생
onmouseup		마우스 버튼을 클릭 후 놓으면 발생
onchange	evt.args.newValue (문자열) (변경된 value) evt.args.oldValue (문자열) (변경되기 전 value)	객체의 값이 바뀌면 발생 변경되기 전의 값과 변경된 값의 정보를 얻을 수 있음
oncontextmenu		객체를 우클릭하면 발생
oncomplete		입력 포맷에 맞게 입력이 완료되면 발생
onincomplete		입력이 미완료된 후 focus를 잃으면 발생
oncleared		입력 값을 모두 지우면 발생

5. Y[wai] Component 도움말

H. Edit_decimal

소수 입력 형태를 지원합니다.

type 속성을 decimal로 입력하여 사용합니다.

```
<edit type='decimal'></edit>
```

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Edit Component의 id (HTML 객체 id와 동일)
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
name	문자열		edit의 name 속성
style	문자열		edit의 style 속성
value	문자열		edit의 value 속성
class	문자열		edit의 class 속성
tabIndex	정수형		edit의 tabIndex 속성
title	문자열		edit의 title 속성
autofocus	불린형		edit의 autofocus 설정
disabled	불린형		객체 비활성화 여부 (true: disabled / false: enabled(default))
readonly	불린형		읽기 전용 / 읽기 쓰기 가능 (true: readonly / false: readable and writable(default))
placeholder	문자열		edit의 placeholder 속성 (힌트 표시)
type	문자열 (option)		마스킹 형식 설정 (mask, currency, date, regex, decimal, integer)
digits	정수형		소수점 이하 자릿수

autoGroup	불린형		정수부 자릿수 나누기 자동 설정 여부
groupSize	정수형		정수부 자릿수를 나눌 길이
groupSeparator	문자열		정수부 자리 나눔 기호
integerDigits	정수형		정수부 자릿수 총 길이
allowMinus	불린형		마이너스 기호 표시 여부

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
unmaskedValue	마스킹을 제거한 value를 가져옴		문자열 (masking을 제거한 입력 객체의 value를 반환)
getValue	화면에 표시되는 객체의 현재 value를 가져옴		문자열 (입력 객체의 value를 반환)

■ Events

이벤트명	아규먼트	설명
onkeydown		키를 누르는 순간 발생
onkeypress		키를 누른 상태에서 발생
onkeyup		키를 해제하면 발생
onblur		객체가 focus를 잃으면 발생
onfocus		객체가 focus를 얻으면 발생
onclick		객체를 클릭하면 발생
ondblclick		객체를 더블클릭하면 발생
onmousedown		마우스 버튼을 아래로 누르면 발생
onmouseover		마우스 커서가 객체 위로 이동하면 발생
onmouseup		마우스 버튼을 클릭 후 놓으면 발생
onchange	evt.args.newValue (문자열) (변경된 value) evt.args.oldValue (문자열) (변경되기 전 value)	객체의 값이 바뀌면 발생 변경되기 전의 값과 변경된 값의 정보를 얻을 수 있음
oncontextmenu		객체를 우클릭하면 발생
oncomplete		입력 포맷에 맞게 입력이 완료되면 발생
onincomplete		입력이 미완료된 후 focus를 잃으면 발생
oncleared		입력 값을 모두 지우면 발생

5. Y[wai] Component 도움말

I. Edit_integer

정수 입력 형태를 지원합니다.

type 속성을 integer로 입력하여 사용합니다.

```
<edit type='integer'></edit>
```

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Edit Component의 id (HTML 객체 id와 동일)
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
name	문자열		edit의 name 속성
style	문자열		edit의 style 속성
value	문자열		edit의 value 속성
class	문자열		edit의 class 속성
tabIndex	정수형		edit의 tabIndex 속성
title	문자열		edit의 title 속성
autofocus	불린형		edit의 autofocus 설정
disabled	불린형		객체 비활성화 여부 (true: disabled / false: enabled(default))
readonly	불린형		읽기 전용 / 읽기 쓰기 가능 (true: readonly / false: readable and writable(default))
maxlength	정수형		text의 최대 길이
placeholder	문자열		edit의 placeholder 속성 (힌트 표시)
type	문자열 (option)		마스킹 형식 설정 (mask, currency, date, regex, decimal, integer)

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
unmaskedValue	마스킹을 제거한 value를 가져옴		문자열 (masking을 제거한 입력 객체의 value를 반환)
getValue	화면에 표시되는 객체의 현재 value를 가져옴		문자열 (입력 객체의 value를 반환)

■ Events

이벤트명	아규먼트	설명
onkeydown		키를 누르는 순간 발생
onkeypress		키를 누른 상태에서 발생
onkeyup		키를 해제하면 발생
onblur		객체가 focus를 잃으면 발생
onfocus		객체가 focus를 얻으면 발생
onclick		객체를 클릭하면 발생
ondblclick		객체를 더블클릭하면 발생
onmousedown		마우스 버튼을 아래로 누르면 발생
onmouseover		마우스 커서가 객체 위로 이동하면 발생
onmouseup		마우스 버튼을 클릭 후 놓으면 발생
onchange	evt.args.newValue (문자열) (변경된 value) evt.args.oldValue (문자열) (변경되기 전 value)	객체의 값이 바뀌면 발생 변경되기 전의 값과 변경된 값의 정보를 얻을 수 있음
oncontextmenu		객체를 우클릭하면 발생
oncomplete		입력 포맷에 맞게 입력이 완료되면 발생
onincomplete		입력이 미완료된 후 focus를 잃으면 발생
oncleared		입력 값을 모두 지우면 발생

5. Y[wai] Component 도움말

J. DateEdit

달력 서식을 지원합니다.

입력 객체와 날짜 값을 입력할 수 있는 달력으로 구성됩니다.

입력 객체의 오른쪽 아이콘을 클릭하면 캘린더가 팝업됩니다.

한 주의 첫 번째 날은 일요일, language는 english로 기본값이 설정됩니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		DateEdit Component의 id (HTML 객체 id와 동일)
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
name	문자열		edit의 name 속성
style	문자열		edit의 style 속성
value	문자열		edit의 value 속성
class	문자열		edit의 class 속성
tabIndex	정수형		edit의 tabIndex 속성
title	문자열		edit의 title 속성
disabled	불린형		객체 비활성화 여부 (true: disabled / false: enabled(default))
readonly	불린형		읽기 전용 / 읽기 쓰기 가능 (true: readonly / false: readable and writable(default))
lang	문자열 (option)		달력 표시 언어 (English(default) / Korean / Chinese)
dateOrder	문자열 (option)		year/month/day 표시 순서 (yyyymmdd(default) / ddmmyyyy / mmddyyyy)
autoCheck	불린형		입력 포맷 설정 여부

validate	불린형		날짜 유효성 체크
placeholder	문자열		edit의 placeholder 속성 (힌트 표시)
calWidth	정수형		달력의 넓이 (단위: px)
calHeight	정수형		달력의 높이 (단위: px)
btnImg	문자열		달력의 버튼 이미지 URL
firstDay	문자열 (option)		한 주의 첫 번째 날 (Sun(default) / Mon / Tue / Wed / Thu / Fri / Sat)
dateSeparator	문자열 (option)		날짜 구분자 ([/] (default) / [-] / [.])
todayText	문자열		달력의 today 버튼 텍스트 변경
closeText	문자열		달력의 close 버튼 텍스트 변경

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
unmaskedValue	마스킹을 제거한 value를 가져옴		문자열 (masking을 제거한 입력 객체의 value를 반환)
getValue	화면에 표시되는 객체의 현재 value를 가져옴		문자열 (입력 객체의 value를 반환)
setValue	객체에 입력할 date value를 설정	value (문자열) (입력할 date value)	

■ Events

이벤트명	아규먼트	설명
onkeydown		키를 누르는 순간 발생
onkeypress		키를 누른 상태에서 발생
onkeyup		키를 해제하면 발생
onblur		객체가 focus를 잃으면 발생
onfocus		객체가 focus를 얻으면 발생
onclick		객체를 클릭하면 발생
ondblclick		객체를 더블클릭하면 발생
onmousedown		마우스 버튼을 아래로 누르면 발생
onmouseover		마우스 커서가 객체 위로 이동하면 발생
onmouseup		마우스 버튼을 클릭 후 놓으면 발생
oncontextmenu		객체를 우클릭하면 발생
onSelect		달력의 날짜를 클릭하면 발생
onChange	evt.args.newDate (문자열) (변경된 value) evt.args.oldDate (문자열) (변경되기 전 value)	객체의 값이 바뀌면 발생 변경되기 전의 값과 변경된 값의 정보를 얻을 수 있음

5. Y[wai] Component 도움말

K. tabControl

Tab type을 지원합니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		tabControl Component의 id (HTML 객체 id와 동일)
index	문자열		tabControl의 index 속성
tabTextAlign	문자열		tabControl의 모든 title text 정렬 속성
tabTitleWidth	문자열		tabControl의 모든 title 넓이 정렬 속성
tabFontFamily	문자열		tabControl의 모든 title 폰트 정렬 속성
tabFontSize	문자열		tabControl의 모든 title 폰트 크기 속성
[tabPage] id	문자열		tabPage의 id 속성
[tabPage] title	문자열		tabPage의 title 속성
[tabPage] iconSrc	문자열		tabPage의 icon 속성

■ Methods

함수명	설명	아규먼트	리턴
addTab	새로운 tabPage 추가		
removeTab	특정 tabPage 삭제	index (경수형) (삭제할 tabPage index)	
setTitle	특정 tabPage에 title name을 설정	index (경수형) (title을 변경할 tabPage index) titleNm (문자열) (변경할 title name)	

■ Events

이벤트명	아규먼트	설명
onLoaded		tabControl이 로드되면 호출됩니다.
onTabClick	evt.args.index	tabControl의 탭을 클릭하면 호출됩니다.
onTabChanging	evt.args.index	tabControl의 탭이 변경되면 호출됩니다.

5. Y[wai] Component 도움말

L. Calendar

달력을 지원합니다.

날짜를 선택하고 다음 혹은 이전 달로 이동할 수 있습니다.

한 주의 첫 번째 날은 일요일, language는 english로 기본값이 설정됩니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Calendar Component의 id (HTML 객체 id와 동일)
index	문자열		tabControl의 index 속성
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
name	문자열		calendar의 name 속성
style	문자열		calendar의 style 속성
value	문자열		calendar의 value 속성
title	문자열		calendar의 title 속성
lang	문자열 (option)		calendar의 표시 언어 (English(default) / Korean / Chinese)
dateOrder	문자열 (option)		year/month/day 표시 순서 (yyyymmdd(default) / ddmmyyyy / mmddyyyy)
firstDay	문자열 (option)		한 주의 첫 번째 날 (Sun(default) / Mon / Tue / Wed / Thu / Fri / Sat)

■ Methods

함수명	설명	아규먼트	리턴
getValue	객체에서 선택한 date value를 가져옴		문자열 (객체의 date value를 반환)
setValue	객체에 입력할 date value를 설정	value (문자열) (입력할 date value)	

■ Events

이벤트명	아규먼트	설명
onSelect		calendar의 날짜를 클릭하면 발생

5. Y[wai] Component 도움말

M. Chart

Canvas chart로 사용됩니다.

화면의 현재 값을 표시하며, option에 따라서 focus가 chart로 맞춰지면 서브 값이 팝업됩니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Chart Component의 id (HTML 객체 id와 동일)
name	문자열		chart의 name 속성
dsoName	문자열		연결할 DSO(data source object)의 객체 id
valueField	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
labelField	문자열		연결할 label field 이름 (dsoName에 연결된 DSO의 field 중 선택)
type	문자열 (option)		chart의 표시 형태 (Bar, Doughnut, Pie, Line, PolarArea, Radar)
data	문자열		chart의 현재 value
option	문자열		chart의 option
style	문자열		chart의 style
title	문자열		chart의 title 속성

■ Methods

함수명	설명	아규먼트	리턴
getData	chart의 모든 서브 값을 가져옴 (가져온 값의 형태: Variant)		가변형 (chart의 전체 data를 반환)
update	chart의 특정 data를 update (한 그룹(dataset)의 한 부분(data) update 가능)	value (정수형) (update할 value) dataset (정수형) (update할 부분(data)을 포함하고 있는 그룹의 array index number) data (정수형) (그룹(dataset)의 data에서 update할 부분의 array index number)	
addData	chart에 임의의 서브 값 추가	dataArray (가변형) (그룹(dataset)의 수와 동일한 값의 배열)	
removeData	chart에 서브 값 삭제 chart의 맨 처음 값(data)부터 삭제됨	index (정수형: option) (삭제할 값(data)의 array index number)	
clear	chart의 서브 값을 모두 삭제		
setData	새로운 chart의 값(data)을 받아 chart를 다시 그림 option은 입력하지 않아도 됨	data (가변형) (새로운 chart의 값) option (가변형: option) (새로운 chart의 option)	
setOption	chart에 새로운 option을 입력	option (가변형) (새로운 chart의 option)	

■ Events

이벤트명	아규먼트	설명
onclick	evt (가변형) (chart의 전체 정보) evt.args.data (가변형) (선택한 data의 정보)	객체를 클릭하면 발생 특정 data의 정보를 얻을 수 있음
ondblclick	evt (가변형) (chart의 전체 정보) evt.args.data (가변형) (선택한 data의 정보)	객체를 더블클릭하면 발생 특정 data의 정보를 얻을 수 있음

■ Options

Chart Component 태그 내부에 설정되는 JSON 형태의 chart option입니다.

chart option은 다른 component보다 내용이 많기 때문에 태그 속성(Attribute) 형태가 아니라 태그 내부에 설정하는 형태로 개발되었습니다.

```
<chart id="chart1" >  
  "data" : [ Data Value ],  
  "option" : { Option Value }  
</chart >
```

option 항목과 data 항목은 생략할 수 있습니다. 이러한 경우에는 기본값으로 설정됩니다.

option	유형	설명
Title Options		
Show Title (title.display)	불린형	title 표시 여부
Title Text (title.text)	문자열	title 입력 내용
Position (title.position)	문자열	title 표시 위치
Padding (title.padding)	정수형	title의 여백
Font Size (title.fontSize)	정수형	폰트 크기
Font Style (title.fontStyle)	문자열	폰트 스타일
Font Color (title.fontColor)	문자열	폰트 색상
Font Family (title.fontFamily)	문자열	폰트
Legend Options		
Show Legend (legend.display)	불린형	범례 표시 여부
Reverse (legend.reverse)	불린형	data의 범례를 반대로 표시 여부

Position (legend.position)	문자열	범례 표시 위치
Padding (legend.labels.padding)	정수형	범례의 여백
Box Width (legend.labels.boxWidth)	정수형	databox의 가로 길이
Use Point Style (legend.labels.usePointStyle)	불린형	point(원형)로 표시 여부
Font Size (legend.labels.fontSize)	정수형	폰트 크기
Font Style (legend.labels.fontStyle)	문자열	폰트 스타일
Font Color (legend.labels.fontColor)	문자열	폰트 색상
Font Family (legend.labels.fontFamily)	문자열	폰트
Tooltips Options		
Tooltips (tooltips.enabled)	불린형	마우스 오버 시 data 정보 표시 여부
Mode (tooltips.mode)	문자열	tooltip 모드 (nearest: 근접한 데이터 하나 / index: 해당 라벨 데이터)
Intersect (tooltips.intersect)	불린형	data에 교차할 때만 적용 여부
Fill Color (tooltips.backgroundColor)	문자열	tooltip의 배경 색상
Y Padding (tooltips.yPadding)	정수형	tooltip의 안쪽 y축 여백
X Padding (tooltips.xPadding)	정수형	tooltip의 안쪽 x축 여백
Corner Radius (tooltips.cornerRadius)	정수형	tooltip의 박스 라운드
Border Color (tooltips.borderColor)	문자열	tooltip의 박스 라운드 색상
Border Width (tooltips.borderWidth)	정수형	tooltip의 박스 라운드 굵기
Caret Size (tooltips.caretSize)	정수형	tooltip의 포인트 크기

Caret Padding (tooltips.caretPadding)	정수형	tooltip의 포인트 간격
Title Font Size (tooltips.titleFontSize)	정수형	tooltip의 타이틀 폰트 크기
Title Font Style (tooltips.titleFontStyle)	문자열	tooltip의 타이틀 폰트 스타일
Title Font Color (tooltips.titleFontColor)	문자열	tooltip의 타이틀 폰트 색상
Title Font Family (tooltips.titleFontFamily)	문자열	tooltip의 타이틀 폰트
Body Font Size (tooltips.bodyFontSize)	정수형	tooltip의 폰트 크기
Body Font Color (tooltips.bodyFontColor)	문자열	tooltip의 폰트 색상
Body Font Style (tooltips.bodyFontStyle)	문자열	tooltip의 폰트 스타일
Body Font Family (tooltips.bodyFontFamily)	문자열	tooltip의 폰트
Display Colors (tooltips.displayColors)	불린형	data 색상 표시 여부
Animation Options		
Duration (animation.duration)	정수형	효과 속도
Easing Effects (animation.easing)	문자열	효과 종류
Animation Scale (animation.animateScale)	불린형	PolarArea, pie, doughnut 축척에 효과 적용 여부
Scales Options		
Show xAxes Line (scales.xAxes.display)	불린형	축척선 표시 여부
Show Guide Line (scales.xAxes.gridLines.display)	불린형	X축 가이드라인 표시 여부
Guide Line Color (scales.xAxes.gridLines.color)	문자열	X축 가이드라인 색상
Border Dash (scales.xAxes.gridLines.borderDash)	배열	X축 가이드라인 점선
Line Width	정수형	X축 가이드라인 두께

(scales.xAxes.gridLines.lineWidth)		
offsetGridLines (scales.xAxes.gridLines.offsetGridLines)	불린형	true인 경우 눈금 선이 레이블 사이로 이동 bar chart에서 기본값이 true로 설정됨
Show Scale Label (cales.xAxes.ticks.display)	불린형	X축 축척선 라벨 표시 여부
Font Size (scales.xAxes.ticks.fontSize)	정수형	폰트 크기
Font Style (scales.xAxes.ticks.fontStyle)	문자열	폰트 스타일
Font Color (scales.xAxes.ticks.fontColor)	문자열	폰트 색상
Font Family (scales.xAxes.ticks.fontFamily)	문자열	폰트
Show Scale Title (scales.xAxes.scaleLabel.display)	불린형	축척선 타이틀 표시 여부
Title Text (scales.xAxes.scaleLabel.labelString)	문자열	라벨 내용
Font Size (scales.xAxes.scaleLabel.fontSize)	정수형	폰트 크기
Font Style (scales.xAxes.scaleLabel.fontStyle)	문자열	폰트 스타일
Font Color (scales.xAxes.scaleLabel.fontColor)	문자열	폰트 색상
Font Family (scales.xAxes.scaleLabel.fontFamily)	문자열	폰트
Padding (scales.xAxes.scaleLabel.padding)	정수형	라벨 여백
Line Height (scales.xAxes.scaleLabel.lineHeight)	정수형	라벨 세로 길이
Show Guide Line (scales.yAxes.gridLines.display)	불린형	Y축 가이드라인 표시 여부
Guide Line Color (scales.yAxes.gridLines.color)	문자열	Y축 가이드라인 색상
Border Dash (scales.yAxes.gridLines.borderDash)	배열	Y축 가이드라인 점선
Line Width (scales.yAxes.gridLines.lineWidth)	정수형	Y축 가이드라인 두께

Zero Line Color (scales.yAxes.gridLines.zeroLineColor)	문자열	제로라인 색상
Zero Line Width (scales.yAxes.gridLines.zeroLineWidth)	정수형	제로라인 두께
Zero Border Dash (scales.yAxes.gridLines.zeroLineBorderDash)	배열	제로라인 점선
Show Scale Label (scales.yAxes.ticks.display)	불린형	Y축 축척선 라벨 표시 여부
Min (scales.yAxes.ticks.min)	정수형	최솟값
Max (scales.yAxes.ticks.max)	정수형	최댓값
Step Size (scales.yAxes.ticks.stepSize)	정수형	축척값 간격
Font Size (scales.yAxes.ticks.fontSize)	정수형	폰트 크기
Font Style (scales.yAxes.ticks.fontStyle)	문자열	폰트 스타일
Font Color (scales.yAxes.ticks.fontColor)	문자열	폰트 색상
Font Family (scales.yAxes.ticks.fontFamily)	문자열	폰트
Show Scale Title (scales.yAxes.scaleLabel.display)	불린형	축척선 타이틀 표시 여부
Title Text (scales.yAxes.scaleLabel.labelString)	문자열	라벨 내용
Font Size (scales.yAxes.scaleLabel.fontSize)	정수형	폰트 크기
Font Style (scales.yAxes.scaleLabel.fontStyle)	문자열	폰트 스타일
Font Color (scales.yAxes.scaleLabel.fontColor)	문자열	폰트 색상
Font Family (scales.yAxes.scaleLabel.fontFamily)	문자열	폰트
Padding (scales.yAxes.scaleLabel.padding)	정수형	라벨 여백
Line Height (scales.yAxes.scaleLabel.lineHeight)	정수형	라벨 세로 길이

Scale Options		
Show Guide Line (scale.gridLines.display)	불린형	가이드라인 표시 여부
Guide Line Color (scale.gridLines.color)	문자열	가이드라인 색상
Line Width (scale.gridLines.lineWidth)	정수열	가이드라인 두께
Show Scale Label (scale.pointLabels.display)	불린형	포인트 라벨 표시 여부
Font Size (scale.pointLabels.fontSize)	정수형	폰트 크기
Font Style (scale.pointLabels.fontStyle)	문자열	폰트 스타일
Font Color (scale.pointLabels.fontColor)	문자열	폰트 색상
Font Family (scale.pointLabels.fontFamily)	문자열	폰트
Padding (scale.pointLabels.padding)	정수형	라벨 여백
Line Height (scale.pointLabels.lineHeight)	정수형	라벨 세로 길이
Show Scale Label (scale.ticks.display)	불린형	축척선 라벨 표시 여부
Min (scale.ticks.min)	정수형	최솟값
Max (scale.ticks.max)	정수형	최댓값
Step Size (scale.ticks.stepSize)	정수형	축척값 간격
Font Size (scale.ticks.fontSize)	정수형	폰트 크기
Font Style (scale.ticks.fontStyle)	문자열	폰트 스타일
Font Color (scale.ticks.fontColor)	문자열	폰트 색상
Font Family (scale.ticks.fontFamily)	문자열	폰트

PieceLabel Options		
Show Piece Label (showPieceLabel.enabled)	정수형	포인트 라벨 표시 여부
Mode (showPieceLabel.mode)	문자열	포인트 라벨 모드
Font Size (showPieceLabel.fontSize)	문자열	포인트 라벨의 폰트 크기
Font Style (showPieceLabel.fontStyle)	문자열	포인트 라벨의 폰트 스타일
Font Color (showPieceLabel.fontColor)	문자열	포인트 라벨의 폰트 색상
Font Family (showPieceLabel.fontFamily)	문자열	포인트 라벨의 폰트

5. Y[wai] Component 도움말

N. TreeView

Tree 구조의 리스트 서식을 지원합니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Treeview Component의 id (HTML 객체 id와 동일)
name	문자열		treeview의 name 속성
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
url	문자열		treeview에 data를 세팅하기 위한 JSON 파일의 URL
data	가변형		treeview에 세팅할 JSON 형태의 data
data - fixed data	문자열		data 속성을 정의할 때 존재하는 고정된 값
style	문자열		treeview의 현재 style
animate	불린형		treeview에 애니메이션 적용
lines	불린형		treeview에 Node 라인 표현
checkbox	불린형		모든 Node 앞에 checkbox를 표시할지 여부 설정 함수가 지정되면 true 값을 반환하여 checkbox 표시
cascadeCheck	불린형		cascade check 여부 설정
onlyLeafCheck	불린형		leaf node전에서만 checkbox를 표시할지 여부 설정
dnd	불린형		drag&drop을 활성화할지 여부 설정
editable	불린형		클릭한 node 편집
formatter	문자열		treeview의 formatter 함수
title	문자열		treeview의 title 속성

nodeId	정수형	읽기 전용	현재 선택된 node의 index
nodeLevel	정수형	읽기 전용	현재 선택된 node의 Level
nodeValue	문자열	읽기 전용	현재 선택된 node의 text
selected	불린형	읽기 전용	현재 선택된 node가 있는지 여부
hasChildren	불린형	읽기 전용	현재 선택된 node의 하위 node가 있는지 여부
nodeCount	정수형	읽기 전용	treeview의 전체 node 개수

■ Methods

함수명	설명	아규먼트	리턴
loadData	treeview에 data를 로딩	data (가변형) (새로운 treeview data)	
getData	treeview에서 전체 data Node 정보를 가져옴		가변형 (treeview에 현재 set된 data를 반환)
find	treeview에서 Node id로 target을 찾음	nodeId (경수형) (Node ID)	가변형 (Node ID의 Node 정보를 반환)
getNode	treeview에서 target의 Node 정보를 가져옴	target (가변형) (Node target)	가변형 (Node ID의 Node 정보를 반환)
getSelected	treeview에서 선택된 Node 정보를 가져옴 선택된 Node와 그 하위 Node의 정보를 가져올 때 사용		가변형 (treeview에서 선택된 Node의 정보를 반환)
isLeaf	treeview의 target의 최하위 리스트 여부를 가져옴	target (가변형) (Node target)	불린형 (treeview의 target의 최하위 리스트 여부 반환)
select	treeview에서 임의의 target 선택	target (가변형) (Node target)	
unselect	treeview에서 선택된 상태 해제		
collapse	treeview에서 target의 Node를 접음	target (문자열) (접고자 하는 Node target)	
expand	treeview에서 target의 Node를 펼침	target (문자열) (펼치고자 하는 Node target)	
collapseAll	treeview에서 전체 Node를 접음		
expandAll	treeview에서 전체 Node를 펼침		
expandTo	treeview에서 target까지 Node를 펼침	target (문자열) (Node target)	
toggle	treeview에서 target의 Node open / close를 번갈아 수행	target (문자열) (Node target)	
clear	treeview의 data 삭제		

clearNodeStrData	treeview에서 현재 선택된 Node 정보 중 strData를 모두 삭제		
addNode	treeview에서 target의 Node에 Child Node 추가	pld (정수형) (Child Node를 추가하고자 하는 Node ID) text (문자열) (추가할 Node에 표시될 text) imgIndex (정수형) (추가할 Node에 표시될 image index) selIndex (정수형) (추가할 Node가 선택될 때 표시되는 image index) key (문자열) (추가할 Node 정보의 key) (사용자가 Node 정보를 구분 가능)	
addChildEx	treeview에서 현재 선택된 Node에 Child Node를 추가 key를 입력하면 추후에 key로 Node 이동 가능 (.MoveNodeEx(Key)	text (문자열) (추가할 Node에 표시될 text) imgIndex (정수형) (추가할 Node에 표시될 image index) selIndex (정수형) (추가할 Node가 선택될 때 표시되는 image index) key (문자열) (추가할 Node 정보의 key) (사용자가 Node 정보를 구분 가능)	
addNodeStrData	treeview에서 현재 선택된 Node에 strData 추가	strData (가변형) (Node strData 정보)	
edit	treeview에서 현재 선택된 Node의 정보를 변경	text (문자열) (추가할 Node에 표시될 text) imgIndex (정수형) (추가할 Node에 표시될 image index) selIndex (정수형) (추가할 Node가 선택될 때 표시되는 image index)	
editNodeStrData	treeview에서 현재 선택된	strIndex	

	Node의 정보 중 strIndex의 strData를 변경	(정수형) (Node 정보에서 strData의 index) strData (가변형) (Node 정보에서 strData 정보)	
delete	treeview에서 Node 삭제	nodeId (정수형) (Node ID)	
deleteChildren	treeview에서 현재 선택된 Node의 Children을 모두 삭제		
getCurrentStrData	treeview에서 현재 선택된 Node 정보 중 strIndex의 strData를 가져옴	strIndex (정수형) (Node 정보에서 strData의 index)	문자열 (현재 Node 정보에서 해당 strIndex의 strData를 반환)
moveNode	treeview에서 Node ID로 해당 Node 선택	nodeId (정수형) (Node ID)	
moveParent	treeview에서 현재 선택된 Node의 Parent Node를 선택		
moveNodeEx	treeview에서 key value가 포함된 Node 정보를 가진 Node를 선택	key (문자열) (Node 정보에 포함된 key value)	
moveFirstChild	treeview에서 현재 선택된 Node의 Children 중 첫 번째 Child Node를 선택		
moveNextChild	treeview에서 현재 선택된 Node의 형제 Node 중 자신 다음의 Node를 선택		
moveLastChild	treeview에서 현재 선택된 Node의 Children 중 마지막 Child Node를 선택		
expandEx	treeview에서 현재 선택된 Node를 펼침 argument가 false이면 현재 선택된 Node만 펼침 argument가 true이면 하위 Node까지 모두 펼침	TF (불린형) (선택된 Node의 하위 Node를 모두 expand할지 여부)	
collapseEx	treeview에서 현재 선택된 Node를 접음 argument가 false이면 현재 선택된 Node만 접음 argument가 true이면 하위	TF (불린형) (선택된 Node의 하위 Node를 모두 collapse할지 여부)	

	Node까지 모두 접음		
getChecked	checked 상태의 Node를 가져옴 사용 가능한 상태 값: checked, unchecked, indeterminate 상태가 할당되지 않은 경우 checked 상태의 Node를 반환	code (가변형) (none: checked nodes를 가져옴) (unchecked: unchecked nodes를 가져옴) (indeterminate: indeterminate nodes를 가져옴) (checked, indeterminate: checked and indeterminate nodes를 가져옴)	가변형 (treeview에서 checked 상태의 Node의 정보를 반환)
check	지정된 Node를 checked로 설정	target (가변형) (Node target)	
uncheck	지정된 Node를 unchecked로 설정	target (가변형) (Node target)	
enableDnd	drag & drop 기능 활성화		
disableDnd	drag & drop 기능 비활성화		
beginEdit	Node 편집 시작	target (가변형) (Node target)	
endEdit	Node 편집 끝	target (가변형) (Node target)	
cancelEdit	Node 편집 취소	target (가변형) (Node target)	

■ Events

이벤트명	아규먼트	설명
onContextMenu	evt.args.e (가변형) (우클릭한 event) evt.args.node (가변형) (우클릭한 Node의 정보)	treeview를 우클릭하면 발생 우클릭한 위치의 event와 Node 정보를 얻어서 context menu를 팝업
onLoadSuccess		treeview의 Load가 성공하면 발생
onLoadError		treeview의 Load에 error가 생기면 발생
onClick	evt.args.node (가변형) (클릭한 Node의 정보)	treeview를 클릭하면 발생 클릭한 Node의 정보를 알 수 있음
onDbclick		treeview 버튼을 눌러 grid panel이 열리면 발생
onBeforeExpand	evt.args.node (가변형) (펼치고자 하는 Node의 정보)	treeview에서 target의 Node가 펼쳐지기 전에 발생
onExpand	evt.args.node (가변형) (펼친 Node의 정보)	treeview에서 target의 Node가 펼쳐지면 발생
onBeforeCollapse	evt.args.node (가변형) (접고자 하는 Node의 정보)	treeview에서 target의 Node가 접히기 전에 발생
onCollapse	evt.args.node (가변형) (접힌 Node의 정보)	treeview에서 target의 Node가 접히면 발생
onBeforeSelect	evt.args.node (가변형) (선택하고자 하는 Node의 정보)	treeview에서 target을 선택하기 전에 발생
onSelect	evt.args.node (가변형) (선택한 Node의 정보)	treeview에서 target을 선택하면 발생
onBeforeCheck	evt.args.node (가변형) (check하고자 하는 Node의 정보) evt.args.checked (불린형) (check 여부)	checkbox를 클릭하기 전에 발생 check action을 취소하면 false를 반환
onCheck	evt.args.node (가변형) (check된 Node의 정보) evt.args.checked	checkbox를 클릭하면 발생

	(불린형) (check 여부)	
onBeforeDrag	evt.args.node (가변형) (drag할 Node의 정보)	Node를 drag하기 전에 발생
onStartDrag	evt.args.node (가변형) (drag할 Node의 정보)	Node를 drag하기 시작하면 발생
onStopDrag	evt.args.node (가변형) (drag한 Node의 정보)	Node를 drag하다 멈추면 발생
onDragEnter	evt.args.target (가변형) (drop할 target node) evt.args.source (가변형) (drag할 source node)	Node를 drag하여 drop할 수 있는 target node에 입력하면 발생 false를 반환하면 drop을 거부
onDragOver	evt.args.target (가변형) (drop할 target node) evt.args.source (가변형) (drag할 source node)	Node를 tree 위에서 drag하면 발생
onDragLeave	evt.args.target (가변형) (drop할 target node) evt.args.source (가변형) (drag할 source node)	Node를 drop할 수 있는 target node에 Node를 drag해서 남기면 발생
onBeforeDrop	evt.args.target (가변형) (drop할 target node) evt.args.source (가변형) (drag할 source node) evt.args.point (문자열) (drop 실행 표시) (가능한 value: append, top, bottom)	Node를 drop하기 전에 발생
onDrop	evt.args.target (가변형) (drop할 target node) evt.args.source (가변형) (drag할 source node) evt.args.point (문자열) (drop할 위치 정보) (가능한 value: append, top, bottom)	Node를 drop하면 발생

onBeforeEdit	evt.args.node (가변형) (편집할 Node의 정보)	Node를 편집하기 전에 발생
onAfterEdit	evt.args.node (가변형) (편집한 Node의 정보)	Node를 편집한 후에 발생
onCancelEdit	evt.args.node (가변형) (편집한 Node의 정보)	Node 편집을 취소하면 발생

5. Y[wai] Component 도움말

0. cTree

Canvas 기반의 Canvas Tree입니다.
Tree 구조의 리스트 서식을 지원합니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		cTree Component의 id (HTML 객체 id와 동일)
name	문자열		ctree의 name 속성
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
url	문자열		ctree에 data를 세팅하기 위한 JSON 파일의 URL
data	가변형		ctree에 세팅할 JSON 형태의 data
data - fixed data	문자열		data 속성을 정의할 때 존재하는 고정된 값
disabled	불린형		ctree의 disabled 속성 (true: disabled / false: enabled(default))
dnd	불린형		ctree의 drag & drop 기능 (true: possible / false: impossible(default))
style	문자열		ctree의 현재 style
foldIcon	불린형	읽기 전용	Folder Type node의 앞쪽에 생성되는 작은 삼각형 아이콘 해당 아이콘을 클릭하여 node를 열거나 닫을 수 있음
title	문자열		treeview의 title 속성
nodeId	정수형	읽기 전용	현재 선택된 node의 index
nodeLevel	정수형	읽기 전용	현재 선택된 node의 Level
nodeValue	문자열	읽기 전용	현재 선택된 node의 text

selected	불린형	읽기 전용	현재 선택된 node가 있는지 여부
hasChildren	불린형	읽기 전용	현재 선택된 node의 하위 node가 있는지 여부
nodeCount	정수형	읽기 전용	ctree의 전체 node 개수

■ Methods

함수명	설명	아규먼트	리턴
loadData	ctree에 data를 로딩	data (가변형) (새로운 ctree data)	
getData	ctree에서 전체 data Node 정보를 가져옴		가변형 (ctree에 현재 set된 data를 반환)
find	ctree에서 Node id로 target을 찾음	nodeId (가변형) (Node ID)	가변형 (Node ID의 Node 정보를 반환)
getNode	ctree에서 target의 Node 정보를 가져옴	nodeId (가변형) (Node ID)	가변형 (Node ID의 Node 정보를 반환)
getSelected	ctree에서 선택된 Node 정보를 가져옴 선택된 Node와 그 하위 Node의 정보를 가져올 때 사용		가변형 (ctree에서 선택된 Node의 정보를 반환)
isLeaf	ctree에서 Node의 최하위 리스트 여부를 가져옴 Node ID를 argument로 넘기면 해당 Node의 최하위 리스트 여부를 가져옴 argument가 없을 경우 ctree에서 현재 선택된 Node의 최하위 리스트 여부를 가져옴	nodeId (가변형) (Node ID)	불린형 (ctree의 node의 최하위 리스트 여부 반환)
select	ctree에서 Node를 선택	nodeId (가변형) (Node ID)	
unselect	ctree에서 선택된 상태 해제		
collapse	ctree에서 Node를 접음 argument가 없을 경우 ctree에서 현재 선택된 Node를 접음	nodeId (가변형) (접고자 하는 Node ID)	
expand	ctree에서 Node를 펼침 argument가 없을 경우 ctree에서 현재 선택된 Node를 펼침	nodeId (가변형) (펼치고자 하는 Node ID)	
collapseAll	ctree에서 전체 Node를 접음		
expandAll	ctree에서 전체 Node를 펼침		
expandTo	ctree에서 해당 Node까지	nodeId (가변형)	

	Node를 펼치고 선택	(Node ID)	
toggle	ctree에서 node의 open/close를 번갈아 수행 argument가 없을 경우 현재 선택된 Node의 node open/close를 수행	nodeId (가변형) (Node ID)	
clear	ctree의 data 삭제		
clearNodeStrData	ctree에서 현재 선택된 Node 정보 중 strData를 모두 삭제		
addNode	ctree에서 Node에 Child Node 추가	pId (가변형) (Child Node를 추가하고자 하는 Node ID) text (문자열) (추가할 Node에 표시될 text) type (문자열) (추가할 Node에 표시될 type) strData (가변형) (추가할 Node의 StrData 정보) nodeId (가변형) (추가할 Node 정보의 ID) (사용자가 Node 정보를 구분 가능)	
addChildEx	ctree에서 현재 선택된 Node에 Child Node를 추가 ID를 입력하면 추후에 ID로 Node 선택 가능 (.MoveNodeEx(nodeId))	text (문자열) (추가할 Node에 표시될 text) type (문자열) (추가할 Node에 표시될 type) strData (가변형) (추가할 Node의 StrData 정보) nodeId (가변형) (추가할 Node 정보의 ID) (사용자가 Node 정보를 구분 가능)	
addNodeStrData	ctree에서 현재 선택된 Node에 strData 추가	strData (가변형) (Node strData 정보)	
edit	ctree에서 현재 선택된 Node의	text (문자열)	

	정보를 변경	(변경할 Node에 표시될 text) type (문자열) (변경할 Node에 표시될 type)	
editNodeStrData	ctree에서 현재 선택된 Node의 정보 중 strIndex의 strData를 변경	strIndex (경수형) (Node 정보에서 strData의 index) strData (가변형) (Node 정보에서 strData 정보)	
delete	ctree에서 Node 삭제	nodeId (가변형) (Node ID)	
deleteChildren	ctree에서 현재 선택된 Node의 Children을 모두 삭제		
getCurrentStrData	ctree에서 현재 선택된 Node 정보 중 strIndex의 strData를 가져옴	strIndex (경수형) (Node 정보에서 strData의 index)	문자열 (현재 Node 정보에서 해당 strIndex의 strData를 반환)
moveNode	ctree에서 Node ID로 해당 Node 선택	nodeId (가변형) (Node ID)	
moveParent	ctree에서 현재 선택된 Node의 Parent Node를 선택		
moveNodeEx	ctree에서 Node의 ID를 가진 Node 선택 TF argument에 true 값을 보내면 Node를 선택하고 해당 Node까지 Node가 펼쳐짐 (default: false) method 처리 성공 여부 리턴	nodeId (가변형) (Node ID) TF (불린형) (해당 Node까지 node expand 여부 (default: false)))	
moveFirstChild	ctree에서 현재 선택된 Node의 Children 중 첫 번째 Child Node를 선택		
moveNextChild	ctree에서 현재 선택된 Node의 형제 Node 중 자신 다음의 Node를 선택		
moveLastChild	ctree에서 현재 선택된 Node의 Children 중 마지막 Child Node를 선택		
expandEx	ctree에서 현재 선택된 Node를 펼침 argument가 false이면 현재	TF (불린형) (선택된 Node의 하위 Node를 모두 expand할지	

	선택된 Node만 펼침 argument가 true이면 하위 Node까지 모두 펼침	여부)	
collapseEx	ctree에서 현재 선택된 Node를 접음 argument가 false이면 현재 선택된 Node만 접음 argument가 true이면 하위 Node까지 모두 접음	TF (불린형) (선택된 Node의 하위 Node를 모두 collapse할지 여부)	

■ Events

이벤트명	아규먼트	설명
onContextMenu	evt.args.e (가변형) (우클릭한 event) evt.args.node (가변형) (우클릭한 Node의 정보)	ctree를 우클릭하면 발생 우클릭한 위치의 event와 Node 정보를 얻어서 context menu를 팝업
onLoadSuccess		ctree의 Load가 성공하면 발생
onLoadError		ctree의 Load에 오류가 생기면 발생
onClick	evt.args.node (가변형) (클릭한 Node의 정보)	ctree를 클릭하면 발생 클릭한 Node의 정보를 알 수 있음
onDbclick		ctree 버튼을 눌러 grid panel이 열리면 발생
onBeforeExpand	evt.args.node (가변형) (펼치고자 하는 Node의 정보)	ctree에서 Node가 펼쳐지기 전에 발생
onExpand	evt.args.node (가변형) (펼친 Node의 정보)	ctree에서 Node가 펼쳐지면 발생
onBeforeCollapse	evt.args.node (가변형) (접고자 하는 Node의 정보)	ctree에서 Node가 접히기 전에 발생
onCollapse	evt.args.node (가변형) (접힌 Node의 정보)	ctree에서 Node가 접히면 발생
onBeforeSelect	evt.args.node (가변형) (선택하고자 하는 Node의 정보)	ctree에서 Node를 선택하기 전에 발생
onSelect	evt.args.node (가변형) (선택한 Node의 정보)	ctree에서 Node를 선택하면 발생
onBeforeDrag	evt.args.node (가변형) (drag할 Node의 정보)	Node를 drag하기 전에 발생
onStartDrag	evt.args.node (가변형) (drag할 Node의 정보)	Node를 drag하기 시작하면 발생
onStopDrag	evt.args.node (가변형) (drag한 Node의 정보)	Node를 drag하다 멈추면 발생
onDragOver	evt.args.dragNode	Node를 tree 위에서 drag하면 발생

	(가변형) (drag할 node) evt.args.dropTarget (가변형) (drop할 target node의 정보) evt.args.dropPoint (가변형) (drop할 위치 정보)	
onBeforeDrop	evt.args.dragNode (가변형) (drag할 node) evt.args.dropTarget (가변형) (drop할 target node의 정보) evt.args.dropPoint (가변형) (drop할 위치 정보)	Node를 drop하기 전에 발생
onDrop	evt.args.dragNode (가변형) (drag할 node) evt.args.dropTarget (가변형) (drop할 target node의 정보) evt.args.dropPoint (가변형) (drop할 위치 정보)	Node를 drop하면 발생

5. Y[wai] Component 도움말

P. Bar Progress

Bar 형태의 Progress를 지원합니다.

0~100까지의 value 값을 애니메이션으로 나타냅니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Bar Progress의 id (HTML 객체 id와 동일)
name	문자열		bprogress의 name 속성
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
style	문자열		bprogress의 style 속성
value	정수형		bprogress의 value 속성 (0~100)
title	문자열		bprogress의 title 속성
gradientStart	문자열		bprogress의 value 색상의 그라데이션 시작 값 (default: #3aeabb)
gradientEnd	문자열		bprogress의 value 색상의 그라데이션 끝 값 (default: #fdd250)
valueText	불린형		value 값을 나타내는 text 표시 여부 (true: display(default) / false: do not display)
valueTextColor	문자열		ValueText의 폰트 색상 설정 (default: #000000)
emptyFill	문자열		bprogress의 빈 공간(배경) 색상 (default: #e6e6e6)
linecap	문자열 (option)		bprogress의 끝 모양 (butt / round)
duration	정수형		animation의 동작 시간 (default: 1200 (1.2 seconds))

animationStartValue	정수형 (option)		animation의 표시 시작 값 (0: left → right(default) / 1: right → left)
unit	문자열		value 속성의 표시 단위 (default: No unit)

■ Methods

함수명	설명	아규먼트	리턴
getValue	객체의 value를 가져옴		정수형 (입력 객체의 value를 반환)
setValue	객체에 입력할 value를 설정	value (정수형) (입력할 value)	

■ Events

이벤트명	아규먼트	설명
onclick		객체를 클릭하면 발생
onDrawEnd		animation이 끝나면 발생

5. Y[wai] Component 도움말

Q. Circle Progress

Circle 형태의 Progress를 지원합니다.

0~100까지의 value 값을 애니메이션으로 나타냅니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		Circle Progress의 id (HTML 객체 id와 동일)
name	문자열		cprogress의 name 속성
dsoName	문자열		연결할 DSO(data source object)의 객체 id
fldName	문자열		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
style	문자열		cprogress의 style 속성
value	정수형		cprogress의 value 속성 (0~100)
title	문자열		cprogress의 title 속성
arcCoef	정수형 (option)		객체의 원 둘레 모양 (0~10(default)) 값이 10일 경우: 원 모양 값이 10 이하인 경우: 부채꼴 모양
thickness	정수형		객체의 두께 (default: 20) 1보다 큰 값을 입력
gradientStart	문자열		cprogress의 value 색상의 그라데이션 시작 값 (default: #3aeabb)
gradientEnd	문자열		cprogress의 value 색상의 그라데이션 끝 값 (default: #fdd250)
valueText	불린형		value 값을 나타내는 text 표시 여부 (true: display(default) / false: do not display)
valueTextColor	문자열		ValueText의 폰트 색상 설정 (default: #000000)

emptyFill	문자열		cprogress의 빈 공간(배경) 색상 (default: #e6e6e6)
reverse	불린형		animation의 시작 방향 (true: Counterclockwise / false: Clockwise(default))
linecap	문자열 (option)		cprogress의 끝 모양 (butt / round)
duration	정수형		animation의 동작 시간 (default: 1200 (1.2 seconds))
animationStartValue	정수형 (option)		animation의 표시 시작 값 (0: left → right(default) / 1: right → left)
unit	문자열		value 속성의 표시 단위 (default: No unit)

■ Methods

함수명	설명	아규먼트	리턴
getValue	객체의 value를 가져옴		정수형 (입력 객체의 value를 반환)
setValue	객체에 입력할 value를 설정	value (정수형) (입력할 value)	

■ Events

이벤트명	아규먼트	설명
onclick		객체를 클릭하면 발생
onDrawEnd		animation이 끝나면 발생

5. Y[wai] Component 도움말

R. External

3rd Party Library를 지원합니다.

객체를 정의하고 3rd Party Library를 적용시킬 수 있습니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자열		External의 id (HTML 객체 id와 동일)
name	문자열		external의 name 속성
disabled	불린형		external의 비활성화 여부 (true: disable / false: enable(default))
readonly	불린형		external의 readonly (true: disable / false: enable(default))
style	문자열		external의 style 속성
title	문자열		external의 title 속성
tabindex	정수형		external의 tabindex 속성
createType	문자열	읽기 전용	external의 createType 3rd party library를 적용하여 객체를 생성하는 방법 elements: <elem>으로 생성 (default) customHtml: 사용자 html로 생성
elements	가변형		3rd party library를 적용할 <elem> 객체 (external_grid.elements로 사용 가능)
customHtml	문자열		3rd party library를 적용할 <elem> 객체 (external_grid.elements로 사용 가능)

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
clear	external에 적용된 객체 삭제		
reload	객체를 reload		

■ Events

이벤트명	아규먼트	설명
oninit		객체가 load되면 발생 (이 event에 3rd party library를 적용시키는 함수 호출)

5. Y[wai] Component 도움말

5. SubFrame

컴포넌트 내부에 다른 HTML페이지를 삽입합니다.

subFrame Manager를 이용하여 HTML페이지를 등록하고

등록된 페이지를 subFrame 컴포넌트에서 호출할 수 있습니다.

(※ waiServer와 연동해서 사용하는 컴포넌트입니다.)

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자형		SubFrame의 id (HTML 객체 id와 동일)
name	문자형		subFrame의 이름 속성
screen	문자형		subFrame 내부에 로드할 html 파일의 경로
window	가변형	읽기 전용	subFrame 내부 html 페이지의 contentWindow 객체
style	문자열		SubFrame의 style 속성

■ Methods

함수명	설명	아규먼트	리턴
-			

■ Events

이벤트명	아규먼트	설명
-		

5. Y[wai] Component 도움말

T. Search Edit

인풋박스(wai edit component)와 버튼이 결합된 컴포넌트입니다.

데이터를 입력, 출력, 수정할 수 있습니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자형		SearchEdit의 id (HTML 객체 id와 동일)
value	문자형		SearchEdit의 값
name	문자형		SearchEdit의 Name 속성
style	문자형		SearchEdit의 현재 style
class	문자형		SearchEdit의 현재 class
dsoName	문자형		연결할 DSO(data source object)의 객체 id
fldName	문자형		연결할 field 이름 (dsoName에 연결된 DSO의 field 중 선택)
metald	문자형		metald
bindFieldId	문자형		bindFieldId
EDIT component	가변형		wai edit component의 속성
anchor	문자형		부모 객체를 기준으로 위치와 크기를 자동 조절
disabled	불린형		SearchEdit의 비활성화 여부 (true: disabled / false: enabled(default))
readonly	불린형		SearchEdit의 readonly 속성 (true: disabled / false: enabled(default))
tabindex	정수형		SearchEdit의 TabIndex 속성
title	문자형		SearchEdit의 Title 속성
hideClearButton	불린형		clear 버튼의 숨김 여부 반환

hideSearchButton	불린형		search 버튼의 숨김 여부 반환
searchPosition	문자형		search 버튼의 위치 반환 (인풋박스의 왼쪽 또는 오른쪽)
clearButtonImage	불린형		clear 버튼의 이미지 반환
searchButtonImage	문자형		search 버튼의 이미지 반환
buttonFocusable	불린형		버튼에 포커스 설정값 반환

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
getValue	화면에 보이는 객체의 현재 value를 가져옴		문자형
unmaskedValue	Masking을 제거한 value를 가져옴		문자형
setValue	value를 설정	문자형	
clear	서치 인풋의 value를 빈 값으로 설정		
getEditComponent	컨트롤이 소속된 Edit 컴포넌트의 객체 반환		가변형
search	search 이벤트 발생		

■ Events

이벤트명	아규먼트	설명
onKeyDown		키를 누르는 순간 발생
onKeyPress		키를 누른 상태에서 발생
onKeyUp		키를 해제할 때 발생
onblur		객체의 포커스를 잃는 순간 발생
onfocus		객체가 포커스를 얻으면 발생
onclick		객체를 클릭하면 발생

ondblclick		객체를 더블클릭하면 발생
onmousedown		마우스의 버튼을 아래로 누르면 발생
onmouseover		마우스 포인터가 객체 위로 이동되면 발생
onmouseup		마우스 버튼을 클릭하고 놓으면 발생
onChange	evt.args.newValue (문자열) (변경된 value) evt.args.oldValue (문자열) (변경되기 전 value)	객체의 값이 바뀌고 포커스가 아웃되면 발생 변경되기 전의 값과 변경된 값의 정보를 얻을 수 있음
onContextmenu		우클릭하면 발생
oncomplete	evt.args.value (문자열) (현재 입력된 value)	입력 포맷에 맞게 입력되면 발생
onincomplete	evt.args.value (문자열) (현재 입력된 value)	입력이 미완료된 후 포커스를 잃으면 발생
onCleared		입력값을 모두 지우면 발생
onCancel		입력값을 지우는 취소버튼을 클릭하면 발생
onSearch	evt.args.value (문자열) (현재 입력된 value)	Searchinput의 enter키 또는 검색 버튼을 클릭하여 인풋의 값이 search되면 발생

5. Y[wai] Component 도움말

U. Image Button

이미지 영역과 버튼이 결합된 컴포넌트입니다.

■ Properties

속성명	데이터 유형	읽기 전용	설명
id	문자형		ImageButton의 id (HTML 객체 id와 동일)
value	문자형		ImageButton의 값
name	문자형		ImageButton의 Name 속성
style	문자형		ImageButton의 현재 style
class	문자형		ImageButton의 현재 class
metald	문자형		metald
anchor	문자형		부모 객체를 기준으로 위치와 크기를 자동 조절
disabled	불린형		ImageButton의 비활성화 여부 (true: disabled / false: enabled(default))
readonly	불린형		ImageButton의 readonly 속성 (true: disabled / false: enabled(default))
tabindex	정수형		ImageButton의 TabIndex 속성
title	문자형		ImageButton의 Title 속성
image	문자형		버튼에 표시될 이미지 url
imageAlign	문자형		이미지 위치
imageSize	문자형		이미지 크기
group	문자형		버튼의 그룹명

■ Methods

함수명	설명	아규먼트	리턴
focus	객체로 focus 이동		
click	클릭 이벤트를 발생		
setText	버튼에 표시되는 text 변경	문자형	
setImage	버튼에 표시되는 image 변경	문자형	

■ Events

이벤트명	아규먼트	설명
onblur		객체의 포커스를 잃는 순간 발생
onfocus		객체가 포커스를 얻으면 발생
onclick		객체를 클릭하면 발생
ondblclick		객체를 더블클릭하면 발생
onmousedown		마우스의 버튼을 아래로 누르면 발생
onmouseover		마우스 포인터가 객체 위로 이동되면 발생
onmouseup		마우스 버튼을 클릭하고 놓으면 발생
onKeyDown		키를 누르는 순간 발생
onKeyPress		키를 누른 상태에서 발생
onKeyUp		키를 해제할 때 발생
onContextmenu		우클릭하면 발생

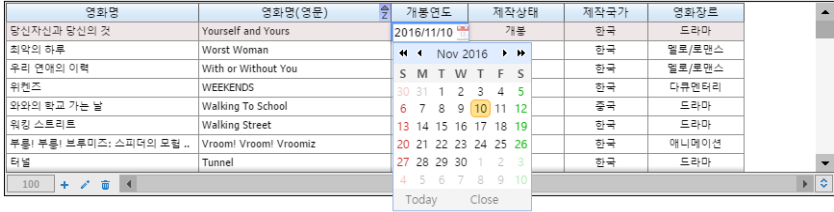
6. Y[wai] Component 예제

V. Grid

■ Basic Properties

style, option, data, editorMode, FormatString, ColumnsWidth, FieldMapping

property	source
style	style="position: relative; width: 861; height: 204; border: 1px solid rgb(234, 234, 234);"
option	<pre>"options" : { "editable":false , "columnHeight":20 , "cellHeight":20 , "fixColumn":0 , "indicator":{"show":false,"width":30} , "selector":{"show":false,"width":20,"checked":"none"} , "selection":{"rowSelect":true,"multiSelect":false} , "columns" : [{ "text":"영화명" , "alignment":"center" , "width":200 , "type":"column" , "height":20 , "editable":false , "visible":true , "sortable":true , "sortType":"string" , "field":"movieNm" , "style":"" , "editor":{"type":"text", "alignment":"left"} }, { "text":"영화명(영문)" , "alignment":"center" , "width":200 , "type":"column" , "height":20 , "editable":false , "visible":true , "sortable":true , "sortType":"string" , "field":"movieNmEn" , "style":""</pre>

	<pre> , "editor": {"type": "text", "alignment": "left"} }, { ~~~~~ } , "style" : {} , "effect": {"columnHighlight": true} , "size": {"vertScrollWidth": 19} , "footer": {"show": true, "height": 24} } </pre>
data	<pre> grid3.SetData(movieList2016); // movieList2016 is JSON format data declared as a variable. </pre>
editorMode	<pre> grid3.editorMode=true; </pre>
result	
	

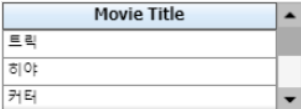
■ Style Properties

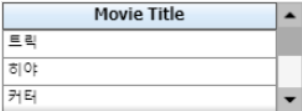
mask, text, password, codecombo, date, currency, checkbox, calculator, button, image

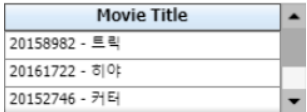
property	source																
mask	example																
	<pre>"mask": "#####-999999" "mask": "[999]9999-9999" "mask": "9999/99/99 99:99:99" "mask": "999999-999999"</pre>																
	columns																
	<pre>"columns" : [{ "field": "FIELD_VALUE", "editable": true, "width": 150, "editor": { "alignment": "center", "type": "mask", "mask": "#####-999999", "text": "주민번호" }, }, { "field": "FIELD_VALUE2", "editable": true, "width": 150, "editor": { "alignment": "center", "type": "mask", "mask": "[999]9999-9999", "text": "핸드폰" }, }, { "field": "FIELD_VALUE3", "editable": true, "width": 150, "editor": { "alignment": "center", "type": "mask", "mask": "9999/99/99 99:99:99", "text": "시간" }, }, { "field": "FIELD_VALUE4", "editable": true, "width": 150, "editor": { "alignment": "center", "type": "mask", "mask": "999999-999999", "text": "주민번호" }, }]</pre>																
result																	
<table><tr><td>주민번호</td><td>핸드폰</td><td>시간</td><td>주민번호</td></tr><tr><td>#####-1234567</td><td>[010]1234-5678</td><td>2019/03/18 20:40:38</td><td>190911-123456</td></tr></table> <table><tr><td>주민번호</td><td>핸드폰</td><td>시간</td><td>주민번호</td></tr><tr><td>#####-1234567</td><td>[010]1234-5678</td><td> 20190318 20:40:38 </td><td>190911-123456</td></tr></table> Edit Mode		주민번호	핸드폰	시간	주민번호	#####-1234567	[010]1234-5678	2019/03/18 20:40:38	190911-123456	주민번호	핸드폰	시간	주민번호	#####-1234567	[010]1234-5678	20190318 20:40:38	190911-123456
주민번호	핸드폰	시간	주민번호														
#####-1234567	[010]1234-5678	2019/03/18 20:40:38	190911-123456														
주민번호	핸드폰	시간	주민번호														
#####-1234567	[010]1234-5678	20190318 20:40:38	190911-123456														

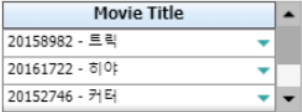
property	source
text	<pre> <grid id="gridText" style="position: relative;height: 41px; width:102px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":false,"width":30} ,"selector":{"show":false,"width":20,"checked":"none"} ,"selection":{"rowSelect":true,"multiSelect":false} ,"columns" : [{ "text":"text" ,"alignment":"center" ,"width":100 ,"type":"column" ,"height":20 ,"editable":true ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"FIELD_VALUE" ,"style":"" ,"editor":{"type":"text","alignment":"left"} }] ,"effect":{"columnHighlight":true} ,"size":{"columnWidth":75,"columnHeight":20,"cellHeight":20,"vertScrollWidth":19} ,"footer":{"show":false,"height":24} }, "data" : [{"FIELD_VALUE":"REF00001"}] </grid> </pre>
result	
text REF00001 text REF00001 Edit Mode	

property	source
password	example
	<pre>"editor":{"alignment":"center","type":"password","mask":"*"} "excelMask":"./g","excelSymbols":"*"</pre>
	columns
	<pre>"columns" : [{ "text":"Movie Title" ,"alignment":"center" ,"width":200 ,"type":"column" ,"height":20 ,"editable":"true" ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"FIELD_VALUE" ,"style":"" ,"editor":{"alignment":"left","type":"password","mask":"*"} ,"excelMask":"./g","excelSymbols":"*" }]</pre>
result	
Movie Title ***** ***** ***** Movie Title ***** ***** ***** Edit Mode	

property	source
codecombo (data load) XML file load	example
	<xml id="F945" src="/F945.xml" style="display: none;"></xml>
	columns
	<pre>"columns" : [{ "text": "Movie Title" , "alignment": "center" , "width": 200 , "type": "column" , "height": 20 , "editable": true , "visible": true , "sortable": true , "sortType": "string" , "field": "FIELD_VALUE" , "style": "" , "editor": { "type": "LookupComboBox" , "listsource": "F945" , "alignment": "left" , "keyfieldnames": "CODE" , "dropdownwidth": "200px" , "dropdownheight": "100px" , "dispField": "CODE_NM" } , "droplist": "CODE, CODE, 50, center; CODE_NM, CODE_NM, 130, left" }]</pre>
result	
	

property	source
codecombo (data load) JSON load	example
	<pre>var json_data =[{CODE:"20158982" , CODE_NM:"트릭"} ,{CODE:"20161722" , CODE_NM:"히야"} ,{CODE:"20152746" , CODE_NM:"커터"} ,{CODE:"20154910" , CODE_NM:"터널"} ,{CODE:"20144444" , CODE_NM:"아가씨"} ,{CODE:"20158482" , CODE_NM:"동주"} ,{CODE:"20130979" , CODE_NM:"덕혜옹주"}] 그리드ID.SetJsonStorage("F945",json_data);</pre>
	columns
	<pre>"columns" : [{ "text":"Movie Title" ,"alignment":"center" ,"width":200 ,"type":"column" ,"height":20 ,"editable":true ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"FIELD_VALUE" ,"style":"" ,"editor":{ "type":"LookupComboBox" ,"codeName":"F945" ,"alignment":"left" ,"keyfieldnames":"CODE" ,"dropdownwidth":"200px" ,"dropdownheight":"100px" ,"dispField":"CODE_NM" } ,"droplist":"CODE,CODE,50,center;CODE_NM,CODE_NM,130,left" }]</pre>
result	
	

property	source
codecombo (display format) mask	example
	"mask":"keyfieldnames - dispField"
	columns
	<pre> "columns" : [{ "text":"Movie Title" ,"alignment":"center" ,"width":200 ,"type":"column" ,"height":20 ,"editable":true ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"FIELD_VALUE" ,"style":"" ,"editor":{ "type":"LookupComboBox" "mask":"keyfieldnames - dispField" ,"codeName":"F945" ,"alignment":"left" ,"keyfieldnames":"CODE" ,"dropdownwidth":"200px" ,"dropdownheight":"100px" ,"dispField":"CODE_NM" ,"droplist":"CODE,CODE,50,center;CODE_NM,CODE_NM,130,left" } }] </pre>
result	
	

property	source
codecombo (display format) showIcon	example
	"showIcon":true
	columns
	<pre> "columns" : [{ "text":"Movie Title" ,"alignment":"center" ,"width":200 ,"type":"column" ,"height":20 ,"editable":true ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"FIELD_VALUE" ,"style":"" ,"editor":{ "type":"LookupComboBox" "showIcon":true, "mask":"keyfieldnames - dispField" ,"codeName":"F945" ,"alignment":"left" ,"keyfieldnames":"CODE" ,"dropdownwidth":"200px" ,"dropdownheight":"100px" ,"dispField":"CODE_NM" ,"droplist":"CODE,CODE,50,center;CODE_NM,CODE_NM,130,left" } }] </pre>
result	
	

property	source
date	<pre> <grid id="griddate" style="position: relative;height: 41px; width:142px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":false,"width":30} ,"selector":{"show":false,"width":20,"checked":"none"} ,"selection":{"rowSelect":true,"multiSelect":false} ,"columns" : [{ "text":"date" ,"alignment":"center" ,"width":140 ,"type":"column" ,"height":20 ,"editable":true ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"FIELD_VALUE" ,"style":"" ,"editor":{"type":"date","alignment":"center","dateOrder":"ddmmyyyy"} }] ,"effect":{"columnHighlight":true} ,"size":{"columnWidth":75,"columnHeight":20,"cellHeight":20,"vertScrollWidth":19} ,"footer":{"show":false,"height":24} }, "data" : [{"FIELD_VALUE": "20170307"}] </grid> </pre>
result	
date 07/03/2017 date 2017/03/07 date 2017/03/07 Mar 2017 S M T W T F S 26 27 28 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 Today Close 해당 component에서 아래의 script를 실행하면 <pre> griddate.gridObj.option.columns[0].editor.dateOrder ="yyymmdd"; var grdData = [{"FIELD_VALUE": "20170307"}]; griddate.SetData(grdData); griddate.Refresh(); </pre> yyymmdd 형태로 날짜가 표시됩니다. Edit Mode	

property	source
currency	<pre> <grid id="gridcurrency" style="position: relative;height: 41px; width:142px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":false,"width":30} ,"selector":{"show":false,"width":20,"checked":"none"} ,"selection":{"rowSelect":true,"multiSelect":false} ,"columns" : [{ "text":"currency" ,"alignment":"center" ,"width":140 ,"type":"column" ,"height":20 ,"editable":true ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"FIELD_VALUE" ,"style":"" ,"editor":{"type":"currency" ,"alignment":"right" ,"prefix" : "₩" ,"digits" : "2" } }] ,"effect":{"columnHighlight":true} ,"size":{"columnWidth":75,"columnHeight":20,"cellHeight":20,"vertScrollWidth":19} ,"footer":{"show":false,"height":24} }, "data" : [{"FIELD_VALUE":"1004"}] </grid> </pre>
	option
	prefix : 통화 기호 (default: 통화 기호 없음) digits : 소수점 이하 자릿수 (default: digits="2")

result

currency
¥ 1,004.00

해당 component에서 아래의 script를 실행하면

```
gridcurrency.gridObj.option.columns[0].editor.prefix='$';  
gridcurrency.Refresh();
```

currency
\$1,004.00

달러로 변환됩니다.

currency
\$1,004.00

Edit Mode

property	source
checkbox	<pre> <grid id="gridcheckbox" style="position: relative;height: 41px; width:142px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":false,"width":30} ,"selector":{"show":false,"width":20,"checked":"none"} ,"selection":{"rowSelect":true,"multiSelect":false} ,"columns" : [{ "text":"checkbox" ,"alignment":"center" ,"width":140 ,"type":"column" ,"height":20 ,"editable":true ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"FIELD_VALUE" ,"style":"" ,"editor":{"type":"checkbox","alignment":"center"} }] ,"effect":{"columnHighlight":true} ,"size":{"columnWidth":75,"columnHeight":20,"cellHeight":20,"vertScrollWidth":19} ,"footer":{"show":false,"height":24} }, "data" : [{"FIELD_VALUE":"true"}] </grid> </pre>
result	
checkbox ☒	

property	source
calculator	<pre> <grid id="grid1" style="height: 204px;width:700px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20} ,"conditionalstyle":[] ,"columns" : [{ "text": "Movie Title" ,"width":260 ,"type": "column" ,"height":20 ,"editable":false ,"visible":true ,"sortable":true ,"sortType": "string" ,"field": "movieNm" ,"style": "" ,"editor":{"type": "text", "alignment": "left"} } ,{ "text": "Sales Amount" ,"width":200 ,"type": "column" ,"height":20 ,"editable":true ,"visible":true ,"sortable":true ,"sortType": "string" ,"field": "salesAcc" ,"style": "" ,"editor":{"type": "CalcEdit", "displayformat": "", "0", "alignment": "right"} } ,{ "text": "Movie Audience" ,"width":190 ,"type": "column" ,"height":20 ,"editable":false ,"visible":true ,"sortable":true ,"sortType": "string" </pre>

property	source
button	<pre>//assign event to button function load(){ grid1.onButtonClick = function(grid, column, Index, RecNo){ var data = grid1.GetData(column.field); alert("ButtonClick Index : " + Index + " RecNo: " + RecNo + " Data: " + data[RecNo][column.field]); } }</pre> <pre><grid id="grid1" style="height: 204px;width:700px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20,"checked":"none"} ,"selection":{"rowSelect":true,"multiSelect":false} ,"conditionalstyle":[] ,"columns" : [{ "text":"Movie Title" ,"alignment":"center" ,"width":200 ,"type":"column" ,"height":20 ,"editable":false ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"movieNm" ,"style":"" ,"editor":{"type":"text","alignment":"left"} } ,{ "text":"Production Status" ,"alignment":"center" ,"width":100 ,"type":"column" ,"height":20 ,"editable":false ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"prdtStatNm" }] }</pre>

```

,"style":""
,"editor":{"type":"button","alignment":"center"}
},
{
"text":"Production Country"
,"alignment":"center"
,"width":87
,"type":"column"
,"height":20
,"editable":false
,"visible":true
,"sortable":true
,"sortType":"string"
,"field":"nationAlt"
,"style":""
,"editor":{"type":"text","alignment":"center"}
}
},
{
"text":"Movie genre"
,"alignment":"center"
,"width":100
,"type":"column"
,"height":20
,"editable":false
,"visible":true
,"sortable":true
,"sortType":"string"
,"field":"genreAlt"
,"style":""
,"editor":{"type":"text","alignment":"center"}
}
]
,"style":{}
,"effect":{"columnHighlight":true}
,"size":{"vertScrollWidth":19}
,"footer":{"show":true,"height":24}
}

```

result

	Movie Title	Production St..	Production ..	Movie genre
1	트릭	개봉	한국	스릴러,드라마
2	히야	개봉	한국	드라마
3	커터	개봉	한국	범죄,드라마
4	타넬	개봉	한국	드라마
5	아가씨	개봉	한국	스릴러,드라마
6	동주	개봉	한국	드라마
7	덕혜옹주	개봉	한국	사극,드라마
8	곡성	개봉	한국	미스터리,스릴러..

property	source
	example
image	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="/Y[wai]/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> </head> <body> <grid id="grid1" style="height: 204px;width:700px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20} ,"columns" : [{ "text": "Movie Title" ,"alignment": "center" ,"width":200 ,"field": "movieNm" ,"editor":{"type": "text", "alignment": "left"} } ,{ "text": "Country" ,"alignment": "center" ,"width":60 ,"field": "nationImg" ,"editor":{"type": "image", "alignment": "center", "imgmode" : "stretch"} } ,{ "text": "Production Country" ,"alignment": "center" ,"width":87 ,"field": "nationAlt" ,"editor":{"type": "text", "alignment": "center"} } ,{ "text": "Movie genre" ,"alignment": "center" ,"width":100 </pre>

	<pre> , "field": "genreAlt" , "editor": { "type": "text", "alignment": "center" } }] , "style": {} , "effect": { "columnHighlight": true } , "size": { "vertScrollWidth": 19 } , "footer": { "show": true, "height": 24 } }, "data": [{ "movieNm": "곡성", "nationImg": ".css/images/Korea.png", "nationAlt": "한국", "genreAlt": "미스터리, 스릴러, 드라마" }, { "movieNm": "밀정", "nationImg": ".css/images/Korea.png", "nationAlt": "한국", "genreAlt": "액션" }, { "movieNm": "나쁜놈은 죽는다", "nationImg": ".css/images/China.png", "nationAlt": "중국, 한국", "genreAlt": "코미디, 액션" }, { "movieNm": "당신자신과 당신의 것", "nationImg": ".css/images/Korea.png", "nationAlt": "한국", "genreAlt": "드라마" }, { "movieNm": "사랑은 부영부영", "nationImg": ".css/images/France.png", "nationAlt": "프랑스", "genreAlt": "드라마, 코미디, 멜로/로맨스" }, { "movieNm": "트릭", "nationImg": ".css/images/Korea.png", "nationAlt": "한국", "genreAlt": "스릴러, 드라마" }, { "movieNm": "서프러제트", "nationImg": ".css/images/America.png", "nationAlt": "미국", "genreAlt": "드라마" }, { "movieNm": "줄리에타", "nationImg": ".css/images/Spain.png", "nationAlt": "스페인", "genreAlt": "드라마" }, { "movieNm": "도성풍운 3", "nationImg": ".css/images/HongKong.png", "nationAlt": "홍콩", "genreAlt": "코미디, 드라마" }, { "movieNm": "아가씨", "nationImg": ".css/images/Korea.png", "nationAlt": "한국", "genreAlt": "스릴러, 드라마" }] </grid> </body> </html> </pre>
	options imgmode Normal column header의 왼쪽 상단 모서리에 배치됩니다. 이미지가 column header보다 크면 잘립니다. Stretch column header 내의 이미지는 column header의 크기에 맞춰 늘어나거나 축소됩니다. Center column header가 이미지보다 큰 경우 이미지가 중앙에 표시됩니다. 이미지가 column header보다 큰 경우 이미지는 column header의 중앙에 배치되고 바깥쪽 가장자리는 잘립니다.

Zoom

크기 비율을 유지하면서 이미지의 크기를 늘리거나 줄입니다.

result

	Movie Title	Country	Production ..	Movie genre
1	국성		한국	미스터리, 스릴러..
2	밀정		한국	액션
3	나쁜놈은 죽는다		중국, 한국	코미디, 액션
4	당신자신과 당신의 것		한국	드라마
5	사랑은 부영부영		프랑스	드라마, 코미디,...
6	트릭		한국	스릴러, 드라마
7	서프러제트		미국	드라마
8	폴리에타		스페인	드라마

10

■ Data Binding

remote data, local data, scroll performance

feature	example
remote data remote endpoint 또는 JSON으로 data를 반환하는 웹 서비스를 지정하면 remote data binding을 사용할 수 있습니다.	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="/Y[wai]/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> <script type="text/javascript" src="/.testData/searchMovieList2016.js"></script> <script type="text/javascript"> \$(document).ready(function() { grid1.SetData(movieList2016); }); </script> </head> <body> <grid id="grid1" style="height: 504px;width: 1000px; border: 1px solid rgb(234, 234, 234); position: relative;"> "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"contextmenu":true ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20,"type":"check"} ,"aggregate": [{ "field": "grid_index", "expression": "count" ,"alias" : "count" }, { "field": "audience", "expression": "SUM" ,"alias" : "sum" ,"displayformat":",0" }, { "field": "audience", "expression": "average","alias" : "average" ,"displayformat":",0.00"}, { "field": "audience", "expression": "min" ,"alias" : "min","displayformat":",0"}, { "field": "audience", "expression": "max","alias" : "max" ,"displayformat":",0"}] ,"columns" : [{ "text": "Movie code", "alignment": "center", "width": 100, "field": "movieCd", "editor": { "type": "text", "alignment": "center" }, "summaryTemplate" : "Count: #=count#" } , { "text": "Korea", "alignment": "center", "width": 100, "field": "movieNm", "edit </pre>

```

or":{"type":"text","alignment":"left"}}

,{"text":"English","alignment":"center","width":100,"field":"movieNmEn","
editor":{"type":"text","alignment":"left"}}
,{"text":"Year of
production","alignment":"center","width":80,"height":20,"field":"prdtYear
","editor":{"type":"text","alignment":"center"}}
,{"text":"Open
year","alignment":"center","width":80,"field":"openDt","editor":{"type":"da
te","alignment":"center","dtOrder":"yyymmdd","dtSeparator":"/"}}

,{"text":"Audience","alignment":"center","width":150,"field":"audience","e
ditor":{"type":"CalcEdit","displayformat":"",
,"alignment":"right"},"summaryTemplate": "Total Audience: #=sum#"}

,{"text":"Country","alignment":"center","width":200,"field":"nationAlt","ed
itor":{"type":"text","alignment":"center"},"summaryTemplate": "Average
Audience: #=average#"}
,{"text":"Movie
genre","alignment":"center","width":150,"height":20,"field":"genreAlt","e
ditor":{"type":"text","alignment":"center"},"summaryTemplate": "Min :
#=min# Max : #=max#"}
}
,"style":{}
,"effect":{"columnHighlight":true}
,"size":{"vertScrollWidth":19}
,"footer":{"show":true,"height":24}
}
</grid>

</body>
</html>

```

result

	Movie code	Korea	English	Year of pr.	Open year	Audience	Country	Movie genre
1	20158982	트릭	Trick	2016	2016/07/13		한국	스릴러,드라마
2	20161722	히야	HIYA	2015	2016/03/10		한국	드라마
3	20152746	커터	Eclipse	2016	2016/03/30		한국	범죄,드라마
4	20154910	터널	Tunnel	2016	2016/08/10		한국	드라마
5	20144444	아가씨	The Handmaiden	2016	2016/06/01		한국	스릴러,드라마
6	20158482	동주	DONGJU: The Po...	2015	2016/02/17		한국	드라마
7	20150979	역여왕	The Last Princess	2016	2016/08/03		한국	사극,드라마
8	20140194	곡성	THE WAILING	2016	2016/05/12		한국	미스터리,스릴러,드라마
9	20169730	당신자신과 당신..	Yourself and Yours	2016	2016/11/10		한국	드라마
10	20159646	죽영	The Age of Shad...	2016	2016/09/07		한국	역전
11	20153443	아수라	Asura : The City ...	2016	2016/09/28		한국	범죄,액션
12	20150901	올레	DETOUR	2016	2016/08/25		한국	드라마,코미디
13	20145502	남과 여	A Man and A Wo...	2015	2016/02/25		한국	멜로/로맨스
14	20165723	메이킹 패밀리	Making Family	2016	2016/11/28		한국,중국	멜로/로맨스,드라마
15	20140704	비밀은 없다	The Truth Benea...	2015	2016/06/23		한국	스릴러
16	20142406	탈영 종말을 사..	The Phantom De...	2016	2016/05/04		한국	드라마,액션
17	20164527	부름! 부름! 부름..	Vroom! Vroom! ..	2016	2016/12/15		한국	애니메이션
18	20161694	저자 유한준 2 - ..		2016	2016/11/10		한국	다큐멘터리
19	20162206	브룩클린	Brooklyn	2016	2016/04/21		미국	드라마,멜로/로맨스
20	20165521	극장판 포리폴스..		2015	2016/07/07		한국,일본	애니메이션,드라마,판타지
21	20155321	최악의 하루	Worst Woman	2016	2016/08/25		한국	멜로/로맨스
22	20161961	나는 용인이다	The Witness	2015	2016/09/14		한국,중국	미스터리,스릴러
23	20154803	우리 연애의 이력	With or Without ..	2015	2016/06/29		한국	멜로/로맨스

feature	example
local data grid의 datasource data 속성을 이용하여 JavaScript 객체/JSON 배열을 제공하면 local data binding을 사용할 수 있습니다.	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="/Y[wai]/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> </head> <body> <grid id="grid1" style="height: 504px;width: 1000px; border: 1px solid rgb(234, 234, 234); position: relative;"> "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"contextmenu":true ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20,"type":"check"} ,"columns" : [{"text":"Movie code", "alignment":"center", "width":100, "field":"movieCd", "editor":{"type": "text", "alignment":"center"} , "summaryTemplate" : "Count: #=count#"} ,{"text":"Korea", "alignment":"center", "width":100, "field":"movieNm", "edit or":{"type":"text", "alignment":"left"}} ,{"text":"English", "alignment":"center", "width":100, "field":"movieNmEn", " editor":{"type":"text", "alignment":"left"}} ,{"text":"Year of production", "alignment":"center", "width":80, "height":20, "field":"prdtYear ", "editor":{"type":"text", "alignment":"center"}} ,{"text":"Open year", "alignment":"center", "width":80, "field":"openDt", "editor":{"type":"da te", "alignment":"center", "dtOrder":"yyymmdd", "dtSeparator":"/"}} ,{"text":"Audience", "alignment":"center", "width":150, "field":"audience", "e ditor":{"type":"CalcEdit", "displayformat":"", , "alignment":"right"}, "summaryTemplate" : "Total Audience: #=sum#"} ,{"text":"Country", "alignment":"center", "width":200, "field":"nationAlt", "ed itor":{"type":"text", "alignment":"center"}, "summaryTemplate" : "Average Audience: #=average#"} ,{"text":"Movie </pre>

	<pre> genre","alignment":"center","width":150,"height":20,"field":"genreAlt","e ditor":{"type":"text","alignment":"center"},"summaryTemplate":"Min : #=min# Max : #=#max#" } ,"style":{} ,"effect":{"columnHighlight":true} ,"size":{"vertScrollWidth":19} ,"footer":{"show":true,"height":24} }, "data":[{ "movieCd": "20158982", "audience": "38093", "movieNm": "트릭", "movieNmEn": "Trick", "prdtYear": "2016", "openDt": "20160713", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": "/css/images/Korea.png", "nationAlt": "한국", "genreAlt": "스릴러,드라마", "repNationNm": "한국", "repGenreNm": "스릴러", "directors": [{ "peopleNm": "이창열" }], "companys": [{ "companyCd": "20160481", "companyNm": "엘씨오픽쳐스" }], . . . { "movieCd": "20166984", "audience": "0", "movieNm": "위험한 유혹", "movieNmEn": "As I Lay Dying", "prdtYear": "2013", "openDt": "20161228", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": "/css/images/America.png", "nationAlt": "미국", "genreAlt": "드라마", "repNationNm": "미국", "repGenreNm": "드라마", "directors": [{ </pre>
--	--

```

"peopleNm": "제임스 프랭코"
}},
"companys": []
} ]
</grid>

</body>
</html>

```

result

	Movie code	Korea	English	Year of pr..	Open year	Audience	Country	Movie genre
1	20158982	트릭	Trick	2016	2016/07/13	38,093	한국	스릴러,드라마
2	20161722	히아	HEYA	2015	2016/03/10	21,495	한국	드라마
3	20152746	커터	Eclipse	2016	2016/03/30	19,399	한국	범죄,드라마
4	20154910	터널	Tunnel	2016	2016/08/10	7,120,166	한국	드라마
5	20144444	아가씨	The Handmaiden	2016	2016/06/01	42,889,698	한국	스릴러,드라마
6	20158482	동주	DONGJU; The Po..	2015	2016/02/17	1,176,920	한국	드라마
7	20150979	덕혜옹주	The Last Princess	2016	2016/08/03	5,599,606	한국	사극,드라마
8	20140194	곡성	THE WAILING	2016	2016/05/12	6,877,896	한국	미스터리,스릴러,드라마
9	20169730	당신자신과 당신..	Yourself and Yours	2016	2016/11/10	180,000	한국	드라마
10	20159646	탈경	The Age of Shad..	2016	2016/09/07	7,499,735	한국	액션
11	20153443	아수라	Asura : The City ...	2016	2016/09/28	2,594,749	한국	범죄,액션
12	20150901	출처	DETOUR	2016	2016/08/25	66,419	한국	드라마,코미디
13	20145502	남과 여	A Man and A Wo...	2015	2016/02/25	203,457	한국	멜로/로맨스
14	20165723	메이킹 패밀리	Making Family	2016	2016/11/28	205	한국,중국	멜로/로맨스,드라마
15	20140704	비밀은 없다	The Truth Benea...	2015	2016/06/23	250,958	한국	스릴러
16	20142406	환영 오실들: 사..	The Phantom De...	2016	2016/05/04	1,430,960	한국	드라마,액션
17	20164527	부릉! 부릉! 부루..	Vroom! Vroom! ...	2016	2016/12/15	11,000	한국	애니메이션
18	20161694	제자 죽음을 2 - ..		2016	2016/11/10	9,533	한국	다큐멘터리
19	20162206	브루클린	Brooklyn	2016	2016/04/21	51,000	미국	드라마,멜로/로맨스
20	20165521	극장판 프리콜스..		2015	2016/07/07	9,629	한국,일본	애니메이션,드라마,판타지
21	20155321	최악의 하루	Worst Woman	2016	2016/08/25	408	한국	멜로/로맨스
22	20161961	나는 용인이다	The Witness	2015	2016/09/14	1,407	한국,중국	미스터리,스릴러
23	20154803	우리 연애의 이력	With or Without ...	2015	2016/06/29	3,043	한국	멜로/로맨스

feature	example
scroll performance	SetData(1,000 Data) 1000 Download Time 0.005 second SetData Time 0.354 second
대량의 data를 로딩합니다.	Setdata(10,000 Data) 10000 Downloaded Time 0.029 second SetData Time 0.128 second
	SetData(100,000 Data) 100000 Downloaded Time 0.215 second SetData Time 1.18 second

result

	Movie code	Korea	English	Year of pr..	Movie type	Productio..	Open year	Country	
1	20227670	그리고 집	I'm Here	2022	단편	기타		한국	
2	20224442	힐한이는 자라서	When You Grow Up	2022	단편	기타		한국	
3	20224443	원수막	The Banners	2022	단편	기타		한국	
4	20212179	해피해피 이혼파티	Happyhappy Divorceparty	2021	단편	기타		한국	
5	20212381	평양청년	Pyeongyang Neangmyeon	2019	단편	기타		한국	
6	20212385	초토화작전	Scorched Earth	2022	장편	기타		한국	
7	20227679	필수에게 자유를	Free Chol Soo Lee	2022	장편	기타		미국	
8	20212393	정순	Jeong-sun	2020	장편	기타		한국	
9	20212372	장기자랑	The Talent Show	2022	장편	가톨릭비		한국	
10	20228571	작은 정원	Little Garden	2022	장편	기타		한국	
11	20212378	유니버스	Universe Department Store	2020	장편	기타		한국	
12	20211382	오마루	Hommage	2021	장편	재봉	2022/05/26	한국	
13	20224319	오 골게온 나의 집	Home Sweet Home	2022	단편	기타		한국	
14	20217027	열집사랑	Next Door	2022	장편	재봉	2022/11/03	한국	
15	20212370	엄마, 영순	A Mother Youngsoon	2019	장편	기타		한국	
16	20225826	원지기	A Good Boy	2022	장편	기타		한국	
17	20211567	수프와 이데올로기	Soup and Ideology	2021	장편	재봉	2022/10/20	한국	
18	20227654	수라	Sura: A Love Song	2022	장편	기타		한국	
19	20200506	섬이없는지도	Map without Island	2021	장편	포반작업		한국	
20	20212353	성주여리	Sanctuary	2021	장편	기타		한국	
21	20225522	사랑합니다 그대님	I Love You, Customer	2021	단편	기타		한국	
22	20230340	에브리씽 에브리웨어 올 앳 원스+	Everything Everywhere All At Once	2022	장편	가톨릭영	2023/03/01	미국	
23	20230641	오토라는 남자	A Man Called Otto	2022	장편	가톨릭영	2023/03/29	미국	

1000

■ Export

PDF, excel, image

feature	example
export to PDF GridID.exportPdf("Download file name");	<pre> <script> function Export(){ grid1.exportPdf(); } </script> <grid id="grid1" style="height: 204px; border: 1px solid rgb(234, 234, 234); position: relative;" "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"conditionalstyle":"" ,"appendmode":false ,"autohiddencols":false ,"autofit":false ,"contextmenu":true ,"indicator":{"show":false,"width":30} ,"selector":{"show":false,"width":20,"type":"check"} ,"columns" : [{"text":"Movie code","alignment":"center","width":100,"type":"column","height":20,"edit able":false,"visible":true,"sortable":true,"sortType":"string","field":"movieC d","style":"","editor":{"type":"text","alignment":"center"}} ,{"text":"Movie Title" ,"type":"group" ,"width":200 ,"height":20 ,"visible":true ,"orientation":"horizontal" ,"columns" : [{"text":"Korea","alignment":"center","width":100,"type":"column","height" :20,"editable":"false","visible":"true","sortable":true,"sortType":"string","fiel d":"movieNm","style":"","editor":{"type":"text","alignment":"left"}} ,{"text":"English","alignment":"center","width":100,"type":"column","heigh t":20,"editable":"false","visible":"true","sortable":true,"sortType":"string","fiel d":"movieNmEn","style":"","editor":{"type":"text","alignment":"left"}}]] } </pre>

	<pre> } } ,{"text":"Year of production","alignment":"center","width":80,"type":"column","height":20, "editable":false,"visible":true,"sortable":true,"sortType":"string","field":"prdtYear","style":"","editor":{"type":"text","alignment":"center"}} ,{"text":"Open year","alignment":"center","width":80,"type":"column","height":20,"editable":false,"visible":true,"sortable":true,"sortType":"string","field":"openDt","style":"","editor":{"type":"date","alignment":"center","dtOrder":"yyyymmdd","dtSeparator":"/"}} ,{"text":"Movie type","alignment":"center","width":100,"type":"column","height":20,"editable":false,"visible":true,"sortable":true,"sortType":"string","field":"typeNm","style":"","editor":{"type":"text","alignment":"center"}} ,{"text":"Status","alignment":"center","width":200,"type":"column","height":20,"editable":false,"visible":true,"sortable":true,"sortType":"string","field":"prdtStatNm","style":"","editor":{"type":"text","alignment":"center"}} ,{"text":"Country","alignment":"center","width":200,"type":"column","height":20,"editable":false,"visible":true,"sortable":true,"sortType":"string","field":"nationAlt","style":"","editor":{"type":"text","alignment":"center"}} ,{"text":"Movie genre","alignment":"center","width":100,"type":"column","height":20,"editable":false,"visible":true,"sortable":true,"sortType":"string","field":"genreAlt","style":"","editor":{"type":"text","alignment":"center"}}] , "style":{} , "effect":{"columnHighlight":true} , "size":{"vertScrollWidth":19} , "footer":{"show":true,"height":24} }, "data":[{ "movieCd": "20158982", "audience": "38093", "movieNm": "트릭", "movieNmEn": "Trick", "prdtYear": "2016", "openDt": "20160713", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": ".css/images/Korea.png", "nationAlt": "한국", </pre>
--	--

```

"genreAlt": "스릴러,드라마",
"repNationNm": "한국",
"repGenreNm": "스릴러",
"directors": [{
"peopleNm": "이창열"
}],
"companys": [{
"companyCd": "20160481",
"companyNm": "엘씨오픽쳐스"
}],
.
.
.
{
"movieCd": "20166984",
"audience": "0",
"movieNm": "위험한 유혹",
"movieNmEn": "As I Lay Dying",
"prdtYear": "2013",
"openDt": "20161228",
"typeNm": "장편",
"prdtStatNm": "개봉",
"nationImg": ".css/images/America.png",
"nationAlt": "미국",
"genreAlt": "드라마",
"repNationNm": "미국",
"repGenreNm": "드라마",
"directors": [{
"peopleNm": "제임스 프랭코"
}],
"companys": []

```

result

Movie code	Movie Title		Year of pr..	Open year	Movie type	Status	Country
	Korea	English					
20158982	트릭	Trick	2016	2016/07/13	장편	개봉	한국
20161722	히야	HIYA	2015	2016/03/10	장편	개봉	한국
20152746	커터	Eclipse	2016	2016/03/30	장편	개봉	한국
20154910	터널	Tunnel	2016	2016/08/10	장편	개봉	한국
20144444	아가씨	The Handmaiden	2016	2016/06/01	장편	개봉	한국
20158482	동주	DONGJU; The Po..	2015	2016/02/17	장편	개봉	한국
20150979	국여공주	The Last Princess	2016	2016/08/03	장편	개봉	한국
20140194	국성	THE WALLING	2016	2016/05/12	Save Excel Save PDF Save Image	개봉	한국
20169730	당신자신과 당신..	Yourself and Yours	2016	2016/11/10		개봉	한국
20159646	달집	The Age of Shad..	2016	2016/09/07		개봉	한국
20153443	아수라	Asura : The City ..	2016	2016/09/28		개봉	한국
20150901	출례	DETOUR	2016	2016/08/25	장편	개봉	한국
20145502	남과 여	A Man and A Wo..	2015	2016/02/25	장편	개봉	한국
20165723	메이킹 패밀리	Making Family	2016	2016/11/28	장편	개봉	한국,중
20140704	비밀은 없다	The Truth Benea..	2015	2016/06/23	장편	개봉	한국
20142406	탐정 홍길동: 사..	The Phantom De..	2016	2016/05/04	장편	개봉	한국
20164527	부릉! 부릉! 부루..	Vroom! Vroom! ..	2016	2016/12/15	장편	개봉	한국

feature	example
export to excel GridID.exportExcel("Download file name");	<pre> <script> function Export(){ grid1.exportExcel(); } </script> <grid id="grid1" style="height: 204px; border: 1px solid rgb(234, 234, 234); position: relative;"> "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"conditionalstyle":"" ,"appendmode":false ,"autohiddencols":false ,"autofit":false ,"contextmenu":true ,"indicator":{"show":false,"width":30} ,"selector":{"show":false,"width":20,"type":"check"} ,"columns" : [{"text":"Movie code","alignment":"center","width":100,"type":"column","height":20,"edit able":false,"visible":true,"sortable":true,"sortType":"string","field":"movieC d","style":"","editor":{"type":"text","alignment":"center"}} ,{"text":"Movie Title" ,"type":"group" ,"width":200 ,"height":20 ,"visible":true ,"orientation":"horizontal" ,"columns" : [{"text":"Korea","alignment":"center","width":100,"type":"column","height" :20,"editable":"false","visible":"true","sortable":true,"sortType":"string","fiel d":"movieNm","style":"","editor":{"type":"text","alignment":"left"}} ,{"text":"English","alignment":"center","width":100,"type":"column","heigh t":20,"editable":"false","visible":"true","sortable":true,"sortType":"string","fi eld":"movieNmEn","style":"","editor":{"type":"text","alignment":"left"}}] } ,{"text":"Year of </pre>

	<pre> production", "alignment": "center", "width": 80, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "prdtYear", "style": "", "editor": { "type": "text", "alignment": "center" } , { "text": "Open year", "alignment": "center", "width": 80, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "openDt", "style": "", "editor": { "type": "date", "alignment": "center", "dtOrder": "yyyymmdd", "dtSeparator": "/" } , { "text": "Movie type", "alignment": "center", "width": 100, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "typeNm", "style": "", "editor": { "type": "text", "alignment": "center" } , { "text": "Status", "alignment": "center", "width": 200, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "prdtStatNm", "style": "", "editor": { "type": "text", "alignment": "center" } , { "text": "Country", "alignment": "center", "width": 200, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "nationAlt", "style": "", "editor": { "type": "text", "alignment": "center" } , { "text": "Movie genre", "alignment": "center", "width": 100, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "genreAlt", "style": "", "editor": { "type": "text", "alignment": "center" } } , "style": {} , "effect": { "columnHighlight": true } , "size": { "vertScrollWidth": 19 } , "footer": { "show": true, "height": 24 } }, "data": [{ "movieCd": "20158982", "audience": "38093", "movieNm": "트릭", "movieNmEn": "Trick", "prdtYear": "2016", "openDt": "20160713", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": "/css/images/Korea.png", "nationAlt": "한국", "genreAlt": "스릴러, 드라마", "repNationNm": "한국", "repGenreNm": "스릴러", "directors": [{ </pre>
--	---


```

"peopleNm": "이창열"
}},
"companys": [{
"companyCd": "20160481",
"companyNm": "엘씨오픽쳐스"
}],
.
.
.
{
"movieCd": "20166984",
"audience": "0",
"movieNm": "위험한 유혹",
"movieNmEn": "As I Lay Dying",
"prdtYear": "2013",
"openDt": "20161228",
"typeNm": "장편",
"prdtStatNm": "개봉",
"nationImg": "/css/images/America.png",
"nationAlt": "미국",
"genreAlt": "드라마",
"repNationNm": "미국",
"repGenreNm": "드라마",
"directors": [{
"peopleNm": "제임스 프랭코"
}],
"companys": []

```

result

Movie code	Movie Title		Year of pr..	Open year	Movie type	Status	Country
	Korea	English					
20158982	트릭	Trick	2016	2016/07/13	장편	개봉	한국
20161722	히야	HIYA	2015	2016/03/10	장편	개봉	한국
20152746	커터	Eclipse	2016	2016/03/30	장편	개봉	한국
20154910	터널	Tunnel	2016	2016/08/10	장편	개봉	한국
20144444	아가씨	The Handmaiden	2016	2016/06/01	장편	개봉	한국
20158482	동주	DONGJU; The Po...	2015	2016/02/17	장편	개봉	한국
20150979	역해술주	The Last Princess	2016	2016/07/13	장편	개봉	한국
20140194	곡성	THE WAILING	2016	2016/07/13	장편	개봉	한국
20169730	당신자신과 당신..	Yourself and Yours	2016	2016/11/28	장편	개봉	한국
20159646	탈영	The Age of Shad..	2016	2016/07/13	장편	개봉	한국
20153443	아수라	Asura : The City ..	2016	2016/07/13	장편	개봉	한국
20150901	출력	DETOUR	2016	2016/08/10	장편	개봉	한국
20145502	남과 여	A Man and A Wo...	2015	2016/02/25	장편	개봉	한국
20165723	메이킹 패밀리	Making Family	2016	2016/11/28	장편	개봉	한국, 중국
20140704	비밀은 없다	The Truth Benea..	2015	2016/06/23	장편	개봉	한국
20142406	환영 호텔들: 사...	The Phantom De...	2016	2016/05/04	장편	개봉	한국
20164527	부릉! 부릉! 브루...	Vroom! Vroom! ..	2016	2016/12/15	장편	개봉	한국

feature	example
export to image GridID.exportImage("Download file name");	<pre> <script> function Export(){ grid1.exportImage(); } </script> <grid id="grid1" style="height: 204px; border: 1px solid rgb(234, 234, 234); position: relative;"> "options" : { "editable":false , "columnHeight":20 , "cellHeight":20 , "fixColumn":0 , "conditionalstyle":"" , "appendmode":false , "autohiddencols":false , "autofit":false , "contextmenu":true , "indicator":{"show":false,"width":30} , "selector":{"show":false,"width":20,"type":"check"} , "columns" : [{"text":"Movie code","alignment":"center","width":100,"type":"column","height":20,"edit able":false,"visible":true,"sortable":true,"sortType":"string","field":"movieC d","style":"","editor":{"type":"text","alignment":"center"}} ,{"text":"Movie Title" , "type":"group" , "width":200 , "height":20 , "visible":true , "orientation":"horizontal" , "columns" : [{"text":"Korea","alignment":"center","width":100,"type":"column","height" :20,"editable":"false","visible":"true","sortable":true,"sortType":"string","fiel d":"movieNm","style":"","editor":{"type":"text","alignment":"left"}} , {"text":"English","alignment":"center","width":100,"type":"column","heigh t":20,"editable":"false","visible":"true","sortable":true,"sortType":"string","fi eld":"movieNmEn","style":"","editor":{"type":"text","alignment":"left"}}] } , {"text":"Year of </pre>

	<pre> production", "alignment": "center", "width": 80, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "prdtYear", "style": "", "editor": { "type": "text", "alignment": "center" }} , { "text": "Open year", "alignment": "center", "width": 80, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "openDt", "style": "", "editor": { "type": "date", "alignment": "center", "dtOrder": "yyyymmdd", "dtSeparator": "/" }} , { "text": "Movie type", "alignment": "center", "width": 100, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "typeNm", "style": "", "editor": { "type": "text", "alignment": "center" }} , { "text": "Status", "alignment": "center", "width": 200, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "prdtStatNm", "style": "", "editor": { "type": "text", "alignment": "center" }} , { "text": "Country", "alignment": "center", "width": 200, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "nationAlt", "style": "", "editor": { "type": "text", "alignment": "center" }} , { "text": "Movie genre", "alignment": "center", "width": 100, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "genreAlt", "style": "", "editor": { "type": "text", "alignment": "center" }}] , "style": {} , "effect": { "columnHighlight": true } , "size": { "vertScrollWidth": 19 } , "footer": { "show": true, "height": 24 } }, "data": [{ "movieCd": "20158982", "audience": "38093", "movieNm": "트릭", "movieNmEn": "Trick", "prdtYear": "2016", "openDt": "20160713", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": "/css/images/Korea.png", "nationAlt": "한국", "genreAlt": "스릴러, 드라마", "repNationNm": "한국", "repGenreNm": "스릴러", "directors": [{ </pre>
--	---

```

"peopleNm": "이창열"
}},
"companys": [{
"companyCd": "20160481",
"companyNm": "엘씨오픽쳐스"
}],
.
.
.
{
"movieCd": "20166984",
"audience": "0",
"movieNm": "위험한 유혹",
"movieNmEn": "As I Lay Dying",
"prdtYear": "2013",
"openDt": "20161228",
"typeNm": "장편",
"prdtStatNm": "개봉",
"nationImg": "/css/images/America.png",
"nationAlt": "미국",
"genreAlt": "드라마",
"repNationNm": "미국",
"repGenreNm": "드라마",
"directors": [{
"peopleNm": "제임스 프랭코"
}],
"companys": []

```

result

Movie code	Movie Title		Year of pr.	Open year	Movie type	Status	Country
	Korea	English					
20158982	트릭	Trick	2016	2016/07/13	장편	가봉	한국
20161722	히야	HIYA	2015	2016/03/10	장편	가봉	한국
20152746	커터	Eclipse	2016	2016/03/30	장편	가봉	한국
20154910	터널	Tunnel	2016	2016/08/10	장편	가봉	한국
20144444	아가씨	The Handmaiden	2016	2016/06/01	장편	가봉	한국
20158482	동주	DONGJU: The Po...	2015	2016/02/17	장편	가봉	한국
20150979	역해울주	The Last Princess	2016	Save Excel	장편	가봉	한국
20140194	극성	THE WAILING	2016		장편	가봉	한국
20169730	당신자신과 당신..	Yourself and Yours	2016	Save PDF	장편	가봉	한국
20159646	일정	The Age of Shad..	2016	7	장편	가봉	한국
20153443	아수라	Asura : The City ..	2016	Save Image	장편	가봉	한국
20150901	올레	DETOUR	2016	2016/08/25	장편	가봉	한국
20145502	남과 여	A Man and A Wo..	2015	2016/02/25	장편	가봉	한국
20165723	메이킹 패밀리	Making Family	2016	2016/11/28	장편	가봉	한국,중국
20140704	비밀은 없다	The Truth Benea..	2015	2016/06/23	장편	가봉	한국
20142406	탐정 홍길동: 사..	The Phantom De..	2016	2016/05/04	장편	가봉	한국
20164527	부릉! 부릉! 브루..	Vroom! Vroom! ..	2016	2016/12/15	장편	가봉	한국

■ Column

column grouping, column fixed, column image

feature	source
	example
column grouping data column과 sub-column group을 포함하는 column입니다. child column이 가로 또는 세로로 정렬됩니다. child column group은 parent column group과 다른 방향으로 child column을 정렬할 수 있습니다. column group은 header에 표시되거나 표시되지 않을 수 있습니다. data 행에는 표시되지 않습니다.	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="/Y[wai]/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> </head> <body> <grid id="grid1" style="height: 404px; border: 1px solid rgb(234, 234, 234); position: relative;"> "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"conditionalstyle":"" ,"appendmode":false ,"autohiddencols":false ,"autofit":false ,"contextmenu":true ,"indicator":{"show":false,"width":30} ,"selector":{"show":false,"width":20,"type":"check"} ,"columns" : [{"text":"Movie code","alignment":"center","width":100,"field":"movieCd","editor":{"type": "text","alignment":"center"}} ,{"text":"Movie Title" ,"type":"group" ,"width":200 ,"height":20 ,"visible":true ,"orientation":"horizontal" ,"alignment":"center" ,"style":"" ,"columns" : [{"text":"Korea","alignment":"center","width":100,"visible":"true","sortable" :true,"sortType":"string","field":"movieNm","editor":{"type":"text","alignme </pre>

	<pre> nt":"left"}} ,{"text":"English","alignment":"center","width":100,"visible":"true","sortable":true,"sortType":"string","field":"movieNmEn","editor":{"type":"text","alignment":"left"}}] } ,{"text":"Year of production","alignment":"center","width":80,"field":"prdtYear","editor":{"type":"text","alignment":"center"}} ,{"text":"Open year","alignment":"center","width":80,"field":"openDt","editor":{"type":"date","alignment":"center","dtOrder":"yyyymmdd","dtSeparator":"/"}} ,{"text":"Movie type","alignment":"center","width":100,"field":"typeNm","editor":{"type":"text","alignment":"center"}} ,{"text":"Production" ,"type":"group" ,"width":200 ,"height":20 ,"visible":true ,"orientation":"vertical" ,"columns": [{"text":"Status","alignment":"center","width":200,"field":"prdtStatNm","editor":{"type":"text","alignment":"center"}} ,{"text":"Country","alignment":"center","width":200,"field":"nationAlt","editor":{"type":"text","alignment":"center"}}] } ,{"text":"Movie genre","alignment":"center","width":100,"field":"genreAlt","editor":{"type":"text","alignment":"center"}}] ,"style":{} ,"effect":{"columnHighlight":true} ,"size":{"vertScrollWidth":19} ,"footer":{"show":true,"height":24} } ,"data":[{"movieCd":"20158982","audience":"38093","movieNm":"트릭","movieNmEn":"Trick","prdtYear":"2016","openDt":"20160713","typeNm":"장편","prdtStatNm":"개봉","nationImg":"","css/images/Korea.png","nationAlt":"한국","genreAlt":"스릴러,드라마","repNationNm":"한국","repGenreNm":"스릴러","directors":[{"peopleNm":"이창열"}],"companys":[{"companyCd":"20160481","companyNm":"엘씨오픽쳐스"}]} . </pre>
--	--

```

{
  "movieCd": "20166984", "audience": "0", "movieNm": "위험한
유혹", "movieNmEn": "As I Lay
Dying", "prdtYear": "2013", "openDt": "20161228", "typeNm": "장편", "prdtSta
tNm": "개봉", "nationImg": "/css/images/America.png", "nationAlt": "미국", "g
enreAlt": "드라마", "repNationNm": "미국", "repGenreNm": "드라마", "directors"
:[{"peopleNm": "제임스 프랭코"}], "companys": []}
</grid>

</body>
</html>

```

options

orientation

vertical

column을 세로 방향으로 그룹화합니다.

horizontal

column을 가로 방향으로 그룹화합니다.

result

Movie code	Movie Title		Year of pr..	Open year	Movie type	Production		Movie genre
	Korea	English				Status	Country	
20158982	트릭	Trick	2016	2016/07/13	장편	개봉	한국	스릴러, 드라마
20161722	히어	HIYA	2015	2016/03/10	장편	개봉	한국	드라마
20152746	커터	Eclipse	2016	2016/03/30	장편	개봉	한국	범죄, 드라마
20154910	터널	Tunnel	2016	2016/08/10	장편	개봉	한국	드라마
20144444	아가씨	The Handmaiden	2016	2016/06/01	장편	개봉	한국	스릴러, 드라마
20158482	동주	DONGJU: The Po..	2015	2016/02/17	장편	개봉	한국	드라마
20150979	역적:홍주	The Last Princess	2016	2016/08/03	장편	개봉	한국	사극, 드라마
20140194	곡성	THE WAILING	2016	2016/05/12	장편	개봉	한국	미스터리, 스릴러..

feature	source
column fixed	example
	<pre> <grid id="grid1" style="height: 204px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false , "columnHeight":20 , "cellHeight":20 , "fixColumn":2 , "indicator":{"show":false,"width":30} , "selector":{"show":false,"width":20} , "columns" : [{ "text":"Movie code" , "alignment":"center" , "width":100 , "type":"column" , "height":20 , "editable":false , "visible":true , "sortable":true , "sortType":"string" , "field":"movieCd" , "style":"" , "tooltip":false , "editor":{"type":"text","alignment":"center"} }], { "text":"Movie Title" , "alignment":"center" , "width":200 , "type":"column" , "height":20 , "editable":false , "visible":true , "sortable":true , "sortType":"string" , "field":"movieNm" , "style":"" , "tooltip":false , "editor":{"type":"text","alignment":"left"} }], { "text":"Movie Title(English)" , "alignment":"center" , "width":200 </pre>


```

, "type": "column"
, "height": 20
, "editable": false
, "visible": true
, "sortable": true
, "sortType": "string"
, "field": "movieNmEn"
, "style": ""
, "tooltip": false
, "editor": { "type": "text", "alignment": "left" }
}
, { "text": "Year of production"
, "alignment": "center"
, "width": 80
, "type": "column"
, "height": 20
, "editable": false
, "visible": true
, "sortable": true
, "sortType": "string"
, "field": "prdtYear"
, "style": ""
, "tooltip": false
, "editor": { "type": "text", "alignment": "center" }
}
, {
"text": "Open year"
, "alignment": "center"
, "width": 80
, "type": "column"
, "height": 20
, "editable": false
, "visible": true
, "sortable": true
, "sortType": "string"
, "field": "openDt"
, "style": ""
, "tooltip": false
, "editor": { "type": "date", "alignment": "center", "dtOrder": "yyyymmdd", "dtSe
parator": "/" }
}
, {
"text": "Movie type"
, "alignment": "center"
, "width": 100
, "type": "column"
, "height": 20
, "editable": false

```

	<pre> , "visible": true , "sortable": true , "sortType": "string" , "field": "typeNm" , "style": "" , "tooltip": false , "editor": { "type": "text", "alignment": "center" } } , { "text": "Production Status" , "alignment": "center" , "width": 100 , "type": "column" , "height": 20 , "editable": false , "visible": true , "sortable": true , "sortType": "string" , "field": "prdtStatNm" , "style": "" , "tooltip": false , "editor": { "type": "text", "alignment": "center" } } , { "text": "Production Country" , "alignment": "center" , "width": 87 , "type": "column" , "height": 20 , "editable": false , "visible": true , "sortable": true , "sortType": "string" , "field": "nationAlt" , "style": "" , "tooltip": false , "editor": { "type": "text", "alignment": "center" } } , { "text": "Movie genre" , "alignment": "center" , "width": 100 , "type": "column" , "height": 20 , "editable": false , "visible": true , "sortable": true , "sortType": "string" , "field": "genreAlt" </pre>
--	---

```

, "style": ""
, "tooltip": false
, "editor": { "type": "text", "alignment": "center" }
}
]
, "style": {}
, "effect": { "columnHighlight": true }
, "size": { "vertScrollWidth": 19 }
, "footer": { "show": true, "height": 24 }
}

```

option

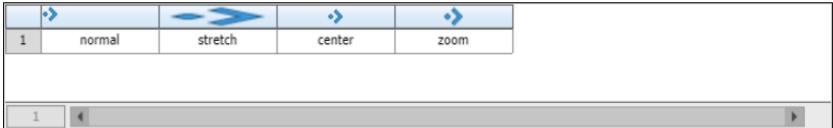
"fixColumn": 고정할 열의 개수

result

Movie code	Movie Title	ie Title(English)	Year of pr..	Open year	Movie type
20158982	트릭		2016	2016/07/13	장편
20161722	히아		2015	2016/03/10	장편
20152746	커터		2016	2016/03/30	장편
20154910	터널		2016	2016/08/10	장편
20144444	아가씨	aiden	2016	2016/06/01	장편
20158482	홍주	he Portrait of A Poet	2015	2016/02/17	장편
20150979	덕혜옹주	ncess	2016	2016/08/03	장편
20140194	극성	IG	2016	2016/05/12	장편

100

feature	source
column image	example
	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="/Y[wai]/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> </head> <body> <grid id="grid1" style="height: 504px; border: 1px solid rgb(234, 234, 234); position: relative;"> "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"contextmenu":true ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20,"type":"check"} ,"columns" : [{"text":"","img": "/Y[wai]/res/indiCursor.png","alignment":"center" ,"alignment":"center","type":"column","width":100,"field":"normal","editor ":"type":"text","alignment":"center"}} {"text":"","img": "/Y[wai]/res/indiCursor.png","imgmode": "stretch" ,"alignment":"center","type":"column","width":100,"field":"stretch","editor ":"type":"text","alignment":"left"}} {"text":"","img": "/Y[wai]/res/indiCursor.png","imgmode": "center" ,"alignment":"center","type":"column","width":100,"field":"center","editor ":"type":"text","alignment":"left"}} {"text":"","img": "/Y[wai]/res/indiCursor.png","imgmode": "zoom" ,"alignment":"center","type":"column","width":100,"field":"zoom","editor ":"type":"text","alignment":"center"}}] ,"style":{} ,"effect":{"columnHighlight":true} ,"size":{"vertScrollWidth":19} ,"footer":{"show":true,"height":24} }, "data":[{"normal":"normal","stretch":"stretch","center":"center","zoom":"zoom"}] </pre>

	</grid> </body> </html>
	options
	imgmode Normal column header의 왼쪽 상단 모서리에 배치됩니다. 이미지가 column header보다 크면 잘립니다. Stretch column header 내의 이미지는 column header의 크기에 맞춰 늘어나거나 축소됩니다. Center column header가 이미지보다 큰 경우 이미지가 중앙에 표시됩니다. 이미지가 column header보다 큰 경우 이미지는 column header의 중앙에 배치되고 바깥쪽 가장자리는 잘립니다. Zoom 크기 비율을 유지하면서 이미지의 크기를 늘리거나 줄입니다.
result	
	

■ Paging

lazy loading, paging

feature	source
lazy loading grid에서 lazy loading은 마지막 row를 스크롤할 때 다음 data를 읽는 방식으로 구현할 수 있습니다.	example
	<pre> <!DOCTYPE html> <html> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <title>Grid</title> <script type="text/javascript" src="https://uitool.org/Y[wai]/wai.js"></script> <script type="text/javascript" src="https://uitool.org/Y[wai]/IDE/support/testData/searchMovieList2016.js"></script> <script type="text/javascript" src="https://uitool.org/Y[wai]/IDE/support/testData/searchMovieList2017.js"></script> <style> body { font-family: sans-serif; font-size: 13px; } </style> <script type="text/javascript"> /***** ***** * 초기 세팅 (TEST영역) ***** *****/ var fnInit = function(){ grid1.SetData(movieList2016); }; function onScrollToBottom(){ console.log("onScrollToBottom"); grid1.AppendData(movieList2017); } function onScrollChanged(event){ var idx = event.args["Index"]; console.log("onScrollChanged" , idx); } </pre>

```

/*****
*****
*   Grid Data Setting 영역
*****
*****/

</script>
</head>
<body>

<h2>Lazy Loading</h2>
<div style="margin: 20px;">
  <p>* Lazy Loading in the grid can be implemented by reading the next
  data when the last row is scrolled.
</div>

<div style="margin: 20px; width:700px;border: 1px solid; background:
rgb(250, 250, 250);">
  <grid id="grid1" style="height: 204px; border: 1px solid rgb(234, 234,
234);" onscrolltobottom="onScrollToBottom(event)"
onscrollchanged="onScrollChanged(event)">
    "options" : {
      "editable":false
      ,"columnHeight":20
      ,"cellHeight":20
      ,"fixColumn":0
      ,"indicator":{"show":false,"width":30}
      ,"selector":{"show":false,"width":20}
      ,"columns" : [
        {
          "text":"Movie code"
          ,"alignment":"center"
          ,"width":100
          ,"field":"movieCd"
          ,"style":""
          ,"editor":{"type":"text","alignment":"center"}
        }
        ,{
          "text":"Movie Title"
          ,"alignment":"center"
          ,"width":200
          ,"field":"movieNm"
          ,"style":""
          ,"editor":{"type":"text","alignment":"left"}
        }
      ]
    }
  </grid>
</div>

```

	<pre> ,{ "text": "Movie Title(English)" , "alignment": "center" , "width": 200 , "field": "movieNmEn" , "style": "" , "editor": { "type": "text", "alignment": "left" } } ,{ "text": "Year of production" , "alignment": "center" , "width": 80 , "field": "prdtYear" , "style": "" , "editor": { "type": "text", "alignment": "center" } } ,{ "text": "Open year" , "alignment": "center" , "width": 80 , "field": "openDt" , "editor": { "type": "date", "alignment": "center", "dtOrder": "yyyymmdd", "dtSeparator": "/" } } ,{ "text": "Movie type" , "alignment": "center" , "width": 100 , "field": "typeNm" , "editor": { "type": "text", "alignment": "center" } } ,{ "text": "Production Status" , "alignment": "center" , "width": 100 , "field": "prdtStatNm" , "editor": { "type": "text", "alignment": "center" } } ,{ "text": "Production Country" , "alignment": "center" , "width": 87 , "field": "nationAlt" , "editor": { "type": "text", "alignment": "center" } } ,{ "text": "Movie genre" , "alignment": "center" , "width": 100 </pre>
--	---


```

        ,"type": "column"
        ,"field": "genreAlt"
        ,"editor": {"type": "text", "alignment": "center"}
    }
    ]
    ,"style": {}
    ,"effect": {"columnHighlight": true}
    ,"size": {"vertScrollWidth": 19}
    ,"footer": {"show": true, "height": 24}
    }
</grid>
</div>

</body>
</html>

```

function

```

$(document).ready(function() {
    //Called when scrolling to the last row of the grid
    grid1.onScrollToBottom = function(){
        console.log("onScrollToBottom");
        grid1.AppendData(movieList2017);
    }
    //Called when scrolling
    grid1.onScrollChanged = function( idx){
        console.log("onScrollChanged" , idx);
    }
});

```

result

Movie code	Movie Title	Movie Title(English)	Year of pr..	Open year	
20158982	트릭	Trick	2016	2016/07/13	
20161722	히야	HIYA	2015	2016/03/10	
20152746	커터	Eclipse	2016	2016/03/30	
20154910	터널	Tunnel	2016	2016/08/10	
20144444	아가씨	The Handmaiden	2016	2016/06/01	
20158482	동주	DONGJU; The Portrait of A Poet	2015	2016/02/17	
20150979	덕혜옹주	The Last Princess	2016	2016/08/03	
20140194	곡성	THE WAILING	2016	2016/05/12	

1100

feature	examples
paging grid는 자체 paging 처리를 구현할 수 있습니다.	<pre data-bbox="591 380 1273 1864"> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <link rel="stylesheet" type="text/css" href="http://uitool.org/wai/waiAll.css"> <script type="text/javascript" src="/Y[wai]/wai.js"> </script> <link href="/css/default.css" rel="stylesheet" type="text/css"> <style> ul { list-style:none; padding-left:0px; text-align: center; display: inline-block; border: 1px solid #ccc; border-right: 0; } ul li { text-align: center; float: left; } ul li a { display: block; font-size: 14px; padding: 9px 12px; border-right: solid 1px #ccc; box-sizing: border-box; } ul li.on { background: #eda712; } ul li.on a { color: #fff; } </style> <script> function setPaging() { grid1.setPaging(true,10); </pre>

	<pre> //페이징 표시 호출 paging(grid1, 1); } function onPageChanging(event){ var grid = event.args["grid"]; var curPage = event.args["curPage"]; console.log("old" ,curPage); } function onPageChanged(event){ var grid = event.args["grid"]; var curPage = event.args["curPage"]; console.log("new" ,curPage); } </script> </head> <body> <grid id="grid1" style="width:1000px; height: 300px; border: 1px solid rgb(234, 234, 234);" onpagechanging="onPageChanging(event)" onpagechanged="onPageChanged(event)"> "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":false,"width":30} ,"selector":{"show":false,"width":20} ,"columns" : [{ "text":"Movie code" ,"alignment":"center" ,"width":100 ,"field":"movieCd" ,"tooltip":false ,"editor":{"type":"text" ,"alignment":"center"} } ,{ "text":"Movie Title" ,"alignment":"center" ,"width":200 ,"field":"movieNm" ,"tooltip":false ,"editor":{"type":"text" ,"alignment":"left"} } ,{ "text":"Movie Title(English)" ,"alignment":"center" </pre>
--	---

```

,"width":200
,"field":"movieNmEn"
,"tooltip":false
,"editor":{"type":"text","alignment":"left"}
}
,{
"text":"Year of production"
,"alignment":"center"
,"width":80
,"field":"prdtYear"
,"style":""
,"tooltip":false
,"editor":{"type":"text","alignment":"center"}
}
,{
"text":"Open year"
,"alignment":"center"
,"width":80
,"field":"openDt"
,"style":""
,"tooltip":false
,"editor":{"type":"date","alignment":"center","dtOrder":"yyymmdd","dtSe
parator":"/"}
}
,{
"text":"Movie type"
,"alignment":"center"
,"width":100
,"field":"typeNm"
,"style":""
,"tooltip":false
,"editor":{"type":"text","alignment":"center"}
}
,{
"text":"Production Status"
,"alignment":"center"
,"width":100
,"field":"prdtStatNm"
,"style":""
,"tooltip":false
,"editor":{"type":"text","alignment":"center"}
}
,{
"text":"Production Country"
,"alignment":"center"
,"width":87
,"field":"nationAlt"
,"style":""
,"tooltip":false

```

	<pre> , "editor": { "type": "text", "alignment": "center" } } ,{ "text": "Movie genre" , "alignment": "center" , "width": 100 , "field": "genreAlt" , "style": "" , "tooltip": false , "editor": { "type": "text", "alignment": "center" } }] , "style": {} , "effect": { "columnHighlight": true } , "size": { "vertScrollWidth": 19 } , "footer": { "show": true, "height": 24 } }, "data": [{ "movieCd": "20158982", "movieNm": "트릭", "movieNmEn": "Trick", "prdtYear": "2016", "openDt": "20160713", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": "/css/images/Korea.png", "nationAlt": "한국", "genreAlt": "스릴러, 드라마", "repNationNm": "한국", "repGenreNm": "스릴러", "directors": [{"peopleNm": "이창열"}], "companys": [{"companyCd": "20160481", "companyNm": "엘씨오픽쳐스"}], . . . { "movieCd": "20166984", "movieNm": "위험한 유혹", "movieNmEn": "As I Lay Dying", "prdtYear": "2013", "openDt": "20161228", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": "/css/images/America.png", "nationAlt": "미국", "genreAlt": "드라마", "repNationNm": "미국", "repGenreNm": "드라마", "directors": [{"peopleNm": "제임스 프랭코"}], "companys": [] } </grid> <button onClick='setPaging()'>setPaging</button> <ul id="pagingul"> <script> function paging(grid, currentPage) { let pageCount = 5; //페이지에 나타낼 페이지 수 let totalPage = grid.getPageCount(); if(totalPage<pageCount){ pageCount=totalPage; </pre>
--	--

```

}

let pageGroup = Math.ceil(currentPage / pageCount); // 페이지 그룹
let last = pageGroup * pageCount; //화면에 보여질 마지막 페이지 번호

if (last > totalPage) {
  last = totalPage;
}

let first = last - (pageCount - 1); //화면에 보여질 첫번째 페이지 번호
let next = last + 1;
let prev = first - 1;
let pageHtml = "";
if (prev > 0) {
  pageHtml += "<li><a href='#' id='prev'> prev </a></li>";
}

//페이징 번호 표시
for (var i = first; i <= last; i++) {
  if (currentPage == i) {
    pageHtml +=
      "<li class='on'><a href='#' id='" + i + "'>" + i + "</a></li>";
  } else {
    pageHtml += "<li><a href='#' id='" + i + "'>" + i + "</a></li>";
  }
}
if (last < totalPage) {
  pageHtml += "<li><a href='#' id='next'> next </a></li>";
}

$("#pagingul").html(pageHtml);

//페이징 번호 클릭 이벤트
$("#pagingul li a").click(function () {

  let $id = $(this).attr("id");
  selectedPage = $(this).text();

  if ($id == "next") selectedPage = next;
  if ($id == "prev") selectedPage = prev;

  //페이징 표시 재호출
  paging(grid, selectedPage);
  console.log("selectedPage", selectedPage);
  grid.setPage(selectedPage);
});
}

</script>

```

</body>
</html>

result

Movie code	Movie Title	Movie Title(English)	Year of pr..	Open year	Movie type	Production St..	Production ..	Movie
20158982	트릭	Trick	2016	2016/07/13	장편	가봉	한국	스릴리,
20161722	히야	HIYA	2015	2016/03/10	장편	가봉	한국	드라
20152746	커터	Eclipse	2016	2016/03/30	장편	가봉	한국	범죄, 스
20154910	터널	Tunnel	2016	2016/08/10	장편	가봉	한국	드라
20144444	아가씨	The Handmaiden	2016	2016/06/01	장편	가봉	한국	스릴리,
20158482	동주	DONGJU: The Portrait of A Poet	2015	2016/02/17	장편	가봉	한국	드라
20150979	역할종주	The Last Princess	2016	2016/08/03	장편	가봉	한국	사극, 스
20140194	곡성	THE WAILING	2016	2016/05/12	장편	가봉	한국	미스터리
20169730	당신자신과 당신의 것	Yourself and Yours	2016	2016/11/10	장편	가봉	한국	드라
20159646	월명	The Age of Shadows	2016	2016/09/07	장편	가봉	한국	역

100

12345next

■ Functionality

aggregates, grouping, filtering, popup editing

feature	source
	example
aggregates grid에 숫자를 표시할 때 grid group 또는 column의 하단에서 집계 계산 결과를 볼 수 있습니다. 가장 많이 사용되는 집계 기능을 지원합니다.	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="/wai/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> </head> <body> <grid id="grid1" style="height: 504px;width: 1000px; border: 1px solid rgb(234, 234, 234); position: relative;"> "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"conditionalstyle":"" ,"appendmode":false ,"autohiddencols":false ,"autofit":false ,"contextmenu":true ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20,"checked":"none","type":"check"} ,"group":[{"field": "typeNm", "title" : "Movie type" } ,{"field": "prdtYear", "title" : "Year of production"}] ,"aggregate": [{ "field": "grid_index", "expression": "count" ,"alias" : "count" }, { "field": "audience", "expression": "SUM" ,"alias" : "sum" ,"displayformat":",0" }, { "field": "audience", "expression": "average","alias" : "average" ,"displayformat":",0.00"}, { "field": "audience", "expression": "min" ,"alias" : "min","displayformat":",0"}, { "field": "audience", "expression": "max","alias" : "max" ,"displayformat":",0"}] , "columns" : [{"text":"Movie </pre>

	<pre> code", "alignment": "center", "width": 100, "field": "movieCd", "editor": {"type": "text", "alignment": "center"}, "summaryTemplate": "Count: #{count#}", "groupSummaryTemplate": "Count: #{count#}" ,{ "text": "Korea", "alignment": "center", "width": 100, "field": "movieNm", "edit or": {"type": "text", "alignment": "left"}} ,{ "text": "English", "alignment": "center", "width": 100, "field": "movieNmEn", " editor": {"type": "text", "alignment": "left"}} ,{ "text": "Year of production", "alignment": "center", "width": 80, "height": 20, "field": "prdtYear", "editor": {"type": "text", "alignment": "center"}} ,{ "text": "Audience", "alignment": "center", "width": 150, "field": "audience", "e ditor": {"type": "CalcEdit", "displayformat": "", "alignment": "right"}, "summaryTemplate": "Total Audience: #{sum#}", "groupSummaryTemplate": "Audience: #{sum#}" ,{ "text": "Country", "alignment": "center", "width": 200, "field": "nationAlt", "ed itor": {"type": "text", "alignment": "center"}, "summaryTemplate": "Average Audience: #{average#}", "groupSummaryTemplate": "Average Audience: #{average#}" ,{ "text": "Movie genre", "alignment": "center", "width": 150, "height": 20, "field": "genreAlt", "e ditor": {"type": "text", "alignment": "center"}, "summaryTemplate": "Min : #{min#} Max : #{max#}" } , "style": {} , "effect": {"columnHighlight": true} , "size": {"vertScrollWidth": 19} , "footer": {"show": true, "height": 24} }, "data": [{"movieCd": "20158982", "movieNm": "트릭", "movieNmEn": "Trick", "prdtYear": "2016", "openDt": "20160713", "typeNm": "장편", "prdtStatNm": "개봉", "nati onImg": "/css/images/Korea.png", "nationAlt": "한국", "genreAlt": "스릴러, 드 라마", "repNationNm": "한국", "repGenreNm": "스릴러", "directors": [{"peopleNm": "이창열"}], "companys": [{"companyCd": "20160481", "companyNm": "엘 씨오픽쳐스"}], . . . { "movieCd": "20166984", "movieNm": "위험한 유혹", "movieNmEn": "As I Lay Dying", "prdtYear": "2013", "openDt": "20161228", "typeNm": "장편", "prdtSta tNm": "개봉", "nationImg": "/css/images/America.png", "nationAlt": "미국", "g enreAlt": "드라마", "repNationNm": "미국", "repGenreNm": "드라마", "directors": [{"peopleNm": "제임스 프랭코"}], "companys": [] </pre>
--	--

</grid> </body> </html>	
options	
expression count item의 수 average 평균값 max 최댓값 min 최솟값	
display format ' 숫자가 세 자리 이상인 경우 세 자리마다 쉼표를 추가합니다. 0.00 숫자에 고정 소수점을 지정합니다.	

result

	Movie code	Korea	English	Year of pr..	Audience	Country	Movie genre
	Movie type : 올니버스 (1 rows)						
	Year of production : 2016 (1 rows)						
1	20158242	무서운 이야기 3 ..	Horror Stories III	2016	97,000	한국	공포(ホラ)
	Count: 1				Audience: 97,000	Average Audience: 97,000.00	
	Count: 1				Audience: 97,000	Average Audience: 97,000.00	
	Movie type : 장편 (99 rows)						
	Year of production : 2011 (1 rows)						
2	20164622	여가 사이클론 ..	SUPER STORM	2011	0	캐나다	액션,드라마,SF
	Count: 1				Audience: 0	Average Audience: 0.00	
	Year of production : 2013 (1 rows)						
3	20166984	위험한 유혹	As I Lay Dying	2013	0	미국	드라마
	Count: 1				Audience: 0	Average Audience: 0.00	
	Year of production : 2014 (2 rows)						
4	20141589	활판기행	End of Winter	2014	4,235	한국	드라마
5	20153467	미스 미스터리	Meet Miss Anjeli	2014	46	중국	드라마,일로/로맨스
	Count: 2				Audience: 4,281	Average Audience: 2,140.50	
	Year of production : 2015 (34 rows)						
6	20161722	히야	HYA	2015	21,495	한국	드라마
7	20158482	동주	DONGJU; The Po..	2015	1,176,920	한국	드라마
8	20145502	남과 여	A Man and A Wo..	2015	203,457	한국	일로/로맨스
9	20140704	비밀은 없다	The Truth Benea..	2015	250,958	한국	스릴러
10	20165521	국장만 프리즘스..		2015	9,629	한국,일본	애니메이션,드라마,판타지
	Count: 100				Total Audience: 121,137,596	Average Audience: 1,211,375.96	Min : 0 Max : 42,889,698
Σ	Count: 100				Total Audience: 121,137,596	Average Audience: 1,211,375.96	Min : 0 Max : 42,889,698

feature	examples
grouping	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="/Y[wai]/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> </head> <body> <grid id="grid1" style="height: 404px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":false,"width":30} ,"selector":{"show":false,"width":20} ,"group":[{ "field": "typeNm", "title" : "Movie type"} ,{ "field": "prdtYear", "title" : "Year of production"}] ,"columns" : [{ "text":"Movie code" ,"alignment":"center" ,"width":100 ,"field":"movieCd" ,"tooltip":false ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Movie Title" ,"alignment":"center" ,"width":200 ,"field":"movieNm" ,"tooltip":false ,"editor":{"type":"text","alignment":"left"} } ,{ "text":"Movie Title(English)" ,"alignment":"center" ,"width":200 ,"field":"movieNmEn" ,"tooltip":false ,"editor":{"type":"text","alignment":"left"} }] } </pre>

```

    }
    ,{"text":"Year of production"
    ,"alignment":"center"
    ,"width":80
    ,"field":"prdtYear"
    ,"tooltip":false
    ,"editor":{"type":"text","alignment":"center"}
    }
    ,{
    "text":"Open year"
    ,"alignment":"center"
    ,"width":80
    ,"field":"openDt"
    ,"tooltip":false

    ,"editor":{"type":"date","alignment":"center","dtOrder":"yyymmdd","dtSe
    parator":"/"}
    }
    ,{
    "text":"Movie type"
    ,"alignment":"center"
    ,"width":100
    ,"field":"typeNm"
    ,"style":""
    ,"tooltip":false
    ,"editor":{"type":"text","alignment":"center"}
    }
    ,{
    "text":"Production Status"
    ,"alignment":"center"
    ,"width":100
    ,"field":"prdtStatNm"
    ,"style":""
    ,"tooltip":false
    ,"editor":{"type":"text","alignment":"center"}
    }
    ,{
    "text":"Production Country"
    ,"alignment":"center"
    ,"width":87
    ,"field":"nationAlt"
    ,"style":""
    ,"tooltip":false
    ,"editor":{"type":"text","alignment":"center"}
    }
    ,{
    "text":"Movie genre"
    ,"alignment":"center"
    ,"width":100

```

```

,"field": "genreAlt"
,"tooltip": false
,"editor": { "type": "text", "alignment": "center" }
}
]
,"style": {}
,"effect": { "columnHighlight": true }
,"size": { "vertScrollWidth": 19 }
,"footer": { "show": true, "height": 24 }
},
"data": [

{"movieCd": "20158982", "movieNm": "트릭", "movieNmEn": "Trick", "prdtYear": "2016", "openDt": "20160713", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": "/css/images/Korea.png", "nationAlt": "한국", "genreAlt": "스릴러, 드라마", "repNationNm": "한국", "repGenreNm": "스릴러", "directors": [{"peopleNm": "이창열"}], "companys": [{"companyCd": "20160481", "companyNm": "엘씨오픽쳐스"}],
.
.
.

{"movieCd": "20166984", "movieNm": "위험한 유혹", "movieNmEn": "As I Lay Dying", "prdtYear": "2013", "openDt": "20161228", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": "/css/images/America.png", "nationAlt": "미국", "genreAlt": "드라마", "repNationNm": "미국", "repGenreNm": "드라마", "directors": [{"peopleNm": "제임스 프랭코"}], "companys": [
]
}
</grid>

</body>
</html>

```

result

Movie code	Movie Title	Movie Title(English)	Year of pr..	Open year	Movie type	Production St..	Production
Movie type : 올니버스 (1 rows)							
Year of production : 2016 (1 rows)							
20158242	루서운 이야기 3 : 화성에서 온 소녀	Horror Stories III	2016	2016/06/01		개봉	한국
Movie type : 장편 (99 rows)							
Year of production : 2011 (1 rows)							
20164622	여가 사이클론 스톰	SUPER STORM	2011	2016/11/21		개봉	캐나다
Year of production : 2013 (1 rows)							
20166984	위험한 유혹	As I Lay Dying	2013	2016/12/28		개봉	미국
Year of production : 2014 (2 rows)							
20141589	절환기 탈	End of Winter	2014	2016/04/21		개봉	한국
20153467	미스 히스테리	Meet Miss Anxiety	2014	2016/04/04		개봉	중국
Year of production : 2015 (34 rows)							
20161722	한아	HANA	2015	2016/03/10		개봉	한국
20158482	동주	DONGJU: The Portrait of A Poet	2015	2016/02/17		개봉	한국
20145502	남과 여	A Man and A Woman	2015	2016/02/25		개봉	한국
20140704	비밀은 없다	The Truth Beneath	2015	2016/06/23		개봉	한국
20165521	극장판 프리즘스쿨 솔라와 솔라	The Truth Beneath	2015	2016/07/07		개봉	한국,일본
20161961	나는 줄인이다	The Witness	2015	2016/09/14		개봉	한국,중국

feature	source
	example
filtering filtering된 내용만 표시합니다.	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="/Y[wai]/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> <script type="text/javascript"> \$(document).ready(function(){ grid1.setColumnFilterApply("prdtYear"); grid1.setColumnFilterApply("typeNm"); grid1.setColumnFilterApply("prdtStatNm"); }); </script> </head> <body> <grid id="grid1" style="height: 504px;width: 1000px; border: 1px solid rgb(234, 234, 234); position: relative;"> "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"contextmenu":true ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20,"type":"check"} ,"columns" : [{ "text":"Movie Title" ,"alignment":"center" ,"width":200 ,"field":"movieNm" ,"style":"" ,"editor":{"type":"text" ,"alignment":"left"} } ,{"text":"Year of production" ,"alignment":"center" ,"width":80 ,"field":"prdtYear" ,"style":"" ,"editor":{"type":"text" ,"alignment":"center"} } ,{ "text":"Movie type" </pre>

	<pre> ,"alignment":"center" ,"width":100 ,"field":"typeNm" ,"style":"" ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Production Status" ,"alignment":"center" ,"width":100 ,"field":"prdtStatNm" ,"style":"" ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Production Country" ,"alignment":"center" ,"width":87 ,"field":"nationAlt" ,"style":"" ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Movie genre" ,"alignment":"center" ,"width":100 ,"field":"genreAlt" ,"style":"" ,"editor":{"type":"text","alignment":"center"} }] ,"style":{} ,"effect":{"columnHighlight":true} ,"size":{"vertScrollWidth":19} ,"footer":{"show":true,"height":24} }, "data":[{"movieCd":"20158982","movieNm":"트릭","movieNmEn":"Trick","prdtYear":"2016","openDt":"20160713","typeNm":"장편","prdtStatNm":"개봉","nationImg":"/css/images/Korea.png","nationAlt":"한국","genreAlt":"스릴러,드라마","repNationNm":"한국","repGenreNm":"스릴러","directors":[{"peopleNm":"이창열"}],"companys":[{"companyCd":"20160481","companyNm":"엘씨오피쳐스"}], . . . {"movieCd":"20166984","movieNm":"위험한 유혹","movieNmEn":"As I Lay Dying","prdtYear":"2013","openDt":"20161228","typeNm":"장편","prdtSta </pre>
--	---

feature	examples
popup editing grid의 popup editor는 data source를 간단하게 확인하면서 data 편집 작업을 지원합니다.	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="https://uitool.org/wai/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> <script type="text/javascript"> /***** ***** * 초기 세팅 (TEST영역) ***** ******/ \$(document).ready(function(){ var json_data =[{CODE:"2011", CODE_NM:"2011"} ,{CODE:"2012", CODE_NM:"2012"} ,{CODE:"2013", CODE_NM:"2013"} ,{CODE:"2014", CODE_NM:"2014"} ,{CODE:"2015", CODE_NM:"2015"} ,{CODE:"2016", CODE_NM:"2016"} ,{CODE:"2017", CODE_NM:"2017"}] grid1.SetJsonStorage("years",json_data); grid1.onButtonClick = function(grid, column, Index, RecNo) { grid.popupEditor(RecNo); } }); /***** ***** * Grid Data Setting 영역 ***** ******/ </script> </head> <body> </pre>

	<pre> <h2>Popup editing</h2> <grid id="grid1" style="top:0px;height: 504px;width: 1000px; border: 1px solid rgb(234, 234, 234); position: relative;"> "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"contextmenu":true ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20,"type":"check"} ,"columns" : [{"text":"Movie code","alignment":"center","width":100,"field":"movieCd","editor":{"type": "text","alignment":"center"}} {"text":"Korea","editable":true,"alignment":"center","width":100,"field":" movieNm","editor":{"type":"text","alignment":"left"}} {"text":"English","editable":true,"alignment":"center","width":100,"field":" movieNmEn","editor":{"type":"text","alignment":"left"}} {"text":"Year of production","editable":true,"alignment":"center","width":80,"height":20,"f ield":"prdtYear" ,"editor":{"type":"LookupComboBox" ,"codeName":"years","keyfieldnames":"CODE" ,"dropdownwidth":"200px" ,"dropdownheight":"100px" ,"dispField":"CODE_NM" ,"droplist":"CODE,CODE,50,center;CODE_NM,CODE_NM,130,left","align ment":"center"}} {"text":"Open year","editable":true,"alignment":"center","width":80,"field":"openDt","edi tor":{"type":"date","alignment":"center","dtOrder":"yyyymmdd","dtSeparat or":"/"}} {"text":"Audience","editable":true,"alignment":"center","width":150,"field ":"audience","editor":{"type":"CalcEdit","displayformat":""," ,"alignment":"right"}} {"text":"","alignment":"center","width":100,"editable":false,"field":"" ,"editor":{"type":"button","alignment":"center","btnText":"Edit"}}] ,"style":{} ,"effect":{"columnHighlight":true} ,"size":{"vertScrollWidth":19} ,"footer":{"show":true,"height":24} }, "data":[{"movieCd":"20158982","movieNm":"트릭","movieNmEn":"Trick","prdtYear </pre>
--	--

```

:"2016","openDt":"20160713","typeNm":"장편","prdtStatNm":"개봉","nationImg":"./css/images/Korea.png","nationAlt":"한국","genreAlt":"스릴러,드라마","repNationNm":"한국","repGenreNm":"스릴러","directors":[{"peopleNm":"이창열"}],"companys":[{"companyCd":"20160481","companyNm":"엘씨오픽쳐스"}],
.
.
.
{movieCd":"20166984","movieNm":"위험한 유혹","movieNmEn":"As I Lay Dying","prdtYear":"2013","openDt":"20161228","typeNm":"장편","prdtStatNm":"개봉","nationImg":"./css/images/America.png","nationAlt":"미국","genreAlt":"드라마","repNationNm":"미국","repGenreNm":"드라마","directors":[{"peopleNm":"제임스 프랭코"}],"companys":[
]
}
</grid>

</body>
</html>

```

result

	Movie code	Korea	English	Year of pr..	Open year	Audience	
1	20158982	트릭	Trick	2016	2016/07/13	38,093	Edit
2	20161722	히야	HIYA	2015	2016/03/10	21,495	Edit
3	20152746	커터	Eclipse	2016	2016/03/30	19,399	Edit
4	20154910	터널	Tunnel	2016	2016/08/10	7,120,166	Edit
5	20144444	아가씨	The Handmaiden	2016	2016/06/01	42,889,698	Edit
6	20158482	종주	DONGJU: The Po..	2015	2016/02/17	1,176,920	Edit
7	20150879	역해술주	The Last Princess	2016	2016/08/03	5,599,606	Edit
8	20140194	곡성	THE WAILING	2016	2016/05/12	6,877,896	Edit
9	20169730	당신자신과 당신...	Yourself and Yours	2016	2016/11/10	180,000	Edit
10	20159646	탈경	The Age of Sha..	2016	2016/09/07	7,499,735	Edit
11	20153443	아수라	Asura - The City ..	2016	2016/09/28	2,594,749	Edit
12	20150901	출재	DETOUR	2016	2016/08/25	66,419	Edit
13	20145502	남과 여	A Man and A Wo..	2015	2016/02/25	203,457	Edit
14	20165723	메이킹 패밀리	Making Family	2016	2016/11/28	205	Edit
15	20140704	비밀은 없다	The Truth Benea..	2015	2016/06/23	250,958	Edit
16	20142406	탐정 홍길동: 사..	The Phantom De..	2016	2016/05/04	1,430,960	Edit
17	20164527	루돌! 루돌! 브루..	Vroom! Vroom! ..	2016	2016/12/15	11,000	Edit
18	20161694	제자 육함글 2 - ..		2016	2016/11/10	9,533	Edit
19	20162206	브루클린	Brooklyn	2016	2016/04/21	51,000	Edit
20	20165521	극장판 프리울스..		2015	2016/07/07	9,629	Edit
21	20155321	최악의 하루	Worst Woman	2016	2016/08/25	408	Edit
22	20161961	나는 증인이다	The Witness	2015	2016/09/14	1,407	Edit
23	20154803	우리 연애의 이력	With or Without ..	2015	2016/06/29	3,043	Edit

■ Selector

selector, selector radio

feature	example
selector grid의 row를 선택합니다.	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <link rel="stylesheet" type="text/css" href="http://uitool.org/wai/waiAll.css"> <script type="text/javascript" src="http://uitool.org/wai/wai.js"> </script> <link href="/css/default.css" rel="stylesheet" type="text/css"> <script> function cellClick(event){ if(event.column.type == "selector"){ var indexs = grid1.getCheckedRowsIndex(); \$('#infoNm1').html('< getCheckedRowsIndex() > Checked index number'); \$('#info1').html('grid1.getCheckedRowsIndex()

' + indexs); } } function getCheckedRows(){ var json = grid1.getCheckedRows(); \$('#infoNm1').html('< getCheckedRows() > Output the contents of the checked row'); \$('#info1').html('[grid1.getCheckedRows()]

' + JSON.stringify(json)); } function getCheckRows(rownum){ var checked = grid1.getCheckRows(rownum); if(checked == 0) checked = "true"; else if(checked == -1) checked = "false"; \$('#infoNm1').html('< getCheckedRows() > Output the contents of the checked row'); \$('#info1').html('[grid1.getCheckedRows()]

' + checked); } </pre>

```

function setCheckRows(rownum){
    var checked = grid1.getCheckRows(rownum);
    if(checked == 0) checked = "false";
    else if(checked == -1) checked = "true";
    grid1.setCheckRows(rownum , checked);

    var indexs = grid1.getCheckedRowsIndex();

    $('#infoNm1').html('< getCheckedRowsIndex() > Checked index
number');
    $('#info1').html('grid1.getCheckedRowsIndex()<br><br><br>' + indexs
);
}
</script>
</head>
<body>
<grid id="grid1" onclick="cellClick(event)" style="height:
204px;width:700px; border: 1px solid rgb(234, 234, 234);" >
    "options" : {
        "editable":false
        ,"columnHeight":20
        ,"cellHeight":20
        ,"fixColumn":0
        ,"indicator":{"show":false,"width":30}
        ,"selector":{"show":true,"width":20}
        ,"columns" : [
            {
                "text":"Movie Title"
                ,"alignment":"center"
                ,"width":200
                ,"field":"movieNm"
                ,"style":""
                ,"editor":{"type":"text","alignment":"left"}
            }
            ,{"text":"Year of production"
                ,"alignment":"center"
                ,"width":80
                ,"field":"prdtYear"
                ,"style":""
                ,"editor":{"type":"text","alignment":"center"}
            }
            ,{
                "text":"Movie type"
                ,"alignment":"center"
                ,"width":100
                ,"field":"typeNm"
                ,"style":""
                ,"editor":{"type":"text","alignment":"center"}
            }
        ]
    }

```

```

,{
  "text":"Production Status"
  ,"alignment":"center"
  ,"width":100
  ,"field":"prdtStatNm"
  ,"style":""
  ,"editor":{"type":"text","alignment":"center"}
}
,{
  "text":"Production Country"
  ,"alignment":"center"
  ,"width":87
  ,"field":"nationAlt"
  ,"style":""
  ,"editor":{"type":"text","alignment":"center"}
}
,{
  "text":"Movie genre"
  ,"alignment":"center"
  ,"width":100
  ,"field":"genreAlt"
  ,"style":""
  ,"editor":{"type":"text","alignment":"center"}
}
]
,"style":{}
,"effect":{"columnHighlight":true}
,"size":{"vertScrollWidth":19}
,"footer":{"show":true,"height":24}
},
"data":[

{"movieCd":"20158982","movieNm":"트릭","movieNmEn":"Trick","prdtYear":"2016","openDt":"20160713","typeNm":"장편","prdtStatNm":"개봉","nationImg":"/css/images/Korea.png","nationAlt":"한국","genreAlt":"스릴러,드라마","repNationNm":"한국","repGenreNm":"스릴러","directors":[{"peopleNm":"이창열"}],"companys":[{"companyCd":"20160481","companyNm":"엘씨오피컬스"}],
.
.
.
{"movieCd":"20166984","movieNm":"위험한 유혹","movieNmEn":"As I Lay Dying","prdtYear":"2013","openDt":"20161228","typeNm":"장편","prdtStatNm":"개봉","nationImg":"/css/images/America.png","nationAlt":"미국","genreAlt":"드라마","repNationNm":"미국","repGenreNm":"드라마","directors":[{"peopleNm":"제임스 프랭코"}],"companys":[]
}
]
</grid>

```

```

<div style="margin:20px">
    Output the contents of the checked row : <button
onClick='getCheckedRows()'>getCheckedRows</button>
</div>
<div style="margin:20px">
    Change the check state of the row : <input id="Inp_setCheckRows"
style="width:30px" value="0"> <button
onClick='setCheckRows(Inp_setCheckRows.value)'>setCheckRows</butto
n>
</div>
<div style="margin:20px">
    Make sure the row is checked : <input id="Inp_getCheckRows"
style="width:30px" value="0"> <button
onClick='getCheckRows(Inp_getCheckRows.value)'>getCheckRows</butto
n>
</div>

<div style="margin: 20px; border: 1px solid; background: rgb(250, 250,
250);">
    <div id="infoNm1" style="margin: 20px; color: red; font-size:
16px;"></div>
    <div id="info1" style="margin: 20px; font-size: 13px;word-wrap:
break-word;">

    </div>
</div>

</body>
</html>

```

result

	Movie Title	Year of pr..	Movie type	Production St..	Production ..	Movie genre
☒	트릭	2016	장편	가봉	한국	스릴러,드라마
☐	죄마	2015	장편	가봉	한국	드라마
☐	커넥트	2016	장편	가봉	한국	범죄, 드라마
☐	타겟	2016	장편	가봉	한국	드라마
☐	아가씨	2016	장편	가봉	한국	스릴러,드라마
☐	올주	2015	장편	가봉	한국	드라마
☐	박해종주	2016	장편	가봉	한국	사극, 드라마
☐	국경	2016	장편	가봉	한국	미스터리,스릴러..

feature	example
selector radio grid에서 하나의 row만 선택합니다.	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <link rel="stylesheet" type="text/css" href="http://uitool.org/wai/waiAll.css"> <script type="text/javascript" src="http://uitool.org/wai/wai.js"> </script> <link href="/css/default.css" rel="stylesheet" type="text/css"> <script> function cellClick(event){ if(event.column.type == "selector"){ var indexs = grid1.getCheckedRowsIndex(); \$('#infoNm1').html('< getCheckedRowsIndex() > Checked index number'); \$('#info1').html('grid1.getCheckedRowsIndex()

' + indexs); } } function getCheckedRows(){ var json = grid1.getCheckedRows(); \$('#infoNm1').html('< getCheckedRows() > Output the contents of the checked row'); \$('#info1').html('[grid1.getCheckedRows()]

' + JSON.stringify(json)); } function getCheckRows(rownum){ var checked = grid1.getCheckRows(rownum); if(checked == 0) checked = "true"; else if(checked == -1) checked = "false"; \$('#infoNm1').html('< getCheckedRows() > Output the contents of the checked row'); \$('#info1').html('[grid1.getCheckedRows()]

' + checked); } function setCheckRows(rownum){ var checked = grid1.getCheckRows(rownum); if(checked == 0) checked = "false"; else if(checked == -1) checked = "true"; </pre>


```

grid1.setCheckRows(rownum , checked);

var indexs = grid1.getCheckedRowsIndex();

$('#infoNm1').html('< getCheckedRowsIndex() > Checked index
number');
$('#info1').html('grid1.getCheckedRowsIndex()<br><br><br>' + indexs
);
}
</script>
</head>
<body>
<grid id="grid1" onclick="cellClick(event)" style="height:
204px;width:700px; border: 1px solid rgb(234, 234, 234);" >
"options" : {
"editable":false
,"columnHeight":20
,"cellHeight":20
,"fixColumn":0
,"indicator":{"show":false,"width":30}
,"selector":{"show":true,"width":20, "type":"radio"}
,"columns" : [
{
"text":"Movie Title"
,"alignment":"center"
,"width":200
,"field":"movieNm"
,"style":""
,"editor":{"type":"text", "alignment":"left"}
}
,{"text":"Year of production"
,"alignment":"center"
,"width":80
,"field":"prdtYear"
,"style":""
,"editor":{"type":"text", "alignment":"center"}
}
,{
"text":"Movie type"
,"alignment":"center"
,"width":100
,"field":"typeNm"
,"style":""
,"editor":{"type":"text", "alignment":"center"}
}
,{
"text":"Production Status"
,"alignment":"center"
,"width":100

```

```

, "field": "prdtStatNm"
, "style": ""
, "editor": { "type": "text", "alignment": "center" }
}
, {
  "text": "Production Country"
, "alignment": "center"
, "width": 87
, "field": "nationAlt"
, "style": ""
, "editor": { "type": "text", "alignment": "center" }
}
, {
  "text": "Movie genre"
, "alignment": "center"
, "width": 100
, "field": "genreAlt"
, "style": ""
, "editor": { "type": "text", "alignment": "center" }
}
]
, "style": {}
, "effect": { "columnHighlight": true }
, "size": { "vertScrollWidth": 19 }
, "footer": { "show": true, "height": 24 }
},
"data": [

{ "movieCd": "20158982", "movieNm": "트릭", "movieNmEn": "Trick", "prdtYear": "2016", "openDt": "20160713", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": "/css/images/Korea.png", "nationAlt": "한국", "genreAlt": "스릴러, 드라마", "repNationNm": "한국", "repGenreNm": "스릴러", "directors": [ { "peopleNm": "이창열" } ], "companys": [ { "companyCd": "20160481", "companyNm": "엘씨오픽쳐스" } ],
.
.
.

{ "movieCd": "20166984", "movieNm": "위험한 유혹", "movieNmEn": "As I Lay Dying", "prdtYear": "2013", "openDt": "20161228", "typeNm": "장편", "prdtStatNm": "개봉", "nationImg": "/css/images/America.png", "nationAlt": "미국", "genreAlt": "드라마", "repNationNm": "미국", "repGenreNm": "드라마", "directors": [ { "peopleNm": "제임스 프랭코" } ], "companys": [
]
}
]
</grid>

<div style="margin:20px">
  Output the contents of the checked row : <button
onClick='getCheckedRows()'>getCheckedRows</button>

```

```

</div>
<div style="margin:20px">
    Change the check state of the row : <input id="Inp_setCheckRows"
    style="width:30px" value="0"> <button
    onClick='setCheckRows(Inp_setCheckRows.value)')setCheckRows</button>
</div>
<div style="margin:20px">
    Make sure the row is checked : <input id="Inp_getCheckRows"
    style="width:30px" value="0"> <button
    onClick='getCheckRows(Inp_getCheckRows.value)')getCheckRows</button>
</div>

<div style="margin: 20px; border: 1px solid; background: rgb(250, 250,
250);">
    <div id="infoNm1" style="margin: 20px; color: red; font-size:
16px;"></div>
    <div id="info1" style="margin: 20px; font-size: 13px;word-wrap:
break-word;">

</div>
</div>

</body>
</html>

```

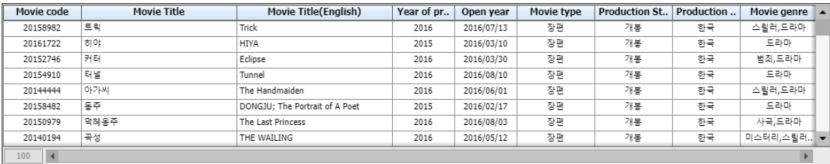
result

	Movie Title	Year of pr..	Movie type	Production St..	Production ..	Movie genre
☒	트릭	2016	장편	가봉	한국	스릴러,드라마
☒	피아	2015	장편	가봉	한국	드라마
☒	카타	2016	장편	가봉	한국	범죄,드라마
☒	타넬	2016	장편	가봉	한국	드라마
☒	아가씨	2016	장편	가봉	한국	스릴러,드라마
☒	올주	2015	장편	가봉	한국	드라마
☒	악몽을꾸	2016	장편	가봉	한국	사극,드라마
☒	극성	2016	장편	가봉	한국	미스터리,스릴러...

■ Excel

excel download, excel upload, excel upload nohead

feature	result
excel download GridID.exportExcel("Download file name");	

feature	example
excel upload data만 upload합니다.	<pre>function ExcelUpload(){ // set on grid in callback function var file = document.getElementById("excel").files[0]; // console.log(document.getElementById("excel").value ,file); \$wai.util.ExcelUpload(file, callback) ; } function callback (data){ var j = JSON.parse(data); grid2.SetData(j.data); }</pre>
result	
	

feature	example
excel upload 모두 upload합니다.	alignment fieldName ;,0.00 ,: #,##0 0.00 : 0.00 alignment < : left ^ : center > : right function ExcelUpload(){ // set on grid in callback function var file = document.getElementById("excel").files[0]; // console.log(document.getElementById("excel").value ,file); \$wai.util.ExcelUpload(file, callback) ; } function callback (data){ grid2.SetGrid(data); }

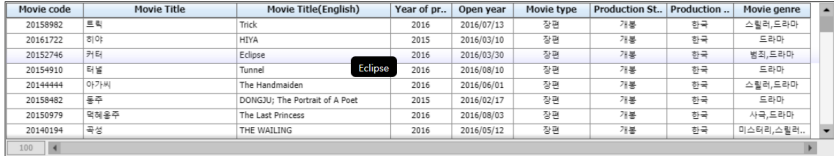
result

movieCd	movieNm	movieNm..	prdtYear	openDt	typeNm	prdtStat..	nationAlt	genreAlt
20158982	트릭	Trick	2016	20160713	장편	개봉	한국	스릴러,드라..
20161722	민아	HIYA	2015	20160310	장편	개봉	한국	로리극
20152746	카타	Eclipse	2016	20160330	장편	개봉	한국	범죄,로리극
20154910	터널	Tunnel	2016	20160810	장편	개봉	한국	로리극
20144444	아가씨	The Handm...	2016	20160601	장편	개봉	한국	스릴러,드라..
20158482	올주	DONGJU, T...	2015	20160217	장편	개봉	한국	로리극
20150979	국립올주	The Last Pri...	2016	20160803	장편	개봉	한국	사극,로리극
20140194	국경	THE WALL...	2016	20160512	장편	개봉	한국	미스터리,스...

feature	example																																																																																	
excel upload nohead excel에 field 이름이 정의되어 있지 않은 경우	<pre>function ExcelUploadNoHead(){ // field name is not defined in excel // upload excel separately from the grid // set on grid in callback function // ExcelUploadNoHead (file, "field name definition", callback function after upload complete, start row, start column); var file = document.getElementById("excel").files[0]; \$wai.util.ExcelUploadNoHead(file,"movieCd,movieNm,movieNmEn,prdtYear,openDt,typeNm,prdtStatNm ,nationAlt,genreAlt", NoHeadcallback , 3,2) ; } function NoHeadcallback (data){ var j = JSON.parse(data); j = j.data; grid4.SetData(j); }</pre>																																																																																	
result																																																																																		
<table><tr><th>Movie code</th><th>Movie Title</th><th>Movie Title(English)</th><th>Year of pr..</th><th>Open year</th><th>Movie type</th><th>Production ST..</th><th>Production ..</th><th>Movie genre</th></tr><tr><td>20158982</td><td>트릭</td><td>Trick</td><td>2016</td><td>2016/07/13</td><td>장편</td><td>가봉</td><td>한국</td><td>스릴러,드라마</td></tr><tr><td>20161722</td><td>히타</td><td>HITVA</td><td>2015</td><td>2016/03/10</td><td>장편</td><td>가봉</td><td>한국</td><td>드라마</td></tr><tr><td>20152746</td><td>커리</td><td>Eclipse</td><td>2016</td><td>2016/03/30</td><td>장편</td><td>가봉</td><td>한국</td><td>범죄, 드라마</td></tr><tr><td>20154910</td><td>터널</td><td>Tunnel</td><td>2016</td><td>2016/08/10</td><td>장편</td><td>가봉</td><td>한국</td><td>드라마</td></tr><tr><td>20144444</td><td>아가씨</td><td>The Handmaiden</td><td>2016</td><td>2016/06/01</td><td>장편</td><td>가봉</td><td>한국</td><td>스릴러,드라마</td></tr><tr><td>20158482</td><td>동주</td><td>DONGJU: The Portrait of A Poet</td><td>2015</td><td>2016/02/17</td><td>장편</td><td>가봉</td><td>한국</td><td>드라마</td></tr><tr><td>20150979</td><td>옥녀음주</td><td>The Last Princess</td><td>2016</td><td>2016/08/03</td><td>장편</td><td>가봉</td><td>한국</td><td>사극,드라마</td></tr><tr><td>20140194</td><td>곡성</td><td>THE WAILING</td><td>2016</td><td>2016/05/12</td><td>장편</td><td>가봉</td><td>한국</td><td>미스터리,스릴러...</td></tr></table>		Movie code	Movie Title	Movie Title(English)	Year of pr..	Open year	Movie type	Production ST..	Production ..	Movie genre	20158982	트릭	Trick	2016	2016/07/13	장편	가봉	한국	스릴러,드라마	20161722	히타	HITVA	2015	2016/03/10	장편	가봉	한국	드라마	20152746	커리	Eclipse	2016	2016/03/30	장편	가봉	한국	범죄, 드라마	20154910	터널	Tunnel	2016	2016/08/10	장편	가봉	한국	드라마	20144444	아가씨	The Handmaiden	2016	2016/06/01	장편	가봉	한국	스릴러,드라마	20158482	동주	DONGJU: The Portrait of A Poet	2015	2016/02/17	장편	가봉	한국	드라마	20150979	옥녀음주	The Last Princess	2016	2016/08/03	장편	가봉	한국	사극,드라마	20140194	곡성	THE WAILING	2016	2016/05/12	장편	가봉	한국	미스터리,스릴러...
Movie code	Movie Title	Movie Title(English)	Year of pr..	Open year	Movie type	Production ST..	Production ..	Movie genre																																																																										
20158982	트릭	Trick	2016	2016/07/13	장편	가봉	한국	스릴러,드라마																																																																										
20161722	히타	HITVA	2015	2016/03/10	장편	가봉	한국	드라마																																																																										
20152746	커리	Eclipse	2016	2016/03/30	장편	가봉	한국	범죄, 드라마																																																																										
20154910	터널	Tunnel	2016	2016/08/10	장편	가봉	한국	드라마																																																																										
20144444	아가씨	The Handmaiden	2016	2016/06/01	장편	가봉	한국	스릴러,드라마																																																																										
20158482	동주	DONGJU: The Portrait of A Poet	2015	2016/02/17	장편	가봉	한국	드라마																																																																										
20150979	옥녀음주	The Last Princess	2016	2016/08/03	장편	가봉	한국	사극,드라마																																																																										
20140194	곡성	THE WAILING	2016	2016/05/12	장편	가봉	한국	미스터리,스릴러...																																																																										

■ Tooltip

data 영역에 마우스를 가져가면 tooltip이 표시됩니다.

feature	source
tooltip	example
	"tooltip":true
	columns
	<pre> "columns" : [{ "text":"Movie code" ,"alignment":"center" ,"width":100 ,"type":"column" ,"height":20 ,"editable":false ,"visible":true ,"sortable":true ,"sortType":"string" ,"field":"movieCd" ,"style":"" ,"tooltip":true ,"editor":{"type":"text","alignment":"center"} }] </pre>
result	
	

■ Conditional Style

feature	source
conditional style row style을 변경합니다.	example
	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="/Y[wai]/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> </head> <body> <grid id="grid1" style="height: 204px;width:700px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20} ,"selection":{"rowSelect":true,"multiSelect":false} ,"conditionalstyle":[{"field":"grid_index","conditional":"value % 2 == 0","color":"rgba(250, 250, 250, 1)"} ,{"field":"grid_index","conditional":"value % 2 != 0","color":"rgba(0, 240, 240 , 0.2)"}] ,"columns" : [{ "text":"Movie Title" ,"alignment":"center" ,"width":200 ,"field":"movieNm" ,"style":"" ,"editor":{"type":"text","alignment":"left"} } ,{"text":"Year of production" ,"alignment":"center" ,"width":80 ,"field":"prdtYear" ,"style":"" ,"editor":{"type":"text","alignment":"center"} }] } </pre>

	<pre> ,{ "text":"Movie type" ,"alignment":"center" ,"width":100 ,"field":"typeNm" ,"style":"" ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Production Status" ,"alignment":"center" ,"width":100 ,"field":"prdtStatNm" ,"style":"" ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Production Country" ,"alignment":"center" ,"width":87 ,"field":"nationAlt" ,"style":"" ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Movie genre" ,"alignment":"center" ,"width":100 ,"field":"genreAlt" ,"style":"" ,"editor":{"type":"text","alignment":"center"} }] ,"style":{} ,"effect":{"columnHighlight":true} ,"size":{"vertScrollWidth":19} ,"footer":{"show":true,"height":24} } ,"data":[{"movieCd":"20158982","movieNm":"트릭","movieNmEn":"Trick","prdtYear":"2016","openDt":"20160713","typeNm":"장편","prdtStatNm":"개봉","nationImg":"/css/images/Korea.png","nationAlt":"한국","genreAlt":"스릴러,드라마","repNationNm":"한국","repGenreNm":"스릴러","directors":[{"peopleNm":"이창열"}],"companys":[{"companyCd":"20160481","companyNm":"엘씨오피컷스"}], . . . </pre>
--	--

feature	example
conditional style '2015년도 작품'과 '스릴러'의 스타일을 변경합니다.	<pre> <!DOCTYPE html> <html lang="en" style="width:1280px;height:1024px"> <head> <meta charset="UTF-8"> <title></title> <script type="text/javascript" src="/Y[wai]/wai.js"></script> <link href="/css/default.css" rel="stylesheet" type="text/css"> </head> <body> <grid id="grid1" style="height: 204px;width:700px; border: 1px solid rgb(234, 234, 234);" > "options" : { "editable":false ,"columnHeight":20 ,"cellHeight":20 ,"fixColumn":0 ,"indicator":{"show":true,"width":30} ,"selector":{"show":false,"width":20} ,"conditionalstyle":[{"field":"prdtYear","conditional":"value <= 2015","color":"rgba(0, 250, 0, 0.5)","range":"cell"} ,{"field":"genreAlt","conditional":"value.indexOf('스릴러') > -1","color":"rgba(250, 250, 0, 0.5)","range":"cell"}] ,"columns" : [{ "text":"Movie Title" ,"alignment":"center" ,"width":200 ,"field":"movieNm" ,"style":"" ,"editor":{"type":"text","alignment":"left"} } ,{"text":"Year of production" ,"alignment":"center" ,"width":80 ,"field":"prdtYear" ,"style":"" ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Movie type" ,"alignment":"center" ,"width":100 ,"field":"typeNm" </pre>

	<pre> ,"style":"" ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Production Status" ,"alignment":"center" ,"width":100 ,"field":"prdtStatNm" ,"style":"" ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Production Country" ,"alignment":"center" ,"width":87 ,"field":"nationAlt" ,"style":"" ,"editor":{"type":"text","alignment":"center"} } ,{ "text":"Movie genre" ,"alignment":"center" ,"width":100 ,"field":"genreAlt" ,"style":"" ,"editor":{"type":"text","alignment":"center"} } } ,"style":{} ,"effect":{"columnHighlight":true} ,"size":{"vertScrollWidth":19} ,"footer":{"show":true,"height":24} } ,"data":[{"movieCd":"20158982","movieNm":"트릭","movieNmEn":"Trick","prdtYear":"2016","openDt":"20160713","typeNm":"장편","prdtStatNm":"개봉","nationImg":"/css/images/Korea.png","nationAlt":"한국","genreAlt":"스릴러,드라마","repNationNm":"한국","repGenreNm":"스릴러","directors":[{"peopleNm":"이창열"}],"companys":[{"companyCd":"20160481","companyNm":"엘씨오피쳐스"}], . . . {"movieCd":"20166984","movieNm":"위험한 유혹","movieNmEn":"As I Lay Dying","prdtYear":"2013","openDt":"20161228","typeNm":"장편","prdtStatNm":"개봉","nationImg":"/css/images/America.png","nationAlt":"미국","genreAlt":"드라마","repNationNm":"미국","repGenreNm":"드라마","directors":[{"peopleNm":"제임스 프랭코"}],"companys":[</pre>
--	---

]
</grid>

</body>
</html>

result

	Movie Title	Year of pr..	Movie type	Production St..	Production ..	Movie genre ▲
1	트릭	2016	장편	개봉	한국	스릴러,드라마
2	미아	2015	장편	개봉	한국	드라마
3	커터	2016	장편	개봉	한국	범죄,드라마
4	터널	2016	장편	개봉	한국	드라마
5	아가씨	2016	장편	개봉	한국	스릴러,드라마
6	동주	2015	장편	개봉	한국	드라마
7	국제시장	2016	장편	개봉	한국	사극,드라마
8	곡성	2016	장편	개봉	한국	미스터리,스릴러 ▼

100

■ Methods

SetGrid, GetGrid, Clear, SelectRow, SetValue, GetValue, AddColumn, DelColumn, RecordCount, DeleteRow, IndicatorShow, SelectorShow, SetColumnVisible, GetSelection, getSelectIndex, getSelectColumn

method	source
SetGrid	grid1.SetData(movieList);

result

영화코드	영화명	영화명(영문)	제작연도	개봉연도	영화유형	제작상태
20158982	트릭	Trick	2016	2016/07/13	장편	개봉
20161722	히야	HIYA	2015	2016/03/10	장편	개봉
20152746	커터	Eclipse	2016	2016/03/30	장편	개봉
20154910	터널	Tunnel	2016	2016/08/10	장편	개봉
20144444	아가씨	The Handmaiden	2016	2016/06/01	장편	개봉
20158482	동주	DONGJU: The Portrait of A Poet	2015	2016/02/17	장편	개봉
20150979	덕혜옹주	The Last Princess	2016	2016/08/03	장편	개봉
20140194	곡성	THE WAILING	2016	2016/05/12	장편	개봉

100

method	source
GetGrid	grid1.GetData();
result	
<pre> < getData() > [소스 : grid의 전체 데이터를 가지고 옵니다.] var node = grid1.getData(); [결과 : object로 return 됩니다.] [{"movieCd":"20158982","movieNm":"트릭","movieNmEn":"Trick","prdtYear":"2016","openDt":"20160713","typeNm":"장편","prdtStatNm":"개봉","nationAlt":"한국","genreAlt":"스릴러,드라마","repNationNm":"한국","repGenreNm":"스릴러","directors":[{"peopleNm":"이장열"}],"companys":[{"companyCd":"20160481","companyNm":"엘씨오피컬처스"}]},{"movieCd":"20161722","movieNm":"히야","movieNmEn":"HIYA","prdtYear":"2015","openDt":"20160310","typeNm":"장편","prdtStatNm":"개봉","nationAlt":"한국","genreAlt":"드라마","repNationNm":"한국","repGenreNm":"드라마","directors":[{"peopleNm":"김지연"}],"companys":[{"companyCd":"20168888","companyNm":"(주)메이저타운"}]},{"movieCd":"20169148","companyNm":"(주)더재미팩토리"}]},{"companyCd":"20100392","companyNm":"(주)알지엔터웍스"}]},{"movieCd":"20152746","movieNm":"커터","movieNmEn":"Eclipse","prdtYear":"2016","openDt":"20160330","typeNm":"장편","prdtStatNm":"개봉","nationAlt":"한국","genreAlt":"범죄,드라마","repNationNm":"한국","repGenreNm":"범죄","directors":[{"peopleNm":"정희성"}],"companys":[{"companyCd":"20160208","companyNm":"(주)에픽픽처스"}]},{"movieCd":"20154910","movieNm":"터널"}]</pre>	

method	source
Clear	grid1.clear();
result	
component에 data가 삭제됩니다.	

method	source
SelectRow	grid1.SelectRow(1);
result	
 두 번째 줄이 선택됩니다.	

method	source
SetValue (field, value)	grid1.SetValue(grid1.getSelectedColumn().field,InpSelectVal);
result	

영화코드	영화명	영화명(영문)
20158982	트릭	Trick
20161722	히야	HIYA
20152746	커타	Eclipse
20154910	터널	Tunnel
20144444	아가씨	The Handmaiden
20158482	동주	DONGJU: The Portrait of a Poet
20150979	말죽거리	The Last Princess

선택한 Row에 지정된 Cell :

트릭

SetValue

설정할 cell을 선택합니다.

script source를 실행합니다.

해당 cell의 value가 변경됩니다.

method	source
GetValue	<pre>var FieldName = grid1.getSelectedColumn().field; returnValue = grid1.GetValue(FieldName);</pre>

result

Movie code	Movie Title	Movie Title(English)
20158982	트릭	Trick
20161722	히야	HIYA
20152746	커타	Eclipse
20154910	터널	Tunnel
20144444	아가씨	The Handmaiden
20158482	동주	DONGJU: The Portrait of A Poet
20150979	말죽거리	The Last Princess
20140194	곡성	THE WAILING

100

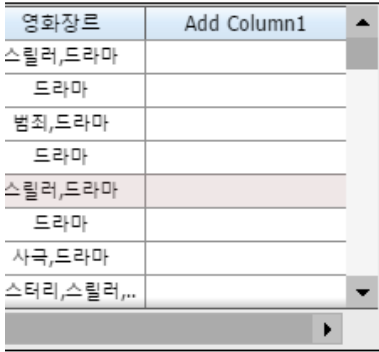
< GetValue(FieldName) >

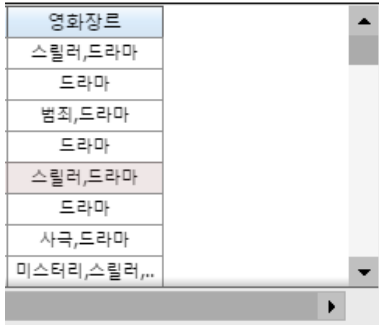
[Source : Returns the Field value of the selected row.]

var FieldName = grid1.getSelectedColumn().field;
grid1.GetValue(FieldName);

[Result : It is returned as text.]

The Handmaiden

method	source
AddColumn (column)	<pre>grid1.AddColumn({'text':"Add Column' + ++addFieldCount + ' ', "field": "addCol' + addFieldCount +'", "alignment":"center", "width":120, "height":20 , "visible":true, "editor":{"type":"text", "alignment":"left"}}); addFieldNames.push("addCol" + addFieldCount);</pre>
result	
	

method	source
DelColumn (fieldName)	<pre>grid1.DelColumn(addFieldNames.pop());</pre>
result	
	

method	source
RecordCount	grid1.RecordCount();

result

Movie code	Movie Title	Movie Title(English)
20158982	트릭	Trick
20161722	히야	HIYA
20152746	커터	Eclipse
20154910	터널	Tunnel
20144444	아가씨	The Handmaiden
20158482	동주	DONGJU; The Portrait of A Poet
20150979	덕혜옹주	The Last Princess
20140194	곡성	THE WAILING

100

< RecordCount() >

[Source : Returns the number of rows in the Grid.]

var node = grid1.RecordCount();

[Result : It is returned as text.]

100

method	source
DeleteRow	grid1.DeleteRow()

result

영화코드	영화명	영화명(영문)	제작연도	개봉연도	영화유형
20158982	트릭	Trick	2016	2016/07/13	장편
20161722	히야	HIYA	2015	2016/03/10	장편
20152746	커터	Eclipse	2016	2016/03/30	장편
20154910	터널	Tunnel	2016	2016/08/10	장편
20144444	아가씨	The Handmaiden	2016	2016/06/01	장편
20158482	동주	DONGJU; The Portrait of A Poet	2015	2016/02/17	장편
20150979	덕혜옹주	The Last Princess	2016	2016/08/03	장편
20140194	곡성	THE WAILING	2016	2016/05/12	장편

100

영화코드	영화명	영화명(영문)	제작연도	개봉연도	영화유형
20158982	트릭	Trick	2016	2016/07/13	장편
20161722	히야	HIYA	2015	2016/03/10	장편
20154910	터널	Tunnel	2016	2016/08/10	장편
20144444	아가씨	The Handmaiden	2016	2016/06/01	장편
20158482	동주	DONGJU; The Portrait of A Poet	2015	2016/02/17	장편
20150979	덕혜옹주	The Last Princess	2016	2016/08/03	장편
20140194	곡성	THE WAILING	2016	2016/05/12	장편
20169730	당신자신과 당신의 것	Yourself and Yours	2016	2016/11/10	장편

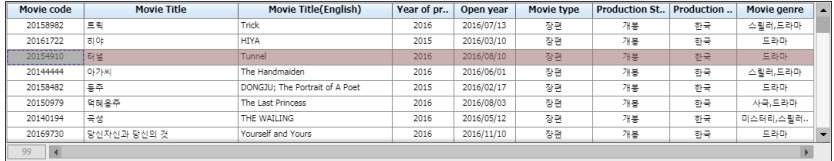
99

선택된 row가 삭제됩니다.

method	source
IndicatorShow (visible)	grid1.IndicatorShow(true);
result	
 grid의 첫 번째 column에 row의 순서가 표시됩니다.	

method	source
SelectorShow (visible)	grid1.SelectorShow(true);
result	
 grid의 첫 번째 column에 checkbox가 생성됩니다.	

method	source
SetColumnVisible (index, visible)	grid1.setColumnVisible(2, false);
result	
	
	
column index가 2인 영화명(영문) column이 숨겨집니다.	

method	source
GetSelection	grid1.getSelection();
result	
	
<pre>< getSelection() > [Source : Returns the selected Record.] var node = grid1.getSelection(); [Result : The selected record is returned to JSON.] {"movieCd":"20154910","movieNm":"터널","movieNmEn":"Tunnel","prdtYear":"2016","openDt":"20160810","typeNm":"장편","prdtStatNm":"개봉","nationAlt":"한국","genreAlt":"드라마","repNationNm":"한국","repGenreNm":"드라마","directors":[{"peopleNm":"김성훈"}],"companies":[{"companyCd":"20100386","companyNm":"(주)아나더스토리"}],"companyCd":"20138935","companyNm":"(주)비아이엔터테인먼트"}</pre>	

method	source
getSelectIndex	grid1.getSelectIndex();

result

Movie code	Movie Title	Movie Title(English)	Y
20158982	트릭	Trick	
20161722	히야	HIYA	
20154910	터널	Tunnel	
20144444	아가씨	The Handmaiden	
20158482	동주	DONGJU; The Portrait of A Poet	
20150979	역적:올주	The Last Princess	
20140194	곡성	THE WALLING	
20169730	당신자신과 당신의 것	Yourself and Yours	

< getSelectIndex() >

[Source : Returns the index of the selected record.]

```
var node = grid1.getSelectIndex();
```

[Result : Index is returned as text.]

3

method	source
getSelectColumn	grid1.getSelectColumn();

result																																																																																	
<table><tr><th>Movie code</th><th>Movie Title</th><th>Movie Title(English)</th><th>Year of pr..</th><th>Open year</th><th>Movie type</th><th>Production St..</th><th>Production ..</th><th>Movie genre</th></tr><tr><td>20158982</td><td>트릭</td><td>Trick</td><td>2016</td><td>2016/07/13</td><td>장편</td><td>가봉</td><td>한국</td><td>스릴러,드라마</td></tr><tr><td>20161722</td><td>히야</td><td>HIYA</td><td>2015</td><td>2016/03/10</td><td>장편</td><td>가봉</td><td>한국</td><td>드라마</td></tr><tr><td>20152746</td><td>카타</td><td>Eclipse</td><td>2016</td><td>2016/03/30</td><td>장편</td><td>가봉</td><td>한국</td><td>범죄,드라마</td></tr><tr><td>20154910</td><td>터널</td><td>Tunnel</td><td>2016</td><td>2016/08/10</td><td>장편</td><td>가봉</td><td>한국</td><td>드라마</td></tr><tr><td>20144444</td><td>아가씨</td><td>The Handmaiden</td><td>2016</td><td>2016/06/01</td><td>장편</td><td>가봉</td><td>한국</td><td>스릴러,드라마</td></tr><tr><td>20158482</td><td>동주</td><td>DONGJU: The Portrait of A Poet</td><td>2015</td><td>2016/02/17</td><td>장편</td><td>가봉</td><td>한국</td><td>드라마</td></tr><tr><td>20150979</td><td>역적:올주</td><td>The Last Princess</td><td>2016</td><td>2016/08/03</td><td>장편</td><td>가봉</td><td>한국</td><td>사극,드라마</td></tr><tr><td>20140194</td><td>곡성</td><td>THE WALLING</td><td>2016</td><td>2016/05/12</td><td>장편</td><td>가봉</td><td>한국</td><td>미스터리,스릴러..</td></tr></table> 100 < getSelectColumn() > [Source : Returns the selected Column.] var node = grid1.getSelectColumn(); [Result : The selected column is returned as JSON.] { "text": "Movie code", "alignment": "center", "width": 100, "type": "column", "height": 20, "editable": false, "visible": true, "sortable": true, "sortType": "string", "field": "movieCd", "style": "", "editor": { "type": "text", "alignment": "center" } }	Movie code	Movie Title	Movie Title(English)	Year of pr..	Open year	Movie type	Production St..	Production ..	Movie genre	20158982	트릭	Trick	2016	2016/07/13	장편	가봉	한국	스릴러,드라마	20161722	히야	HIYA	2015	2016/03/10	장편	가봉	한국	드라마	20152746	카타	Eclipse	2016	2016/03/30	장편	가봉	한국	범죄,드라마	20154910	터널	Tunnel	2016	2016/08/10	장편	가봉	한국	드라마	20144444	아가씨	The Handmaiden	2016	2016/06/01	장편	가봉	한국	스릴러,드라마	20158482	동주	DONGJU: The Portrait of A Poet	2015	2016/02/17	장편	가봉	한국	드라마	20150979	역적:올주	The Last Princess	2016	2016/08/03	장편	가봉	한국	사극,드라마	20140194	곡성	THE WALLING	2016	2016/05/12	장편	가봉	한국	미스터리,스릴러..
Movie code	Movie Title	Movie Title(English)	Year of pr..	Open year	Movie type	Production St..	Production ..	Movie genre																																																																									
20158982	트릭	Trick	2016	2016/07/13	장편	가봉	한국	스릴러,드라마																																																																									
20161722	히야	HIYA	2015	2016/03/10	장편	가봉	한국	드라마																																																																									
20152746	카타	Eclipse	2016	2016/03/30	장편	가봉	한국	범죄,드라마																																																																									
20154910	터널	Tunnel	2016	2016/08/10	장편	가봉	한국	드라마																																																																									
20144444	아가씨	The Handmaiden	2016	2016/06/01	장편	가봉	한국	스릴러,드라마																																																																									
20158482	동주	DONGJU: The Portrait of A Poet	2015	2016/02/17	장편	가봉	한국	드라마																																																																									
20150979	역적:올주	The Last Princess	2016	2016/08/03	장편	가봉	한국	사극,드라마																																																																									
20140194	곡성	THE WALLING	2016	2016/05/12	장편	가봉	한국	미스터리,스릴러..																																																																									

■ Events

oncellclick, oncelldbclick, oneditchanged, oncolumnheaderclick, onrecordchange, onrecordclick

```
/*-----  
* SCRIPT - use event  
-----*/  
  
function cellclick(event) {  
    $('#info2').html('');  
    $('#info2').html($('#info2').html() +  
    '<font size="4" color="red"> < oncellclick > </font><br></br>' +  
    ' Fires when a specific cell is clicked.<br>' +  
    '<br>[ Index ] ' + event.Index +  
    '<br>[ RecNo ] ' + event.RecNo  
    );  
}  
  
function celldbclick(event) {  
    $('#info2').html($('#info2').html() +  
    '<br><br><font size="4" color="red"> < oncelldbclick > </font><br></br>' +  
    ' Fires when a specific cell is double-clicked.<br>' +  
    '<br>[ column ] ' + JSON.stringify(event.column) +  
    '<br>[ Index ] ' + event.Index +  
    '<br>[ RecNo ] ' + event.RecNo  
    );  
}  
  
function columnheaderclick(event) {  
    $('#info2').html('');  
    $('#info2').html($('#info2').html() +  
    '<font size="4" color="red"> < oncolumnheaderclick > </font><br></br>' +  
    ' Fires when a specific column is clicked in the grid. <br>' +  
    '<br>[ column ] ' + JSON.stringify(event.column) +  
    '<br>[ Index ] ' + event.Index  
    );  
}  
  
function recordclick(event) {  
    $('#info2').html($('#info2').html() +  
    '<br><br><font size="4" color="red"> < onrecordclick > </font><br></br>' +  
    ' Fires when a specific row is clicked in the grid. <br>' +  
    '<br>[ column ] ' + JSON.stringify(event.column) +  
    '<br>[ ColumnIndex ] ' + event.ColumnIndex +  
    '<br>[ RecNo ] ' + event.RecNo  
    );  
}
```

```

}

function recordchange(event) {
    $('#info2').html($('#info2').html() +
        '<br><br><font size="4" color="red"> < onrecordchange > </font><br></br>' +
        '<br> Fires when you click a different row in the grid than before.<br>' +
        '<br>[ column ] ' + JSON.stringify(event.column) +
        '<br>[ ColumnIndex ] ' + event.ColumnIndex +
        '<br>[ RecNo ] ' + event.RecNo
    );
}

function editchanged(event) {
    $('#info2').html($('#info2').html() +
        '<br><br><font size="4" color="red"> < oneditchanged > </font><br></br>' +
        '<br> Fires when the value of a particular cell changes in the grid. <br>' +
        '<br>[ Index ] ' + event.Index +
        '<br>[ RecNo ] ' + event.RecNo +
        '<br>[ oldValue ] ' + event.oldValue +
        '<br>[ newValue ] ' + event.newValue
    );
}

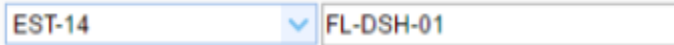
```

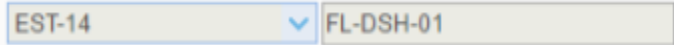
6. Y[wai] Component 예제

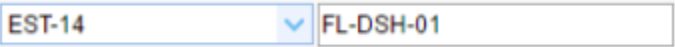
W. CodeCombo

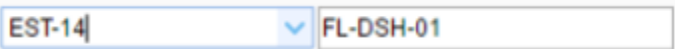
■ Basic Properties

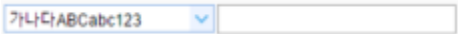
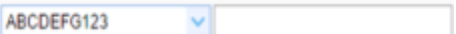
value, disabled, readonly, editable, editType, style, labelWidth, labelStyle, cbDisplayTextField, class, btnImg

property	source
value	combo1.value = 'EST-14';
result	
 원하는 value가 component에 설정됩니다.	

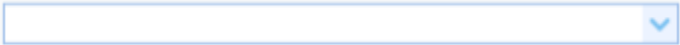
property	source
disabled	combo1.disabled = true;
result	
 component가 작동하지 않습니다.	

property	source
readonly	combo1.readonly = true;
result	
 component가 작동하지 않습니다.	

property	source
editable	combo1.editable = true;
result	
 combobox에 커서를 움직여서 편집합니다. editType 속성을 적용할 수 있습니다.	

property	source	
editType	combo1.editType = 'all'; combo1.editType = 'number';	combo1.editType = 'uppercase'; combo1.editType = 'lowercase';
result		
 한글, 알파벳 대소문자, 숫자 모두 입력 가능  알파벳 대문자와 숫자만 입력 가능  숫자만 입력 가능  알파벳 소문자와 숫자만 입력 가능		

property	source
style	combo1.style = 'width:300px;height:30px';
result	
 component style 속성이 변경됩니다.	

property	source
labelWidth	combo1.labelWidth = '0px'; combo1.labelWidth = '300px';
result	
 label box가 사라집니다. cbDisplayTextField 속성을 적용할 수 있습니다.  label box의 가로 길이를 지정합니다.	

property	source
labelStyle	combo1.labelStyle = 'background-color: #F5EAEA;;';
result	
 label box의 style 속성이 변경됩니다.	

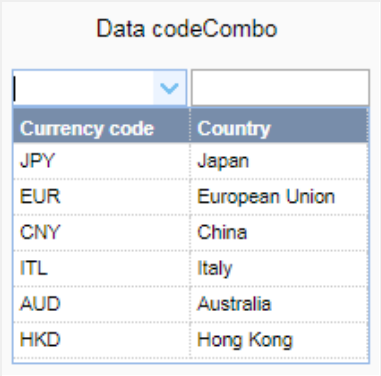
property	source
cbDisplayTextField	EST-10 K9-DL-01 위 그림과 같은 상태에서 labelWidth 속성을 0px로 설정하면 label이 없어지고 box만 표시됩니다. EST-10 해당 속성인 cbDisplayTextField 속성을 적용합니다.
	<pre> combo1.cbDisplayTextField = true; combo1.cbDisplayTextField = false; </pre>
result	
K9-DL-01 label box에 표시된 TextField의 data 값을 표시합니다. EST-10 기본 IdField의 data 값을 표시합니다.	

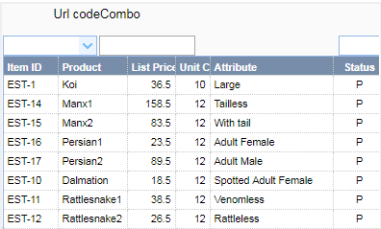
property	source
class	<pre> .classTest1 { background-color: #99cdff; } combo1.class = 'classTest1' </pre>
result	
class를 적용합니다.	

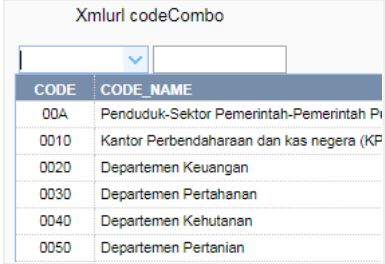
property	source
btnImg	combo1.btnImg = './testData/searchbox_button.png';
result	
 combobox에 있는 combo button을 변경합니다.	

■ Data Properties

data, url, xmurl, idField, textField

property	source
data	<pre> combo2_data.data = [{"CODE": "JPY", "CODE_NM": "일본"}, {"CODE": "EUR", "CODE_NM": "유럽연합"}, {"CODE": "CNY", "CODE_NM": "중국"}, {"CODE": "ITL", "CODE_NM": "이탈리아"}, {"CODE": "AUD", "CODE_NM": "호주"}, {"CODE": "HKD", "CODE_NM": "홍콩"}]; </pre>
result	
	

property	source
url	<pre> combo1.disabled = true; </pre>
result	
	

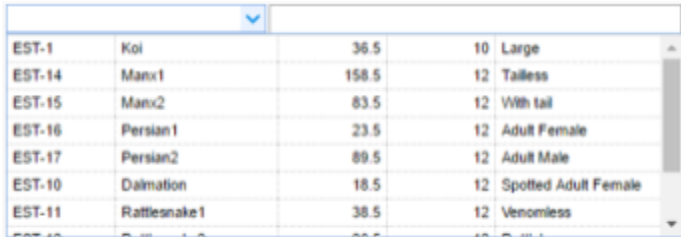
property	source
xmurl	combo2_xmurl.xmlurl= 'testData/F945.xml';
result	
	

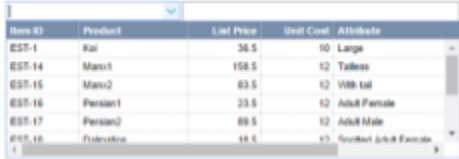
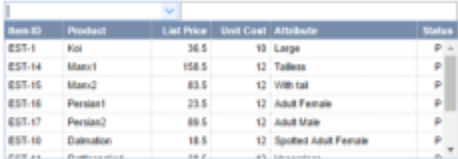
property	source
idField textField	combo2_data.idField = 'CODE_NM'; combo2_data.textField = 'CODE';
result	
	

■ datagrid(panel) Properties

panelWidth, panelHeight, showHeader, fitColumns, columns, dropdownkeycode, itemCount

property	source
panelWidth panelHeight	combo3.panelWidth = 500px; combo3.panelHeight = 100px;
result	
 panel의 가로와 세로 길이가 변경됩니다.	

property	source
showHeader	combo3.showHeader = true; combo3.showHeader = false;
result	
 해당 속성을 false로 설정 시 panel에서 data 표의 header가 사라집니다.	

property	source
fitColumns	<pre> combo3.fitColumns = false; combo3.fitColumns = true; </pre>
result	
 가로 길이를 초과하여 scroll이 생긴 panel에 해당 속성을 true로 설정하면  가로 길이가 조정되면서 scroll이 사라집니다.	

property	source
columns	<pre> combo3.columns =[{field:'itemid',title:'Item ID',width:80,sortable:true}, {field:'productname',title:'Product',width:120}, {field:'status',title:'Status',width:60,align:'center'}, {field:'listprice',title:'List Price',width:80,align:'right'}, {field:'unitcost',title:'Unit Cost',width:80,align:'right'}, {field:'attr1',title:'Attribute',width:200}]; </pre>
result	
 column이 변경됩니다.	

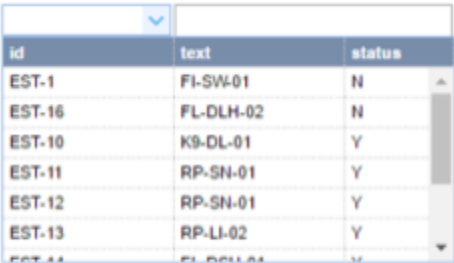
property	source
dropdownkeycode	combo3.dropdownkeycode = 145;
result	
Scroll Lock 키를 누르면 combobox의 panel이 dropdown됩니다.	

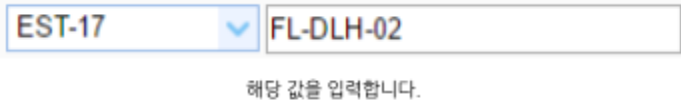
property	source
itemCount	alert("codecombo data count: "+combo3.itemCount);
result	
127.0.0.1 says codecombo data count : 10 OK data의 개수를 알 수 있습니다.	

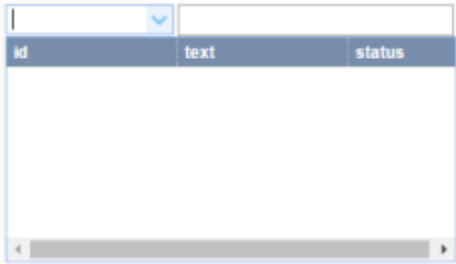
■ Basic Methods

focus, setData, setValue, clear, getValue, getText, getSelected, getData

method	source
focus	combo4.focus();
result	
 component의 combobox로 커서가 이동합니다.	

method	source
setData (json)	<pre> combo4.setData([{"text": "FI-SW-01", "id": "EST-1", "status": "N"}, {"text": "FL-DLH-02", "id": "EST-16", "status": "N"}, {"text": "K9-DL-01", "id": "EST-10", "status": "Y"}, {"text": "RP-SN-01", "id": "EST-11", "status": "Y"}, {"text": "RP-SN-01", "id": "EST-12", "status": "Y"}, {"text": "RP-LI-02", "id": "EST-13", "status": "Y"}, {"text": "FL-DSH-01", "id": "EST-14", "status": "Y"}, {"text": "FL-DSH-01", "id": "EST-15", "status": "Y"}, {"text": "FL-DLH-02", "id": "EST-17", "status": "Y"}, {"text": "AV-CB-01", "id": "EST-18", "status": "Y"}]); </pre>
result	
 component에 data가 세팅되고 data가 panel에 표시됩니다.	

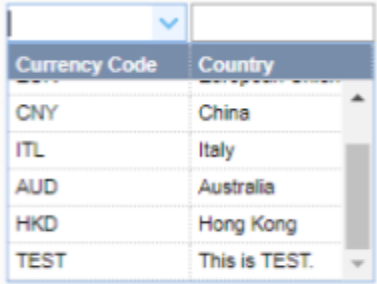
method	source
setValue (value)	combo4.setValue('EST-17');
result	
	

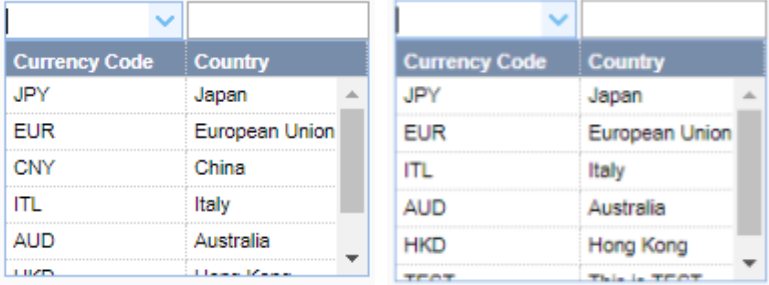
method	source
clear	combo4.clear();
result	
	

method	source
getValue getText getSelected getData	<pre> var val = combo4.getValue(); var text = combo4.getText(); var selVal = combo4.getSelected(); var data = combo4.getData(); alert(".getValue() : " + val + "\n.getText() : " + text + "\n.getSelected() : Displayed in console of developer tools" + "\n.getData() : Displayed in console of developer tools"); console.log(".getValue() : ", val, "\n.getText() : ", text, "\n.getSelected() : ", selVal, "\n.getData() : ", data); </pre>
result	
 Alert 화면  개발자 도구의 Console 화면	

■ Item Methods

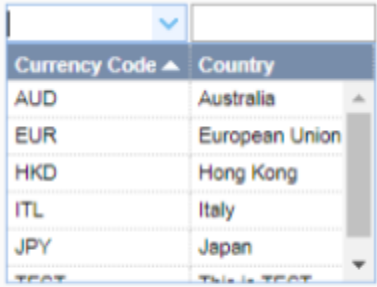
addItem, deleteItem, itemIndex, itemValue, sort

method	source
addItem (id, text)	combo5.addItem('TEST', '테스트');
result	
 data table의 맨 아래쪽 row에 삽입됩니다.	

method	source
deleteItem (index)	combo5.deleteItem(2);
result	
 해당 index의 data가 삭제됩니다.	

method	source
itemIndex (value)	combo5.itemIndex('JPY');
result	
127.0.0.1 says * The current index of an arbitrary value. combo5.itemIndex('JPY'); --> 0 OK	

method	source
itemValue (index, fieldName)	combo5.itemValue(2); combo5.itemValue(4,'CODE_NM');
result	
127.0.0.1 says * The value of index 2 combo5.itemValue(2); --> ITL * A specific field of Value (CODE_NM) with index 4, combo5.itemValue(4,'CODE_NM'); --> Hong Kong OK	

method	source														
sort (fieldName, sortOrder)	combo5.sort('CODE','asc');														
result															
 <table border="1"> <thead> <tr> <th>Currency Code ▲</th><th>Country</th></tr> </thead> <tbody> <tr> <td>AUD</td><td>Australia</td></tr> <tr> <td>EUR</td><td>European Union</td></tr> <tr> <td>HKD</td><td>Hong Kong</td></tr> <tr> <td>ITL</td><td>Italy</td></tr> <tr> <td>JPY</td><td>Japan</td></tr> <tr> <td>TEST</td><td>This is TEST</td></tr> </tbody> </table> data 중에서 code field 값을 기준으로 오름차순 정렬됩니다.		Currency Code ▲	Country	AUD	Australia	EUR	European Union	HKD	Hong Kong	ITL	Italy	JPY	Japan	TEST	This is TEST
Currency Code ▲	Country														
AUD	Australia														
EUR	European Union														
HKD	Hong Kong														
ITL	Italy														
JPY	Japan														
TEST	This is TEST														

■ Column Methods

hideColumn, showColumn

method	source
hideColumn (ColumnNm)	combo6.hideColumn('attr1');
result	
 해당하는 data field의 값을 숨깁니다.	

method	source
showColumn (ColumnNm)	combo6.showColumn('attr1');
result	
 해당하는 data field의 값을 data 표에 표시합니다.	

■ Events

onSelect, onChange, onShowPanel, onHidePanel, onLoadSuccess, onLoadError, onSortColumn, onclick, ondblclick, onblur, onfocus, oncontextmenu, onkeydown, onkeypress, onkeyup, onmouseover, onmousedown, onmouseup

```
/*-----  
* BODY  
-----*/  
  
<codecombo id="combo7"  
style="width:300px"  
value="EST-12"  
labelWidth="180px"  
idField="id"  
textField="text"  
url="testData/datagrid_data2.json"  
fitColumns="true"  
  
onSelect      ="eventAll(event);"  
onChange      ="eventAll(event);"  
onShowPanel   ="eventAll(event);"  
onHidePanel   ="eventAll(event);"  
onLoadSuccess ="eventAll(event);"  
onLoadError   ="eventAll(event);"  
onSortColumn  ="eventAll(event);"  
onclick       ="eventAll(event);"  
ondblclick    ="eventAll(event);"  
onblur        ="eventAll(event);"  
onfocus      ="eventAll(event);"  
oncontextmenu ="eventAll(event);"  
onkeydown     ="eventAll(event);"  
onkeypress    ="eventAll(event);"  
onkeyup       ="eventAll(event);"  
onmouseover   ="eventAll(event);"  
onmousedown   ="eventAll(event);"  
onmouseup     ="eventAll(event);"  
>  
"columns" : [  
    {"field":"id", "title":"key", "width":80, "sortable":true, "align":"left"},  
    {"field":"text", "title":"value", "width":80},  
    {"field":"status", "title":"비고", "width":50}  
]  
</codecombo>  
  
/*-----  
* SCRIPT - use event  
-----*/
```

```

function eventAll(obj) {
if(obj.event == "onSelect"){
    console.log('★----- onSelect Event');
    console.log("index : " + obj.args.index + "⚡nrowData : " , obj.args.rowData);
    console.log("rowData : " +
obj.args.rowData.text,obj.args.rowData.id,obj.args.rowData.status);
    console.log('-----★');
}
else if(obj.event == "onChange"){
    console.log('★----- onChange Event');
    console.log("newValue : " + obj.args.newValue + "⚡noldValue : " + obj.args.oldValue);
    console.log('-----★');
}
else if(obj.event == "onShowPanel"){
    console.log('★----- onShowPanel Event');
}
else if(obj.event == "onHidePanel"){
    console.log('★----- onHidePanel Event');
}

    else if(obj.event == "onLoadSuccess"){
        console.log('★----- onLoadSuccess Event');
    }
else if(obj.event == "onLoadError"){
    console.log('★----- onLoadError Event');
}
else if(obj.event == "onSortColumn"){
    console.log('★----- onSortColumn Event');
    console.log("column : " + obj.args.column + "⚡norder : " + obj.args.order);
    console.log('-----★');
}
else if(obj.event == "onclick"){
    console.log('★----- onclick Event');
}
else if(obj.event == "ondblclick"){
    console.log('★----- ondblclick Event');
}
else if(obj.event == "onblur"){
    console.log('★----- onblur Event');
}
else if(obj.event == "onfocus"){
    console.log('★----- onfocus Event');
}
else if(obj.event == "oncontextmenu"){
    console.log('★----- oncontextmenu Event');
}
else if(obj.event == "onkeydown"){
    console.log('★----- onkeydown Event');
}
else if(obj.event == "onkeypress"){

```

```
        console.log('★----- onkeypress Event');
    }
    else if(obj.event == "onkeyup"){
        console.log('★----- onkeyup Event');
    }
    else if(obj.event == "onmouseover"){
        console.log('★----- onmouseover Event');
    }
    else if(obj.event == "onmousedown"){
        console.log('★----- onmousedown Event');
    }
    else if(obj.event == "onmouseup"){
        console.log('★----- onmouseup Event');
    }
    }
```

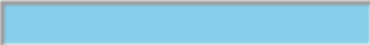
6. Y[wai] Component 예제

X. Edit

■ Common Properties

value, class, style, title, disabled, readonly, maxlength, placeholder

property	source
value	edit1.value="abcde";
result	
 원하는 value를 설정합니다.	

property	source
class	edit1.class="className"; .className{ Background-color:skyblue; }
result	
 원하는 class를 설정합니다.	

property	source
style	edit1.style="width:150px;border:1px solid blue;"
result	
 원하는 style을 설정합니다.	

property	source
title	edit1.title="Hello";
result	
 원하는 title을 설정합니다.	

property	source
disabled	edit1.disabled=true;
result	
 객체를 비활성화합니다.	

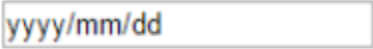
property	source
readonly	edit1.readonly=true;
result	
Read Only 객체를 읽기 전용으로 설정합니다.	

property	source
maxlength	edit1.maxlength=5;
result	
1111 입력 최대 길이를 설정합니다.	

property	source
placeholder	edit1.placeholder="Enter the ID";
result	
Enter the ID 객체에 힌트가 표시됩니다.	

■ Type Properties

date, datetime, currency, decimal, integer, mask, regex

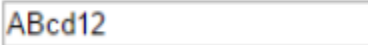
property	source
date	edit1.type="date";
result	
 날짜 입력 edit	

property	source
datetime	edit1.type="datetime";
result	
 시간 입력 edit	

property	source
currency	edit1.type="currency";
result	
 금액 입력 edit	

property	source
decimal	edit1.type="decimal";
result	
 소수 입력 edit	

property	source
integer	edit1.type="integer";
result	
 정수 입력 edit	

property	source
mask	edit1.type="mask";
result	
 마스킹 형태 edit	

property	source
regex	edit1.type="regex";
result	
010-0012-9952 정규 표현식 형태 edit	

■ Sub Properties

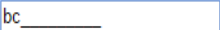
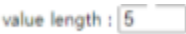
date, currency, decimal, mask, regex

property	source
date (dateOrder)	<edit id="dateOrder1" type="date" dateSeparator="ddmmyyyy"></edit>
date (dateSeparator)	<edit id="dateSeparator1" type="date" dateSeparator="-"></edit>
result	
2023/06/09 날짜 표시 순서 yyyymmdd (default) 09/06/2023 날짜 표시 순서 ddmmyyyy 06/09/2023 날짜 표시 순서 mmddyyyy 2024/05/01 날짜 구분자 / (default)	

property	source
currency (prefix)	<edit id="prefix1" type="currency" prefix="₩"></edit>
currency (digits)	<edit id="digits1" type="currency" digits="3"></edit>
currency (roundType)	<edit id="round1" type="currency" roundType="round"></edit>
result	
\$4,835.00 ₩4,835.00 통화 기호 (default: 통화 기호 없음) 279.00 279.00000 소수점 이하 자릿수 (default: 2) 45.00 소수점 이하 반올림 (default: round)	

property	source
decimal (digits)	<edit id="digits2" type="decimal" digits="3"></edit>
decimal (roundType)	<edit id="round2" type="currency" roundType="round"></edit>
result	
6,578,746 6578746 소수점 이하 자릿수 (default: 소수점 이하 20자리) 45.00 소수점 이하 반올림 (default: round)	

property	source
mask (mask)	<edit id="ma" type="mask" mask="AA99[***]"></edit>
mask (defaultValue)	<edit id="defVal" type="mask" mask="*****" defaultValue="a"></edit>
mask (dataLength)	<edit id="defVal" type="mask" dataLength="5" value="AAA99"></edit>
mask (displayMask)	<edit id="displayMask1" type="mask" mask="9999/99/99" displayMask="false"></edit>

result			
	 	 	
마스킹 형태	일부의 값만 입력한 후 객체에서 blur되면 빈 마스킹 부분을 defaultValue로 채움	value 속성의 value 길이 표시 (읽기 전용 속성)	마우스 오버 시 마스킹 미리 보기 설정

property	source
regex (regex)	<edit id="re" type="regex" regex="[1-9]{1}[0-9]{2}-[0-9]{3}"></edit>
result	
	

■ Methods

focus, unmaskedValue, getValue

method	source
focus	edit1.focus();
result	
 커서가 객체로 이동합니다.	

method	source
unmaskedValue	edit1.unmaskedValue();
result	
마스킹을 제외한 value를 가져옵니다. 2012/12/31 → 20121231 121,200.00 → 121200 ABC_ → ABC	

method	source
getValue	edit1.getValue();
result	
마스킹을 포함한 value를 가져옵니다. 2012/12/31 → 2012/12/31 121,200.00 → 121,200,00 ABC_ → ABC_	

■ Events

onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmouseup, onmouseover, onblur, onfocus, onchange, oncontextmenu, oncomplete, onincomplete, oncleared, onZeroMskComplete

```
/* -----
* body source
* ----- */
onclick      : <edit id="clikevt1"  onclick="eventAll(event);"></edit>
ondblclick   : <edit id="dblclikevt1" ondblclick="eventAll(event);"></edit>
onkeydown    : <edit id="keydownvt1" onkeydown="eventAll(event);"></edit>
onkeypress   : <edit id="keypressevt1" onkeypress="eventAll(event);"></edit>
onkeyup      : <edit id="keyupevt1"  onkeyup="eventAll(event);"></edit>
onmousedown  : <edit id="mousedownvt1" onmousedown="eventAll(event);"></edit>
onmouseup    : <edit id="mouseupevt1" onmouseup="eventAll(event);"></edit>
onmouseover  : <edit id="mouseovervt1" onmouseover="eventAll(event);"></edit>
onblur       : <edit id="blurevt1"   onblur="eventAll(event);"></edit>
onfocus     : <edit id="focusevt1"  onfocus="eventAll(event);"></edit>
onchange     : <edit id="changevt1"  onchange="eventAll(event);"></edit>
oncontextmenu : <edit id="ctmenuvt1"  oncontextmenu="eventAll(event);"></edit>

oncomplete   : <edit id="oncompleteevt1" type="date" oncomplete="eventAll(event);"></edit>
onincomplete : <edit id="onincompleteevt1" type="date"
onincomplete="eventAll(event);"></edit>
oncleared    : <edit id="oncleared1evt" type="date" oncleared="eventAll(event);"></edit>

onZeroMskComplete : <edit id="zmevt" type="mask" mask="999999" defaultValue="0"
onZeroMskComplete="eventAll(event);"></edit>

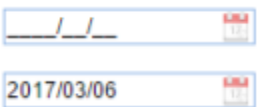
/* -----
* script source
* ----- */
function eventAll(obj){
    if (obj.event == 'onChange') {
        alert("Event : " + obj.event + "WnoldValue: " + obj.args.oldValue + "WnnewValue: " +
obj.args.newValue);
    } else {
        alert("Event : " + obj.event);
    }
}
```

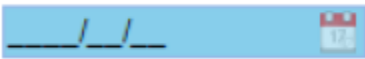
6. Y[wai] Component 예제

Y. DateEdit

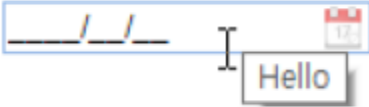
■ Properties

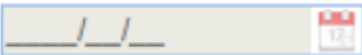
value, class, style, title, disabled, readonly, lang, dateOrder, validate, calWidth, calHeight, btnImg, firstDay, dateSeparator

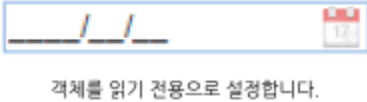
property	source
value	dateedit1.value="abcde"
result	
 value=" " value="today" 원하는 value를 설정합니다.	

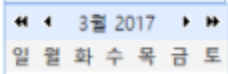
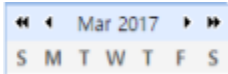
property	source
class	dateedit1.class="className"; .className{ Background-color:skyblue; }
result	
 원하는 class를 설정합니다.	

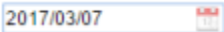
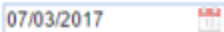
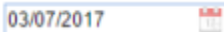
property	source
style	dateedit1.style="width:150px;height:30px;font-size:14px;border:1px solid red;"
result	
 원하는 style을 설정합니다.	

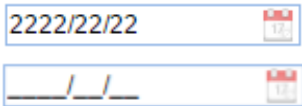
property	source
title	dateedit1.title="Hello";
result	
 원하는 title을 설정합니다.	

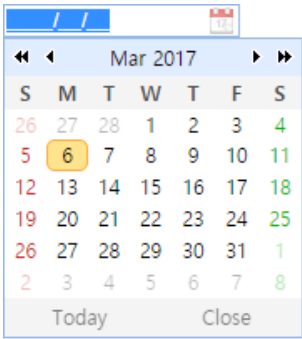
property	source
disabled	dateedit1.disabled=true;
result	
 객체를 비활성화합니다.	

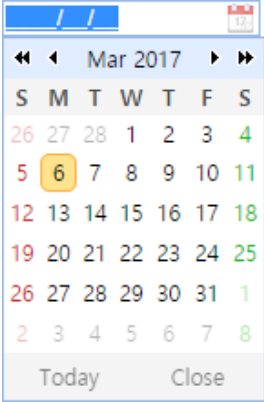
property	source
readonly	dateedit1.readonly=true;
result	
	

property	source
lang	dateedit1.lang="Korean"; dateedit1.lang="English"; dateedit1.lang="Chinese";
result	
달력 상단 표시 언어를 설정합니다.  Korean  English  Chinese	

property	source
dateOrder	dateedit1.dateOrder="yyyymmdd"; dateedit1.dateOrder="ddmmyyyy"; dateedit1.dateOrder="mmddyyyy";
result	
 yyyymmdd  ddmmyyyy  mmddyyyy	

property	source
validate	dateedit1.validate=true;
result	
 잘못된 data를 입력할 경우 blur를 하여 값을 지웁니다.	

property	source
calWidth	dateedit1.calWidth="180px";
result	
	

property	source
calHeight	dateedit1.calHeight="200px";
result	
	

property	source
btnImg	dateedit1.btnImg="이미지경로";
result	
	

■ Methods

focus, unmaskedValue, getValue, setValue

method	source
focus	<code>dateedit1.focus();</code>
result	
 커서가 객체로 이동합니다.	

method	source
unmaskedValue	<code>dateedit1.unmaskedValue();</code>
result	
 → 20170306 마스킹을 제외한 value를 가져옵니다.	

method	source
getValue	<code>dateedit1.getValue();</code>
result	
 → 2017/03/06 마스킹을 포함한 value를 가져옵니다.	

method	source
setValue	dateedit1.setValue(value);
result	
2016/06/10 	

■ Events

onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmouseup, onmouseover, onblur, onfocus, oncontextmenu, onSelect, onChange

```
onclick    :<dateedit id="clিকেব1"    onclick="eventAll(event);"></dateedit>
ondblclick :<dateedit id="dblclিকেব1" ondblclick="eventAll(event);"></dateedit>
onkeydown  :<dateedit id="keydownেব1" onkeydown="eventAll(event);"></dateedit>
onkeypress :<dateedit id="keypresেব1" onkeypress="eventAll(event);"></dateedit>
onkeyup    :<dateedit id="keyuেব1"    onkeyup="eventAll(event);"></dateedit>
onmousedown :<dateedit id="mousedownেব1" onmousedown="eventAll(event);"></dateedit>
onmouseup  :<dateedit id="mouseuেব1"  onmouseup="eventAll(event);"></dateedit>
onmouseover :<dateedit id="mouseoverেব1" onmouseover="eventAll(event);"></dateedit>
onblur     :<dateedit id="blureব1"    onblur="eventAll(event);"></dateedit>
onfocus   :<dateedit id="focuseব1"   onfocus="eventAll(event);"></dateedit>
oncontextmenu :<dateedit id="ctmenueব1" oncontextmenu="eventAll(event);"></dateedit>
onSelect   :<dateedit id="selectেব1"  onSelect="eventAll(event);"></dateedit>
onChange   :<dateedit id="changeেব1"  onChange="eventAll(event);"></dateedit>

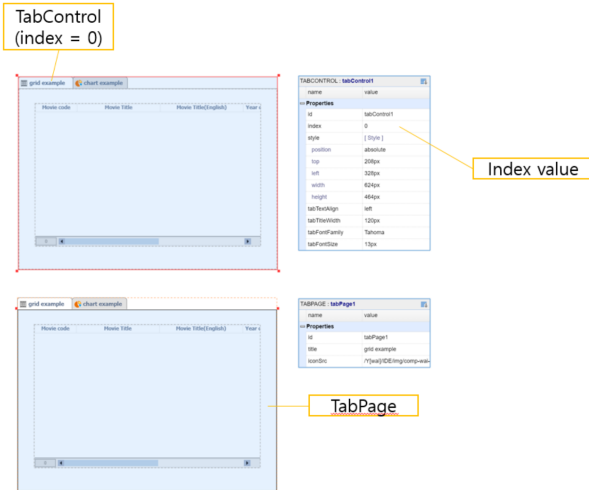
/* -----
* script source
* ----- */

function eventAll(obj){
  if (obj.event == 'onSelect') {
    alert(obj.event + " : " + obj.srcElement.value);
  } else if (obj.event == 'onChange') {
    alert(obj.event + "WnoldDate: " + obj.args.oldDate + "WnnewDate: " + obj.args.newDate);
  } else {
    alert(obj.event);
  }
}
```


6. Y[wai] Component 예제

Z. TabControl

■ Properties

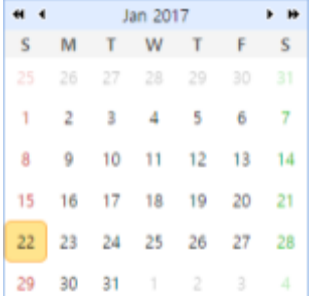
property	source
id	<pre><tabControl id="tabControl1" index="0" tabTextAlign="left" tabTitleWidth="120px" tabFontFamily="Tahoma" tabFontSize="13px" onLoaded="tabControl1Loaded(event);" style="position: absolute; left: 40px; top: 56px; width: 624px; height: 464px;"> <tabPage id="tabPage0" title="chart example" iconSrc="/Y[wai]/IDE/windows/editors/img/icon-pie.gif" style="width: 100%; height: 100%; display: block;"> </tabPage> <tabPage id="tabPage1" title="newTabPage" style="width: 100%; height: 100%; display: none;"> </tabPage> </tabControl></pre>
index	
tabTextAlign	
tabTitleWidth	
tabFontFamily	
tabFontSize	
[tabPage] id	
[tabPage] title	
[tabPage] iconSrc	
result	
	

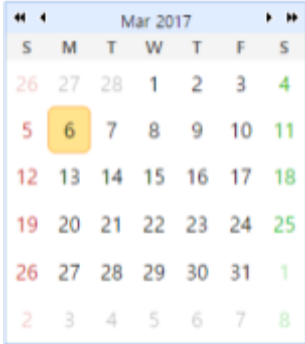
6. Y[wai] Component 예제

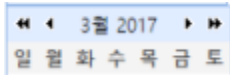
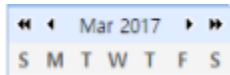
AA. Calendar

■ Properties

value, style, lang, dateOrder, firstDay

property	source
value	calendar1.value="20170122";
result	
 원하는 value를 설정합니다.	

property	source
style	calendar1.style="width:220px;height:250px;font-size:15px;"
result	
 원하는 style을 설정합니다.	

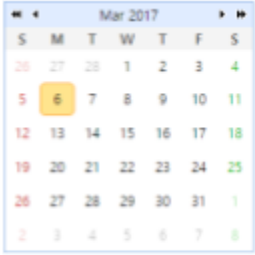
property	source
lang	calendar1.lang="Korean"; calendar1.lang="English"; calendar1.lang="Chinese";
result	
달력 상단 표시 언어를 설정합니다.    Korean English Chinese	

property	source
dateOrder	calendar1.dateOrder="yyymmdd"; calendar1.dateOrder="ddmmyyyy"; calendar1.dateOrder="mmdyyy";
result	
dateOrder 속성에 따른 value 값 (yyymmdd: 20170306) (ddmmyyyy: 06032017) (mmdyyy: 03062017)	

property	source
firstDay	calendar1.firstDay="sun"; calendar1.firstDay="mon"; calendar1.firstDay="tue"; calendar1.firstDay="wed"; calendar1.firstDay="thu"; calendar1.firstDay="fri"; calendar1.firstDay="sat";
result	
달력의 한 주 시작 요일을 설정합니다. ◀◀ Mar 2017 ▶▶ S M T W T F S sunday ◀◀ Mar 2017 ▶▶ T F S S M T W thursday ◀◀ Mar 2017 ▶▶ M T W T F S S monday ◀◀ Mar 2017 ▶▶ F S S M T W T friday ◀◀ Mar 2017 ▶▶ T W T F S S M tuesday ◀◀ Mar 2017 ▶▶ S S M T W T F saturday ◀◀ Mar 2017 ▶▶ W T F S S M T wednesday	

■ Methods

getValue, setValue

method	source
getValue	calendar1.getValue();
result	
 마스킹을 포함한 value를 가져옵니다. (yyyymmdd: 2017/03/06) (ddmmyyyy: 06/03/2017) (mmdyyyy: 03/06/2017)	

method	source
setValue	calendar1.setValue(value);
result	
dateOrder 속성값에 따라 날짜값을 세팅합니다. ID.setValue(); 또는 ID.setValue(''); 와 같이 값을 주면 오늘 날짜로 설정됩니다.	

■ Events

onSelect

```
/*-----  
* BODY  
-----*/  
<calendar id="calEvent"  
onSelect ="eventAll(event);"  
>  
</ calendar >  
  
/*-----  
* SCRIPT - use event  
-----*/  
function eventAll(obj) {  
    alert("Event : " + obj.event);  
    console.log("Today : ", selEvt.year, selEvt.month, selEvt.day);  
    console.log("Select : ", selEvt.selYear, selEvt.selMonth, selEvet.selDay);  
}
```

6. Y[wai] Component 예제

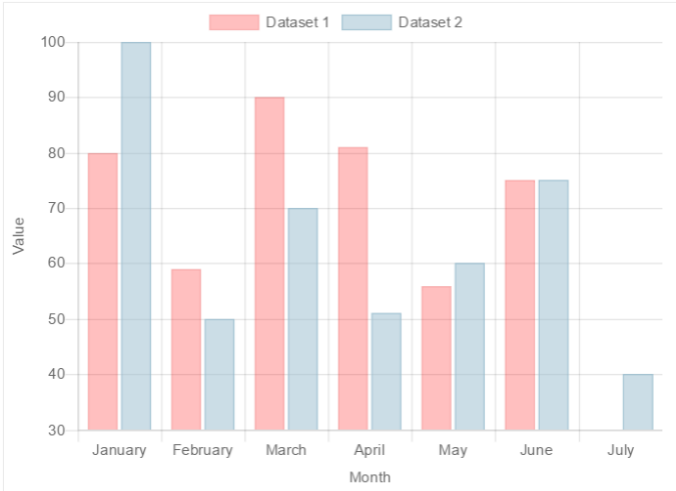
BB. Chart

■ Properties

type, data, option, style

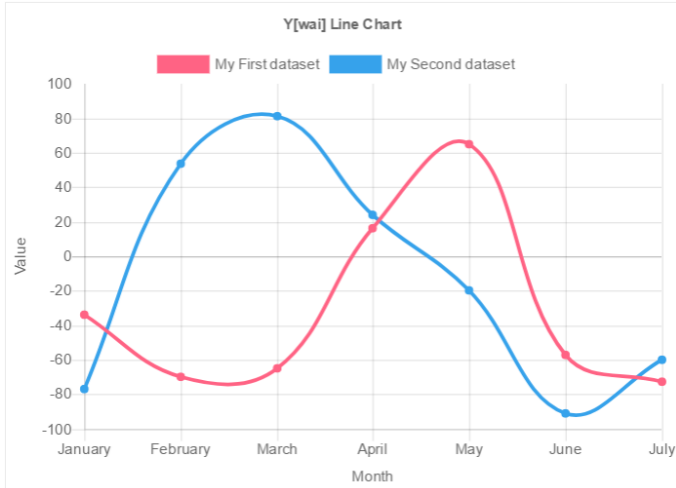
✓ Bar

property	source
type	type="bar"
data	<pre>"data" : { "labels": ["January", "February", "March", "April", "May", "June", "July"], "datasets": [{ "label": "Dataset 1 ", "backgroundColor": "rgba(255, 0, 0, 0.25)", "borderColor": "rgba(220, 0, 0, 0.2)", "borderWidth": 1, "hoverBackgroundColor":"rgba(220, 0, 0, 0.5)", "hoverBorderColor":"rgba(220,0, 0, 0.5)", "hoverBorderWidth":1, "data": [80,59,90,81,56,75,30] }, { "label": "Dataset 2 ", "backgroundColor": "rgba(151,187,205,0.5)", "borderColor": "rgba(151,187,205,0.8)", "borderWidth": 1, "hoverBackgroundColor":"rgba(151,187,205,0.8)", "hoverBorderColor":"rgba(151,187,205,1)", "hoverBorderWidth":1, "data": [100,50,70,51,60,75,40] }], },</pre>
option	<pre>"option": { "responsive": true, "tooltips": { "mode": "index", "intersect": false }, "hover": { "mode": "nearest",</pre>

	<pre> "intersect": true }, "scales": { "xAxes": [{ "display": true, "scaleLabel": { "display": true, "labelString": "Month" } }], "yAxes": [{ "display": true, "scaleLabel": { "display": true, "labelString": "Value" } }] } } }</pre>																								
style	style="position: absolute; top: 47px; left: 21px; width: 300px; height: 300px;"																								
result																									
 <table><tr><th>Month</th><th>Dataset 1</th><th>Dataset 2</th></tr><tr><td>January</td><td>80</td><td>100</td></tr><tr><td>February</td><td>59</td><td>50</td></tr><tr><td>March</td><td>90</td><td>70</td></tr><tr><td>April</td><td>81</td><td>51</td></tr><tr><td>May</td><td>56</td><td>60</td></tr><tr><td>June</td><td>75</td><td>75</td></tr><tr><td>July</td><td>0</td><td>40</td></tr></table>		Month	Dataset 1	Dataset 2	January	80	100	February	59	50	March	90	70	April	81	51	May	56	60	June	75	75	July	0	40
Month	Dataset 1	Dataset 2																							
January	80	100																							
February	59	50																							
March	90	70																							
April	81	51																							
May	56	60																							
June	75	75																							
July	0	40																							

✓ Line

property	source
type	type="line"
data	<pre> "data" : { "labels": ["January", "February", "March", "April", "May", "June", "July"], "datasets": [{ "label": "My First dataset", "backgroundColor": "rgb(255, 99, 132)", "borderColor": "rgb(255, 99, 132)", "data": [-34,-70,-65,16,65,-57,-73], "fill": false }, { "label": "My Second dataset", "fill": false, "backgroundColor": "rgb(54, 162, 235)", "borderColor": "rgb(54, 162, 235)", "data": [-77,54,81,24,-20,-91,-60] }] }, </pre>
option	<pre> "option": { "responsive": true, "title": { "display": true, "text": "Y[wai] Line Chart", "position": "top" }, "tooltips": { "mode": "index", "intersect": false }, "hover": { "mode": "nearest", "intersect": true }, "scales": { "xAxes": [{ "display": true, "scaleLabel": { "display": true, "labelString": "Month" } }] }, </pre>

	<pre>"yAxes": [{ "display": true, "scaleLabel": { "display": true, "labelString": "Value" } }] }</pre>																								
style	style = "position: absolute; top: 47px; left: 21px; -index: 1000; width: 300px; height: 300px;"																								
result																									
 The chart displays two sinusoidal waves. The blue line, 'My Second dataset', starts at -80 in January, peaks at 80 in March, crosses zero in April, reaches a trough at -90 in June, and ends at -60 in July. The red line, 'My First dataset', starts at -30 in January, reaches a trough at -70 in February, crosses zero in April, peaks at 65 in May, reaches a trough at -60 in June, and ends at -70 in July. <table><tr><th>Month</th><th>My First dataset</th><th>My Second dataset</th></tr><tr><td>January</td><td>-30</td><td>-80</td></tr><tr><td>February</td><td>-70</td><td>55</td></tr><tr><td>March</td><td>-65</td><td>80</td></tr><tr><td>April</td><td>15</td><td>20</td></tr><tr><td>May</td><td>65</td><td>-20</td></tr><tr><td>June</td><td>-60</td><td>-90</td></tr><tr><td>July</td><td>-70</td><td>-60</td></tr></table>		Month	My First dataset	My Second dataset	January	-30	-80	February	-70	55	March	-65	80	April	15	20	May	65	-20	June	-60	-90	July	-70	-60
Month	My First dataset	My Second dataset																							
January	-30	-80																							
February	-70	55																							
March	-65	80																							
April	15	20																							
May	65	-20																							
June	-60	-90																							
July	-70	-60																							

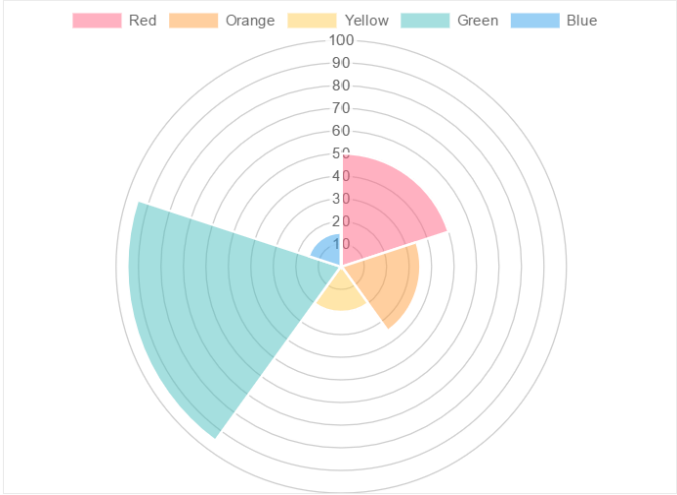
✓ Radar

property	source
type	type="radar"
data	<pre> "data" : { "labels": [["Eating", "Dinner"], ["Drinking", "Water"], "Sleeping", ["Designing", "Graphics"], "Coding", "Cycling", "Running"], "datasets": [{ "label": "My First dataset", "backgroundColor": "rgba(255, 99, 132, 0.1)", "borderColor": "rgba(255, 99, 132, 0.5)", "pointBackgroundColor": "rgba(255, 99, 132, 0.1)", "data": [65,59,null,81,56,55,40], "fill": false },{ "label": "My Second dataset", "backgroundColor": "rgba(54, 162, 235, 0.1)", "borderColor": "rgba(54, 162, 235, 0.5)", "pointBackgroundColor": "rgba(54, 162, 235, 0.1)", "data": [28,48,40,19,96,27,100] }] } </pre>
option	<pre> "option" : { "title" : { "display":false, "position":"top" }, "legend" : { "display":true, "reverse":false, "position":"top", "labels":{ "usePointStyle":false } }, "scale":{ "display" : true, "gridLines" : { "display" : true }, "pointLabels" : { "display" : true }, "ticks" : { "display" : true } }, "tooltips" : { "enabled" : true, "mode" : "nearest", "intersect" : true, "displayColors" : true } } </pre>

	<pre>}, "animation":{ "animateScale" : false } }</pre>																								
style	style= “position: absolute; top: 47px; left: 21px; z-index: 1000; width: 300px; height: 300px;”																								
result																									
The radar chart displays two datasets across seven categories. The scale ranges from 0 to 100 in increments of 10. The legend indicates that the red line represents 'My First dataset' and the blue line represents 'My Second dataset'. <table><tr><th>Activity</th><th>My First dataset</th><th>My Second dataset</th></tr><tr><td>Eating Dinner</td><td>60</td><td>40</td></tr><tr><td>Drinking Water</td><td>40</td><td>30</td></tr><tr><td>Sleeping</td><td>20</td><td>30</td></tr><tr><td>Designing Graphics</td><td>80</td><td>30</td></tr><tr><td>Coding</td><td>40</td><td>100</td></tr><tr><td>Cycling</td><td>20</td><td>30</td></tr><tr><td>Running</td><td>30</td><td>100</td></tr></table>		Activity	My First dataset	My Second dataset	Eating Dinner	60	40	Drinking Water	40	30	Sleeping	20	30	Designing Graphics	80	30	Coding	40	100	Cycling	20	30	Running	30	100
Activity	My First dataset	My Second dataset																							
Eating Dinner	60	40																							
Drinking Water	40	30																							
Sleeping	20	30																							
Designing Graphics	80	30																							
Coding	40	100																							
Cycling	20	30																							
Running	30	100																							

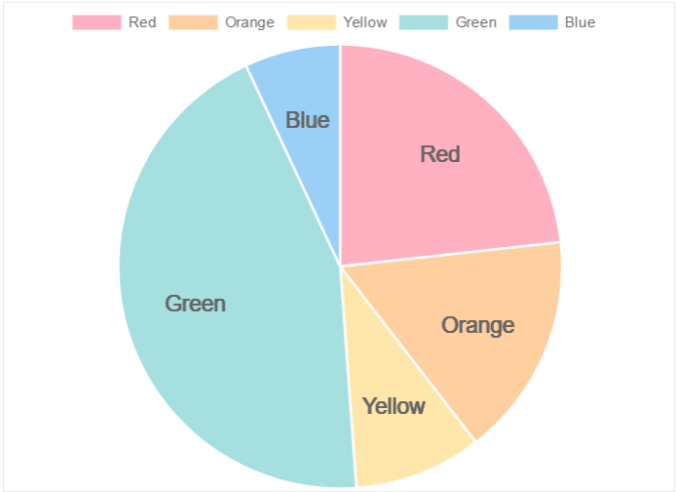
✓ Polar Area

property	source
type	type="polarArea"
data	<pre> "data" : { "datasets": [{ "data": [50,35,20,95,15], "backgroundColor": ["rgba(255, 99, 132, 0.5)", "rgba(255, 159, 64, 0.5)", "rgba(255, 205, 86, 0.5)", "rgba(75, 192, 192, 0.5)", "rgba(54, 162, 235, 0.5)"], "hoverBackgroundColor": ["rgba(255, 99, 132, 1)", "rgba(255, 159, 64, 1)", "rgba(255, 205, 86, 1)", "rgba(75, 192, 192, 1)", "rgba(54, 162, 235, 1)"], "label": "My dataset" }], "labels": ["Red", "Orange", "Yellow", "Green", "Blue"] } </pre>
option	<pre> "option" : { "title" : { "display":false, "position":"top" }, "legend" : { "display":true, "reverse":false, "position":"top", "labels":{ "usePointStyle":false } }, "scale":{ "display" : true, </pre>

	<pre>"gridLines" : { "display" : true }, "pointLabels" : { "display" : true }, "ticks" : { "display" : true } }, "tooltips" : { "enabled" : true, "mode" : "nearest", "intersect" : true, "displayColors" : true }, "animation":{ "animateScale" : false } }</pre>
style	style = "position: absolute; top: 47px; left: 21px; z-index: 1000; width: 300px;height: 300px;"
result	
	

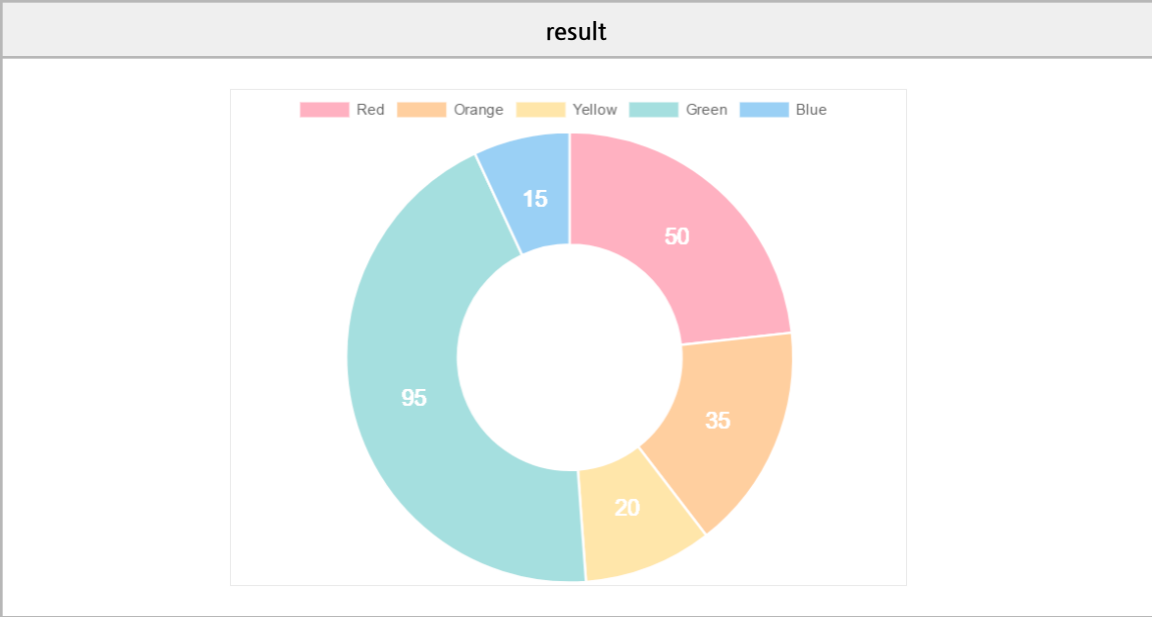
✓ Pie

property	source
type	type="pie"
data	<pre> "data" : { "datasets": [{ "data": [50,35,20,95,15], "backgroundColor": ["rgba(255, 99, 132, 0.5)", "rgba(255, 159, 64, 0.5)", "rgba(255, 205, 86, 0.5)", "rgba(75, 192, 192, 0.5)", "rgba(54, 162, 235, 0.5)"], "hoverBackgroundColor": ["rgba(255, 99, 132, 1)", "rgba(255, 159, 64, 1)", "rgba(255, 205, 86, 1)", "rgba(75, 192, 192, 1)", "rgba(54, 162, 235, 1)"], "label": "Dataset 1" }], "labels": ["Red", "Orange", "Yellow", "Green", "Blue"] } </pre>
option	<pre> "option" : { "tooltips": { "enabled" : false }, "showPieceLabel":{ "enabled": true, "mode": "label", "fontSize": 18, "fontStyle": "bold", "fontColor": "#666", "fontFamily": "'Helvetica Neue', 'Helvetica', 'Arial', sans-serif" } } </pre>

style	style = "position: absolute; top: 47px; left: 21px; z-index: 1000; width: 300px; height: 300px;"
result	
	

✓ Doughnut

property	source
type	type="doughnut"
data	<pre> "data" : { "datasets": [{ "data": [50,35,20,95,15], "backgroundColor": ["rgba(255, 99, 132, 0.5)", "rgba(255, 159, 64, 0.5)", "rgba(255, 205, 86, 0.5)", "rgba(75, 192, 192, 0.5)", "rgba(54, 162, 235, 0.5)"], "label": "Dataset 1" }], "labels": ["Red", "Orange", "Yellow", "Green", "Blue"] } </pre>
option	<pre> "option" : { "tooltips": { "enabled" : false }, "showValue":true, "showPieceLabel":{ "enabled": true } } </pre>
style	<pre> style = " position: absolute; top: 47px; left: 21px; z-index: 1000; width: 300px; height: 300px;" </pre>



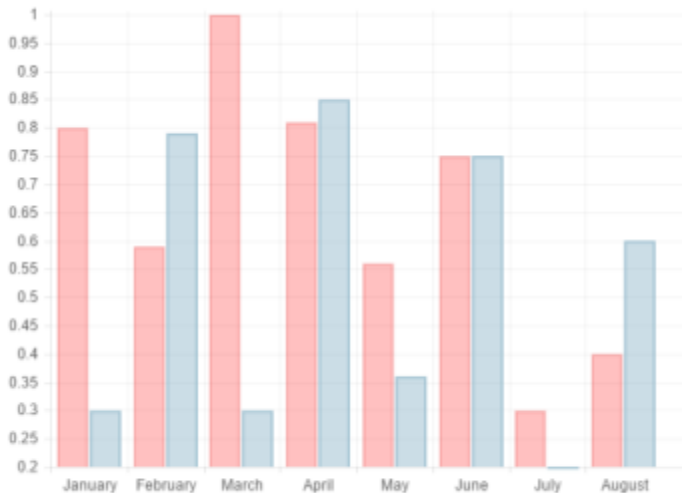
■ Methods

getData, update, addData, removeData, setData

✓ Bar

method	source
getData	<pre> function bar_getData(){ var returnValue = bar.getData(); alert('It takes an entire data array.' + '\n\n Check getData object in the console window.' + '\n\n' + 'ex) Use the return value to extract the desired value.' + '\n\n' + 'returnValue[0].value[0] : ' + returnValue[0].value[0]); console.log('bar_getData : ' , returnValue); } </pre>
result	
127.0.0.1 says It takes an entire data array. Check the getData object in the console window. ex) Use the return value to extract the desired value. returnValue[0].data[0] : 80 OK The screenshot shows a web browser window with a yellow alert box in the foreground. The alert box contains the text: "127.0.0.1 says", "It takes an entire data array.", "Check the getData object in the console window.", "ex) Use the return value to extract the desired value.", and "returnValue[0].data[0] : 80". There is an "OK" button at the bottom right of the alert box. In the background, the browser's developer console is open, showing the output of the bar_getData function. The console output is a JSON object with two elements. The first element is an array of two objects. The first object has properties: backgroundColor: "rgba(255, 0, 0, 0.25)", borderColor: "rgba(220, 0, 0, 0.2)", borderWidth: 1, data: Array(7) [0: 80, 1: 59, 2: 90, 3: 81, 4: 56, 5: 75, 6: 30], length: 7, pop: f(), push: f(), shift: f(), splice: f(), unshift: f(), __chartjs: {listeners: Array(1)}, __proto__: Array(0). The second object has properties: backgroundColor: "rgba(151,187,205,0.5)", borderColor: "rgba(151,187,205,0.8)", and borderWidth: 1.	

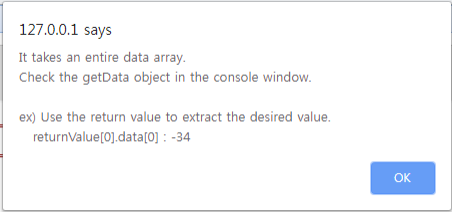
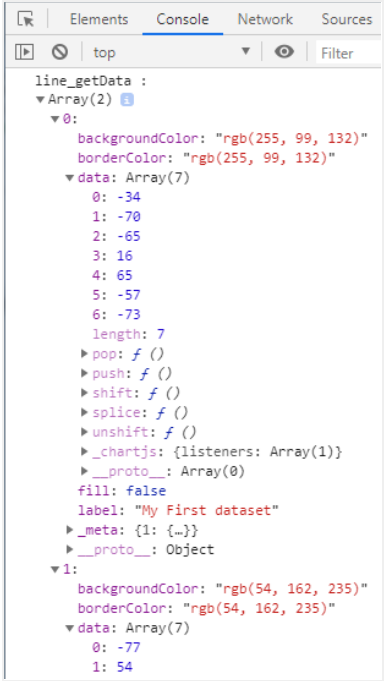
method	source																								
update (value,dataset,index)	<pre>function bar_update(){ bar.update(6,0,2); bar.update(8,1,2); }</pre>																								
result																									
<table><tr><th>Month</th><th>Dataset 1</th><th>Dataset 2</th></tr><tr><td>January</td><td>80</td><td>100</td></tr><tr><td>February</td><td>58</td><td>50</td></tr><tr><td>March</td><td>5</td><td>8</td></tr><tr><td>April</td><td>82</td><td>50</td></tr><tr><td>May</td><td>55</td><td>60</td></tr><tr><td>June</td><td>75</td><td>75</td></tr><tr><td>July</td><td>30</td><td>40</td></tr></table> index 2인 March data의 값이 update됩니다.		Month	Dataset 1	Dataset 2	January	80	100	February	58	50	March	5	8	April	82	50	May	55	60	June	75	75	July	30	40
Month	Dataset 1	Dataset 2																							
January	80	100																							
February	58	50																							
March	5	8																							
April	82	50																							
May	55	60																							
June	75	75																							
July	30	40																							

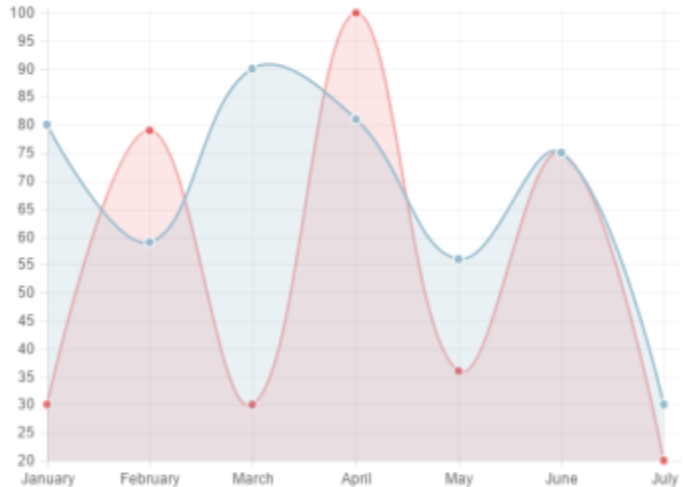
method	source																											
addData (data, label)	<pre>function bar_addData(){ bar.addData([40, 60], "August"); }</pre>																											
result																												
 새로운 data가 추가됩니다. <table><tr><th>Month</th><th>Red Series</th><th>Blue Series</th></tr><tr><td>January</td><td>0.8</td><td>0.3</td></tr><tr><td>February</td><td>0.6</td><td>0.8</td></tr><tr><td>March</td><td>1.0</td><td>0.3</td></tr><tr><td>April</td><td>0.8</td><td>0.85</td></tr><tr><td>May</td><td>0.55</td><td>0.35</td></tr><tr><td>June</td><td>0.75</td><td>0.75</td></tr><tr><td>July</td><td>0.3</td><td>0.0</td></tr><tr><td>August</td><td>0.4</td><td>0.6</td></tr></table>		Month	Red Series	Blue Series	January	0.8	0.3	February	0.6	0.8	March	1.0	0.3	April	0.8	0.85	May	0.55	0.35	June	0.75	0.75	July	0.3	0.0	August	0.4	0.6
Month	Red Series	Blue Series																										
January	0.8	0.3																										
February	0.6	0.8																										
March	1.0	0.3																										
April	0.8	0.85																										
May	0.55	0.35																										
June	0.75	0.75																										
July	0.3	0.0																										
August	0.4	0.6																										

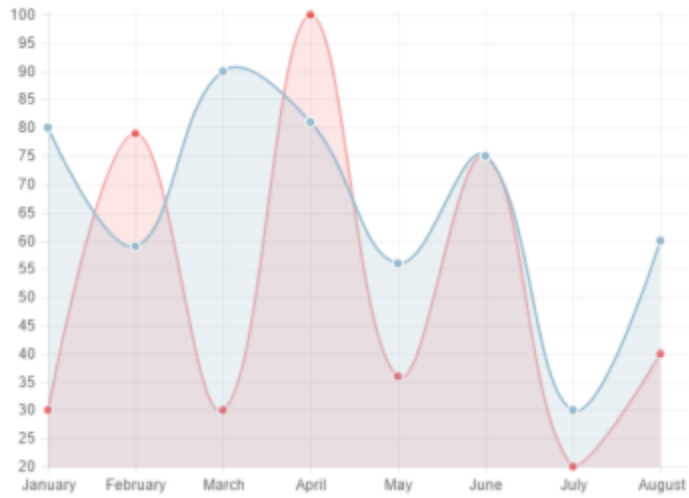
method	source																								
removeData (index)	<pre>function bar_removeData(){ bar.removeData(); }</pre>																								
result																									
<table><tr><th>Month</th><th>Red Series</th><th>Blue Series</th></tr><tr><td>February</td><td>0.58</td><td>0.78</td></tr><tr><td>March</td><td>1.0</td><td>0.3</td></tr><tr><td>April</td><td>0.81</td><td>0.85</td></tr><tr><td>May</td><td>0.55</td><td>0.35</td></tr><tr><td>June</td><td>0.75</td><td>0.75</td></tr><tr><td>July</td><td>0.3</td><td>0.0</td></tr><tr><td>August</td><td>0.4</td><td>0.6</td></tr></table> 첫 번째 data인 January bar data가 삭제됩니다.		Month	Red Series	Blue Series	February	0.58	0.78	March	1.0	0.3	April	0.81	0.85	May	0.55	0.35	June	0.75	0.75	July	0.3	0.0	August	0.4	0.6
Month	Red Series	Blue Series																							
February	0.58	0.78																							
March	1.0	0.3																							
April	0.81	0.85																							
May	0.55	0.35																							
June	0.75	0.75																							
July	0.3	0.0																							
August	0.4	0.6																							

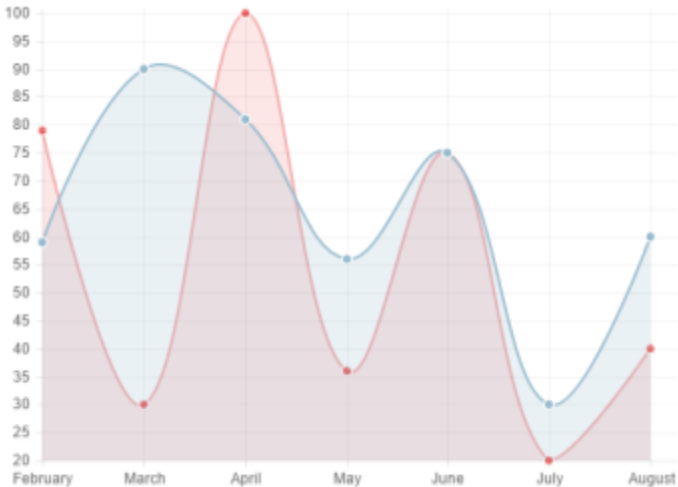
method	source																								
setData (data, option)	<pre>function bar_setData(){ bar.setData(data.barData,option.barOption); }</pre>																								
result																									
<table><tr><th>Month</th><th>Red Series</th><th>Blue Series</th></tr><tr><td>January</td><td>1</td><td>5</td></tr><tr><td>February</td><td>2</td><td>4</td></tr><tr><td>March</td><td>3</td><td>3</td></tr><tr><td>April</td><td>4</td><td>2</td></tr><tr><td>May</td><td>5</td><td>5</td></tr><tr><td>June</td><td>6</td><td>7</td></tr><tr><td>July</td><td>7</td><td>5</td></tr></table> 새로운 data로 세팅되어 chart가 로드됩니다.		Month	Red Series	Blue Series	January	1	5	February	2	4	March	3	3	April	4	2	May	5	5	June	6	7	July	7	5
Month	Red Series	Blue Series																							
January	1	5																							
February	2	4																							
March	3	3																							
April	4	2																							
May	5	5																							
June	6	7																							
July	7	5																							

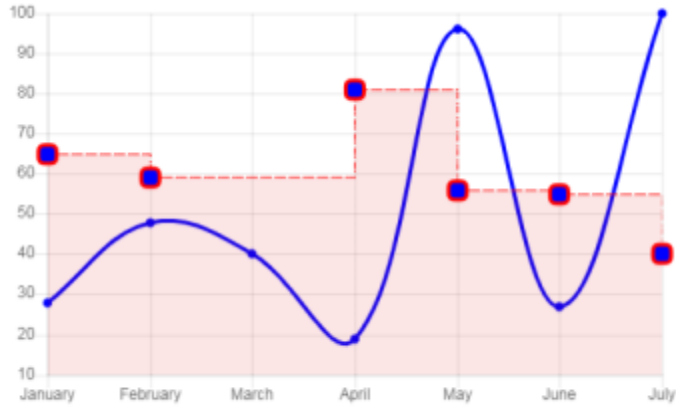
✓ Line

method	source
getData	<pre> function line_getData(){ var returnValue = line.getData(); alert('It takes an entire data array..' + '\n Check getData object in the console window. \n\n' + 'ex) Use the return value to extract the desired value. \n' + 'returnValue[0].value[0] : ' + returnValue[0].value[0]); console.log('line_getData : ' , returnValue); } </pre>
result	
 	

method	source
<code>update</code> (value,dataset,index)	<pre>function line_update(){ line.update(100,0,3); line.update(30,0,2); }</pre>
result	
 March와 April의 빨간 그래프 data가 update됩니다.	

method	source
addData (data, label)	<pre>function line_addData(){ line.addData([40, 60], "August"); }</pre>
result	
 August data가 추가됩니다.	

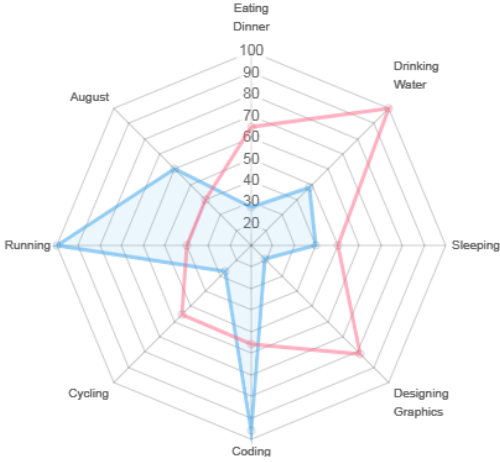
method	source
removeData (index)	<pre>function line_removeData(){ line.removeData(); }</pre>
result	
 첫 번째에 있는 January line data가 삭제됩니다.	

method	source
<code>setData</code> (data, option)	<pre>function line_setData(){ line.setData(data.lineData); line.option = option.lineOption; }</pre>
result	
 새로운 data가 세팅되어 chart가 로드됩니다.	

✓ Radar

method	source
getData	<pre> function radar_getData(){ var returnValue = radar.getData(); alert('It takes an entire data array..' + '\n\n Check getData object in the console window. \n\n' + 'ex) Use the return value to extract the desired value. \n\n' + 'returnValue[0].value[0] : ' + returnValue[0].value[0]); console.log('radar_getData : ', returnValue); } </pre>
result	
127.0.0.1 says It takes an entire data array. Check the getData object in the console window. ex) Use the return value to extract the desired value. returnValue[0].data[0] : 65 OK	

method	source
update (value,dataset,index)	function radar_update(){ radar.update(100,0,1); radar.update(50,0,2); }
result	
 Drinking Water의 value는 100으로, Sleeping의 value는 50으로 update됩니다.	

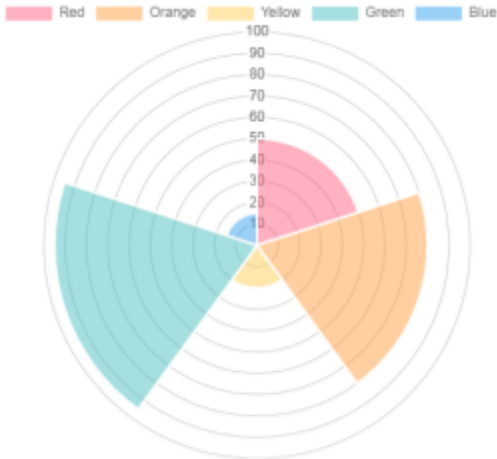
method	source
<code>addData</code> (data, label)	<pre>function radar_addData(){ radar.addData([40, 60], "August"); }</pre>
result	
	

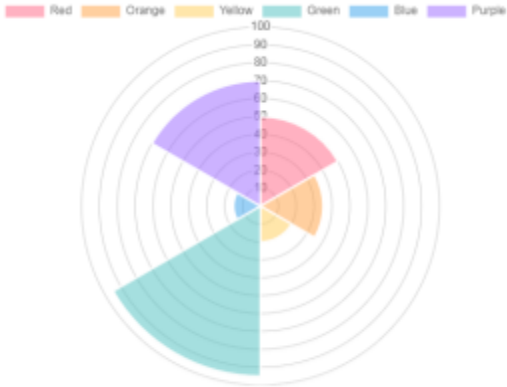
method	source
removeData (index)	<pre>function radar_removeData(){ radar.removeData(); }</pre>
result	
첫 번째에 있는 Eating Dinner data가 삭제됩니다.	

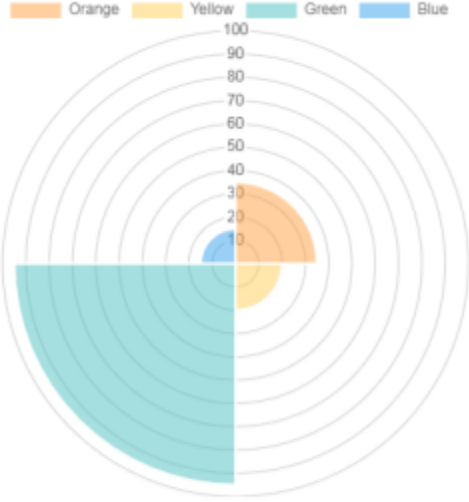
method	source
<code>setData</code> (data, option)	<pre>function radar_setData(){ radar.setData(data.radarData,option.radarOption); }</pre>
result	
 새로운 data가 세팅되어 chart가 로드됩니다.	

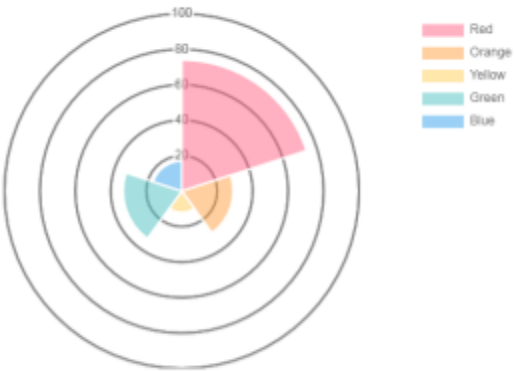
✓ Polar Area

method	source
getData	<pre> function polarArea_getData(){ var returnValue = polarArea.getData(); alert('It takes an entire data array..' + '\nCheck the getData object in the console window.\n\n' + 'ex) Use the return value to extract the desired value.\n' + 'returnValue[0].value : '+returnValue[0].value); console.log('polarArea_getData : ' , returnValue); } </pre>
result	
It takes an entire data array. Check the getData object in the console window. ex) Use the return value to extract the desired value. returnValue[0].data[0] : 50 <pre> polarArea_getData : ▼ [{...}] ⓘ 0: ▶ backgroundColor: (5) ["rgba(255, 99, 132, 0.5 ▶ data: (5) [50, 35, 20, 95, 15, _chartjs: {...}, ▶ hoverBackgroundColor: (5) ["rgba(255, 99, 132 label: "My dataset" ▶ _meta: {3: {...}} ▶ __proto__: Object length: 1 ▶ __proto__: Array(0) </pre>	

method	source
update (value,index)	<pre>function polarArea_update(){ polarArea.update(250,1); }</pre>
result	
Red Orange Yellow Green Blue  index가 1인 초록색 data가 250으로 update됩니다.	

method	source
addData (data, index)	<pre>function polarArea_addData(){ polarArea.addData({ value: 130, color: "#B48EAD", highlight: "#C69CBE", label: "Purple" },1); }</pre>
result	
 마지막 index에 보라색 data가 추가됩니다.	

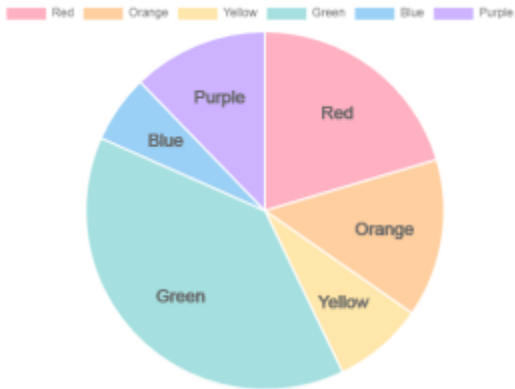
method	source
removeData (index)	<pre>function polarArea_removeData(){ polarArea.removeData(); }</pre>
result	
Orange Yellow Green Blue  첫 번째 index인 빨간색 data가 삭제됩니다.	

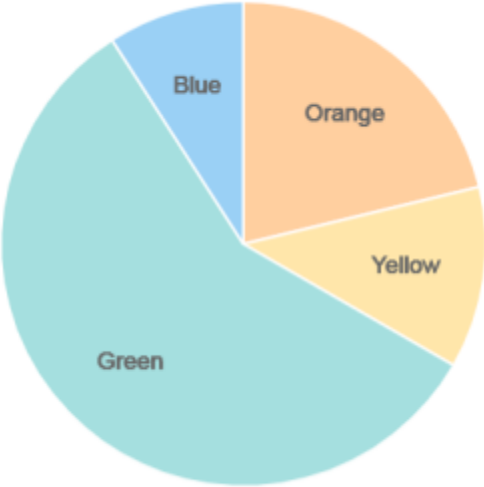
method	source
<code>setData</code> (data, option)	<pre>function polarArea_setData(){ polarArea.setData(data.polarAreaData2); }</pre>
result	
Y[wai] PolarArea Chart  새로운 data가 세팅되어 chart가 로드됩니다.	

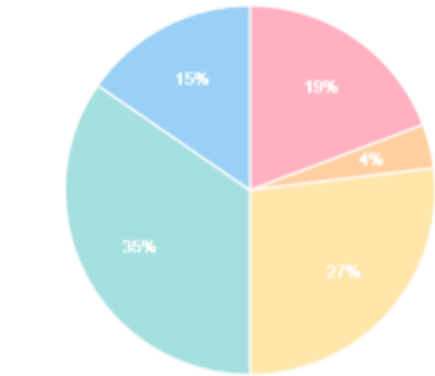
✓ Pie

method	source
getData	<pre>function pie_getData(){ var returnValue = pie.getData(); alert('It takes an entire data array.␣␣Check the getData object in the console window.␣␣␣' + 'ex) Use the return value to extract the desired value.␣␣' + returnValue[0].data[0] : '+ returnValue[0].data[0]); console.log('pie_getData : ' , returnValue); }</pre>
result	
It takes an entire data array. Check the getData object in the console window. ex) Use the return value to extract the desired value. returnValue[0].data[0] : 50 <pre>pie_getData : ▼ [{...}] ⓘ 0: ▶ backgroundColor: (5) ["rgba(255, 99, 132, 0.5' ▶ data: (5) [50, 35, 20, 95, 15, _chartjs: {...}, ▶ hoverBackgroundColor: (5) ["rgba(255, 99, 132 label: "Dataset 1" ▶ _meta: {4: {...}} ▶ __proto__: Object length: 1 ▶ __proto__: Array(0)</pre>	

method	source
<code>update</code> (value, dataset, index)	<pre>function pie_update(){ pie.update(100,1); }</pre>
result	
Red Orange Yellow Green Blue  index가 1인 주황색 data가 update됩니다.	

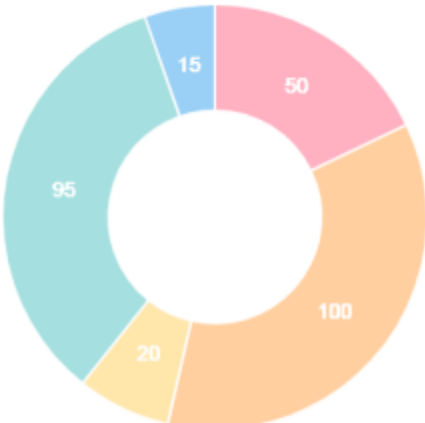
method	source
addData (data, index)	<pre> function pie_addData(){ pie.addData({ value: 130, color: "#B48EAD", highlight: "#C69CBE", label: "Purple" }); } </pre>
result	
 마지막 index에 보라색 data가 추가됩니다.	

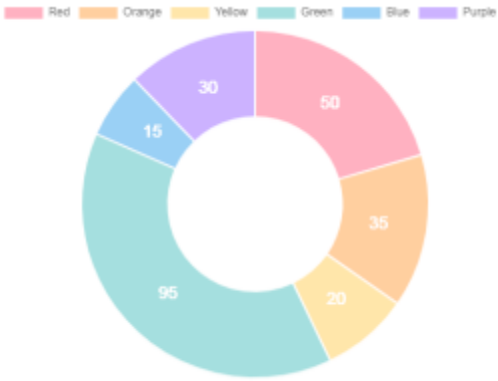
method	source
removeData (index)	<pre>function pie_removeData(){ pie.removeData(2); }</pre>
result	
Orange Yellow Green Blue  index가 0인 data부터 삭제됩니다.	

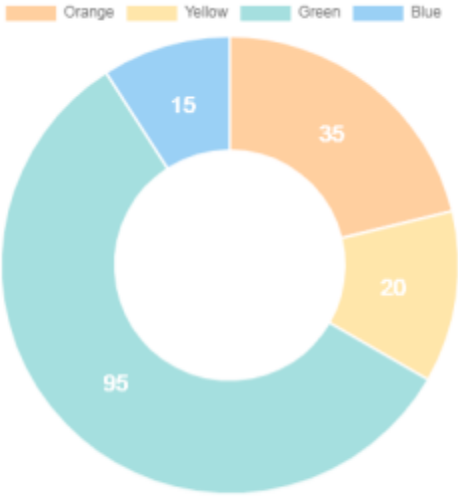
method	source												
setData (data, option)	<pre>function pie_setData(){ pie.setData(data.pieData2,option.pieOption2); }</pre>												
result													
Y[wai] Pie Chart Red Orange Yellow Green Blue <table border="1"><thead><tr><th>Category</th><th>Percentage</th></tr></thead><tbody><tr><td>Red</td><td>19%</td></tr><tr><td>Orange</td><td>4%</td></tr><tr><td>Yellow</td><td>27%</td></tr><tr><td>Green</td><td>35%</td></tr><tr><td>Blue</td><td>15%</td></tr></tbody></table> 새로운 data가 세팅되어 chart가 로드됩니다.		Category	Percentage	Red	19%	Orange	4%	Yellow	27%	Green	35%	Blue	15%
Category	Percentage												
Red	19%												
Orange	4%												
Yellow	27%												
Green	35%												
Blue	15%												

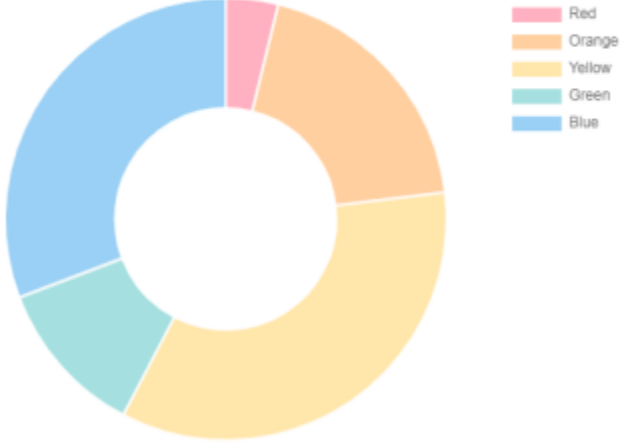
✓ doughnut

method	source
getData	<pre>function doughnut_getData(){ var returnValue = doughnut.getData(); alert('It takes an entire data array.␣␣Check the getData object in the console window.␣␣␣' + 'ex) Use the return value to extract the desired value.␣␣' + returnValue[0].data[0] : '+ returnValue[0].data[0]); console.log('doughnut_getData : ' , returnValue); }</pre>
result	
It takes an entire data array. Check the getData object in the console window. ex) Use the return value to extract the desired value. returnValue[0].data[0] : 50	<pre>doughnut_getData : ▼ [{...}] ⓘ 0: ▶ backgroundColor: (5) ["rgba(255, 99, 132, 0.1", ▶ data: (5) [50, 35, 20, 95, 15, _chartjs: {...}, label: "Dataset 1" ▶ _meta: {5: {...}} ▶ __proto__: Object length: 1 ▶ __proto__: Array(0)</pre>

method	source
<code>update</code> (value, dataset, index)	<pre>function doughnut_update(){ doughnut.update(100,1); }</pre>
result	
Red Orange Yellow Green Blue  index가 1인 주황색 data가 100으로 update됩니다.	

method	source
addData (data, label)	<pre>function doughnut_addData(){ doughnut.addData({ value: 130, color: "#B48EAD", highlight: "#C69CBE", label: "Purple" }); }</pre>
result	
 마지막 index에 보라색 data가 추가됩니다.	

method	source
removeData (index)	<pre>function doughnut_removeData(){ doughnut.removeData(); }</pre>
result	
 index가 0인 data부터 삭제됩니다.	

method	source
<code>setData</code> (data, option)	<pre>function doughnut_setData(){ doughnut.setData(data.doughnutData2,option.doughnutOption2); }</pre>
result	
Y[wai] Doughnut Chart  새로운 data가 세팅되어 chart가 로드됩니다.	

■ Events

onclick, ondblclick

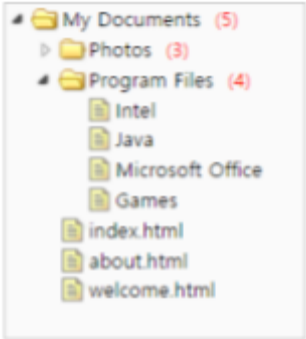
```
/*-----  
 * body source  
-----*/  
<chart id="chart_bar" type="bar"  
    onClick="eventAll(event);"  
    onDbClick="eventAll(event);"  
></chart>  
  
/*-----  
 * script source  
-----*/  
function eventAll(obj){  
    if(obj.event == "onclick"){  
        console.log('★----- onclick Event');  
    }  
    else if(obj.event == "ondblclick"){  
        console.log('★----- ondblclick Event');  
    }  
}
```

6. Y[wai] Component 예제

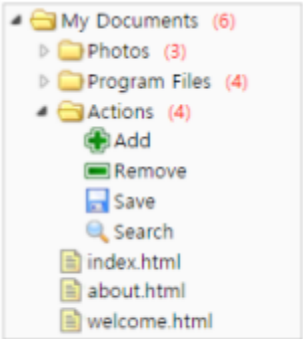
CC. TreeView

■ Basic Properties

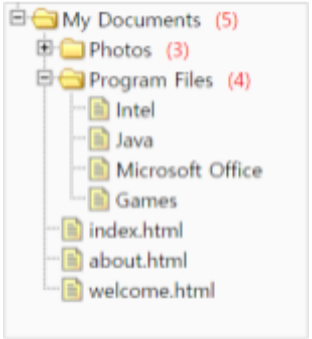
url, data, animate, lines, formatter, style

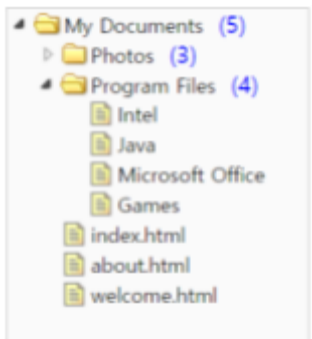
property	source
url	tree1.url='./testData/tree_data1.json';
result	
 해당 url에 있는 json file을 불러 data를 component에 세팅합니다.	

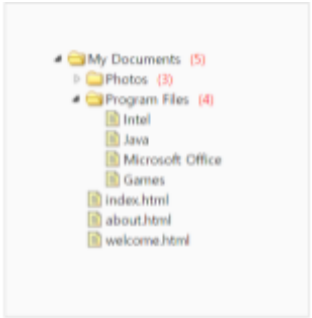
property	source
data	<pre> function dataAttr() { tree1.data = [{ "text": "My Documents", "children": [{ "text": "Photos", "state": "closed", "children": [{ "text": "Friend" }, { "text": "Wife" }, { "text": "Company" }] }, { "text": "Program Files", "state": "closed", "children": [{ "text": "Intel" }, { "text": "Java" }, { "text": "Microsoft Office" }, { "text": "Games" }] }, { "text": "Actions", "children": [{ "text": "Add", "iconCls": "icon-add" }, { "text": "Remove", "iconCls": "icon-remove" }, { "text": "Save", "iconCls": "icon-save" }, { "text": "Search", "iconCls": "icon-search" }] }, { "text": "index.html" }, { "text": "about.html" }, { "text": "welcome.html" }] } </pre>

	<pre> }} }}; }</pre>
result	
 json 형태의 data를 component에 세팅합니다.	

property	source
animate	tree1.animate = 'true';
result	
Tree의 폴더를 열고 닫을 때 animation 효과를 줄 수 있습니다.	

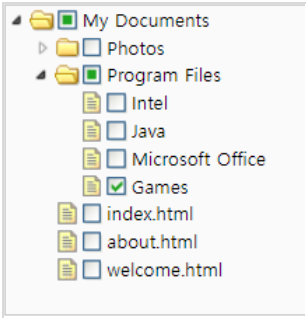
property	source
lines	tree1.lines = 'true';
result	
 tree 형태에서 line이 표시됩니다.	

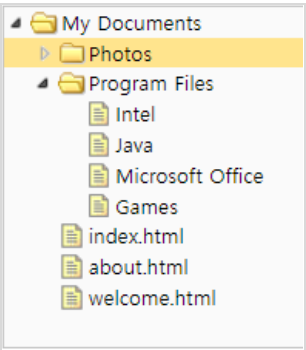
property	source
formatter	<pre>tree1.formatter = 'myFormatter2'; function myFormatter2(node){ var s = node.text; if (node.children){ s += "&nbspnbsp;<'span style=W\"color:blue;font-size:14pxW\"> (" +node.children.length + ")"; } return s; }</pre>
result	
 child node의 개수를 표시합니다.	

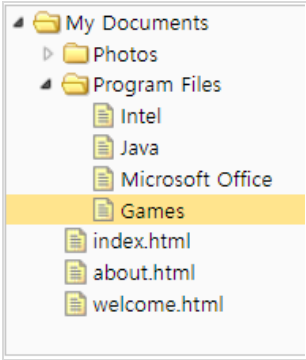
property	source
style	tree1.style = 'margin:40px;height:200px;';
result	
 <pre> # My Documents (5) ├── Photos (3) └── Program Files (4) ├── Intel ├── Java ├── Microsoft Office └── Games index.html about.html welcome.html </pre> component에 style을 적용합니다.	

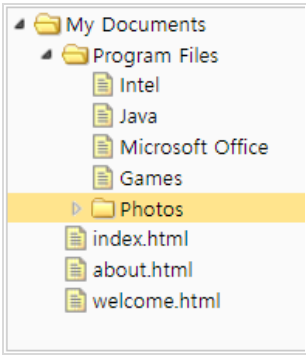
■ Edit Properties

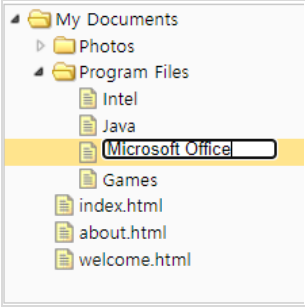
checkbox, cascadeCheck, onlyLeafCheck, dnd, editable

property	source
checkbox	<pre>function checkboxAttr(tf) { tree11.checkbox = tf; }</pre>
result	
	

property	source
cascadeCheck	<pre>function cascadeCheckAttr(tf) { tree11.cascadeCheck = tf; }</pre>
result	
	

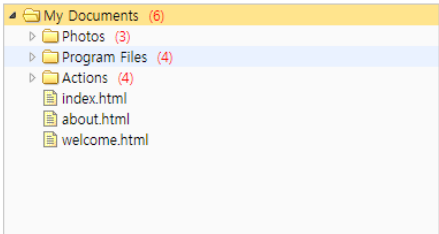
property	source
onlyLeafCheck	<pre>function onlyLeafCheckAttr(tf) { tree11.onlyLeafCheck = tf; }</pre>
result	
	

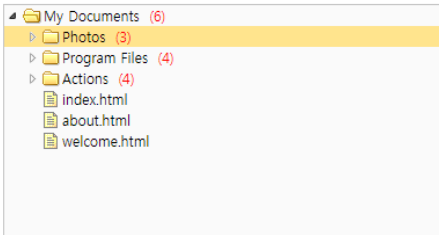
property	source
dnd	<pre>function dndAttr(tf) { tree11.dnd = tf; }</pre>
result	
	

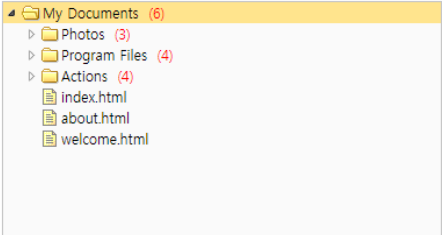
property	source
editable	<pre>function editableAttr(tf) { tree11.editable = tf; }</pre>
result	
 A screenshot of a file explorer window. The left pane shows a tree view of the file system. The right pane shows the contents of the selected folder. The 'Microsoft Office' folder is highlighted in the left pane. The right pane shows a list of files: index.html, about.html, and welcome.html. My Documents Photos Program Files Intel Java Microsoft Office Games index.html about.html welcome.html	

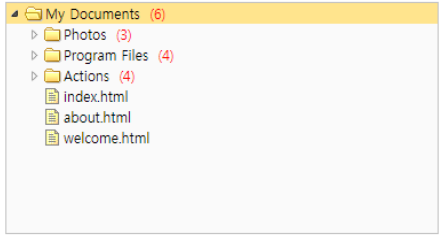
■ Readonly Properties

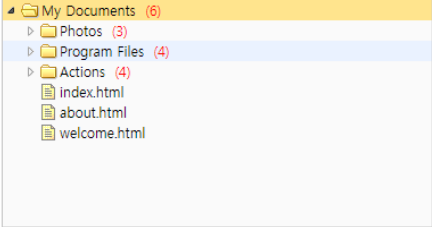
nodeId, nodeLevel, nodeValue, selected, hasChildren, nodeCount

property	source
nodeId	[Source : ID of node selected in tree] tree2.nodeId -->
result	
	

property	source
nodeLevel	[Source : Level of selected node in tree] tree2.nodeLevel -->
result	
	

property	source
nodeValue	[Source : Value of selected node in tree] tree2.nodeValue -->
result	
	

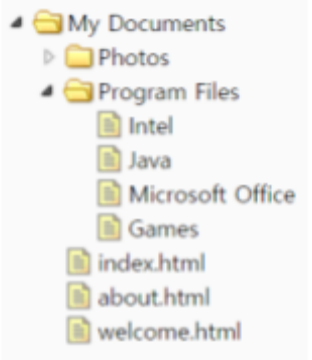
property	source
selected	[Source : Whether node is selected in tree] tree2.selected --> false
result	
	

property	source
hasChildren	[Source : Whether the child node of the selected node in the tree exists] tree2.hasChildren -->
result	
	

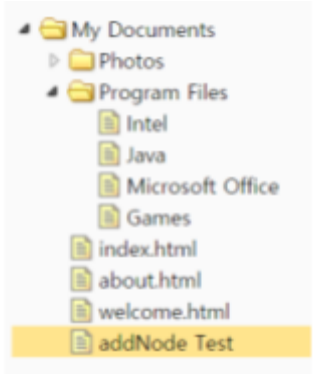
property	source
nodeCount	[Source : total number of nodes in tree] tree2.nodeCount --> 18
result	
	

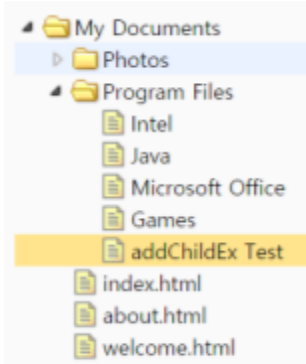
■ Data Methods

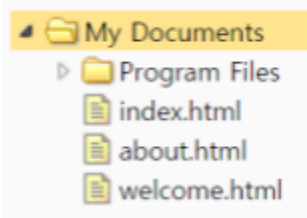
loadData, clear, addNode, addChildEx, delete, deleteChildren, edit

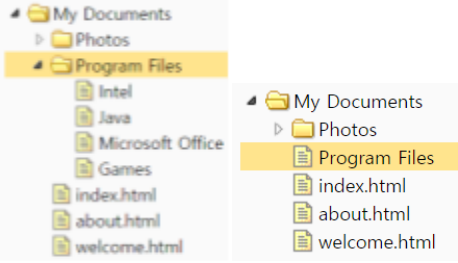
method	source
loadData	<pre>tree3.loadData([{ "id":0,"text":"My Documents","children":[{"id":1,"text":"Photos","state":"closed","children":[{"text":"Friend","_depth":2,"_tld":2,"_parents":"0,1","type":"file"}, {"id":12,"text":"Wife","_depth":2,"_tld":3,"_parents":"0,1","type":"file"}, {"text":"Company","_depth":2,"_tld":4,"_parents":"0,1","type":"file"}]}, {"_depth":1,"_tld":1,"_parents":"0","type":"folder"}, {"id":2,"text":"Program Files","state":"closed","children":[{"text":"Intel","_depth":2,"_tld":6,"_parents":"0,5","type":"file"}, {"text":"Java","type":"js","_depth":2,"_tld":7,"_parents":"0,5"}, {"text":"Microsoft Office","_depth":2,"_tld":8,"_parents":"0,5","type":"file"}, {"text":"Games","_depth":2,"_tld":9,"_parents":"0,5","type":"file"}]}, {"_depth":1,"_tld":5,"_parents":"0","type":"folder"}, {"id":3,"text":"Actions","state":"closed","children":[{"text":"Add","_depth":2,"_tld":11,"_parents":"0,10","type":"file"}, {"text":"Remove","_depth":2,"_tld":12,"_parents":"0,10","type":"file"}, {"text":"Save","_depth":2,"_tld":13,"_parents":"0,10","type":"file"}, {"text":"Search","_depth":2,"_tld":14,"_parents":"0,10","type":"file"}]}, {"_depth":1,"_tld":10,"_parents":"0","type":"folder"}, {"id":4,"text":"index.html","type":"html","_depth":1,"_tld":15,"_parents":"0"}, {"id":5,"text":"about.html","type":"html","_depth":1,"_tld":16,"_parents":"0"}, {"id":6,"text":"welcome.html","type":"html","_depth":1,"_tld":17,"_parents":"0"}]}, {"_depth":0,"_tld":0,"type":"folder","state":"open"}]);</pre>
result	
 json 형태의 data를 component에 세팅하여 로드합니다.	

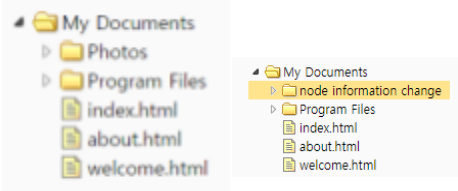
method	source
clear	tree3.clear();
result	
component에 data가 삭제됩니다.	

method	source
addNode (pId, text, imgIndex, selIndex, key)	tree3.addNode(0,"addNode Test"); // -1 : nodeId of My Documents
result	
 임의의 로드 하위 로드를 추가합니다.	

method	source
addChildEx (text, imgIndex, selIndex, key)	<pre>tree3.addChildEx("addChildEx Test");</pre>
result	
 현재 선택된 로드예 하위 로드를 추가합니다.	

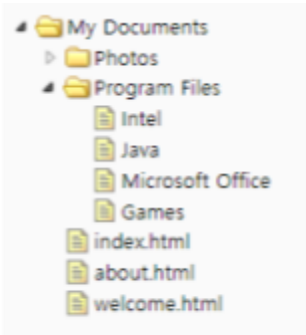
method	source
delete (nodeId)	<pre>tree3.delete(0);</pre>
result	
 Id가 0인 Photos 로드가 삭제됩니다.	

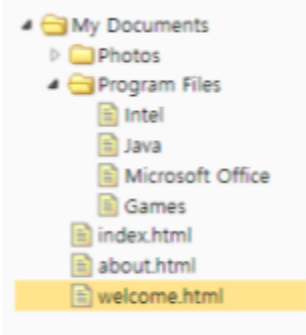
method	source
deleteChildren (nodeId)	tree3.deleteChildren();
result	
 현재 선택된 로드의 하위 로드를 삭제합니다.	

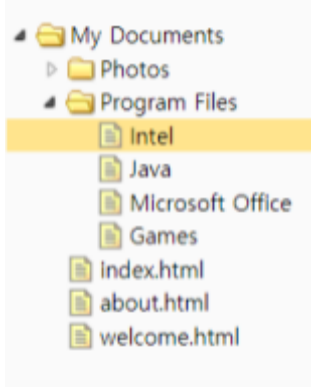
method	source
edit (text,imgIndex,sellIndex)	tree3.edit("Modify the node information.");
result	
 현재 선택된 로드의 정보를 변경합니다.	

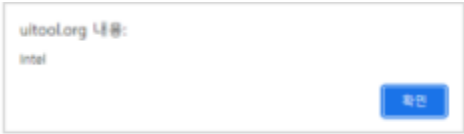
■ Get Methods

getData, find, getNode, getSelected, isLeaf

method	source
getData	tree3.getData();
result	
 tree의 전체 data node를 가지고 옵니다.	

method	source
find (nodeId)	<pre>var node = tree3.find(11); var stringNode = JSON.stringify(node);</pre>
result	
 Id가 11인 node를 찾아서 return합니다.	

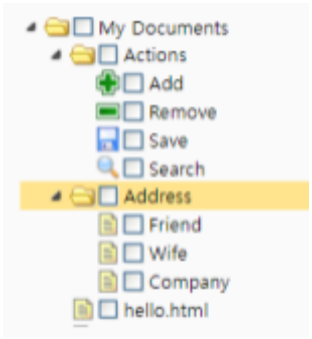
method	source
getNode (target)	<pre>var node_find = tree3.find(5); var node = tree3.getNode(node_find.target);</pre>
result	
<pre>{ "text": "Intel", "state": "open", "target": {}, "id": 5 }</pre>  node가 object로 return됩니다.	

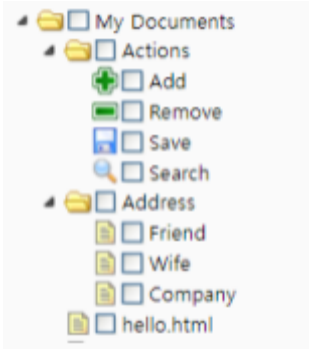
method	source
getSelected	<pre>tree3.getSelected();</pre>
result	
 tree에 선택된 node의 정보를 object로 return합니다.	

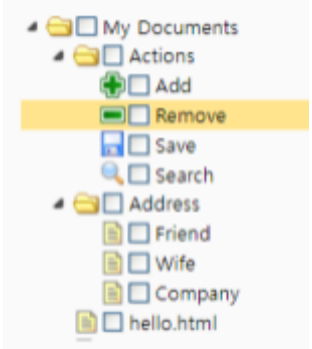
method	source
isLeaf (nodeId)	<pre>var node = tree3.getSelected(); var aa = tree3.isLeaf(node.target);</pre>
result	
target의 최하위 node 여부를 확인합니다. true 혹은 false로 return됩니다.	

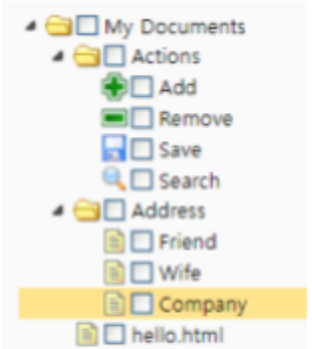
■ Select Methods

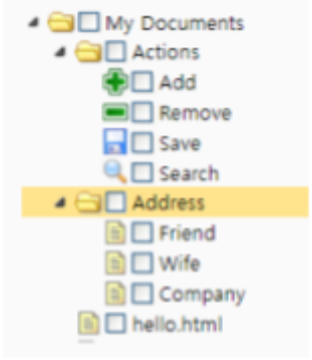
select, unselect, moveNode, moveNodeEx, moveParent, moveFirstChild, moveLastChild, moveNextChild

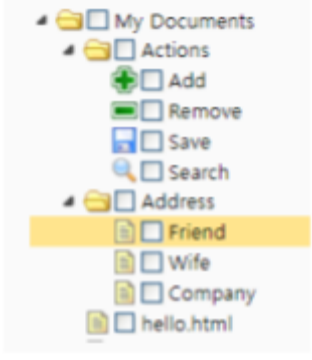
method	source
select (target)	<pre>var node =tree4.find(5); tree4.select(node.target);</pre>
result	
 target을 select합니다.	

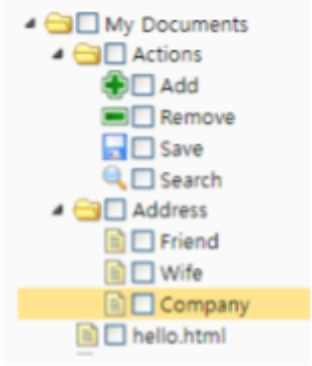
method	source
unselect	<pre>tree4.unselect();</pre>
result	
 선택된 상태를 해제합니다.	

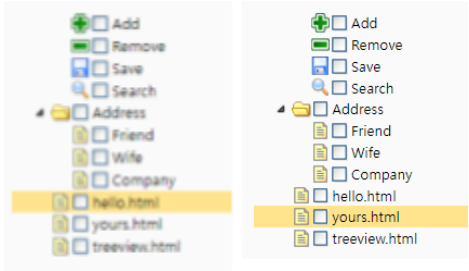
method	source
moveNode (target)	<pre>var node =tree4.moveNode(2); tree4.expandTo(node.target);</pre>
result	
 원하는 node로 이동합니다.	

method	source
moveNodeEx (key)	<pre>tree4.moveNodeEx("test_key"); // The nodeId value for Company node is "test_key".</pre>
result	
 component에서 임의의 key 값이 포함된 node 정보를 가진 node를 선택합니다.	

method	source
moveParent	tree4.moveParent();
result	
 component에서 현재 선택된 node의 Parent Node를 선택합니다.	

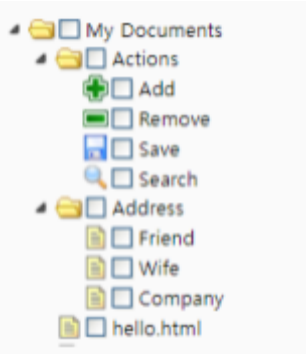
method	source
moveFirstChild	tree4.moveFirstChild();
result	
 현재 선택된 node의 children 중 첫 번째 Child Node를 선택합니다.	

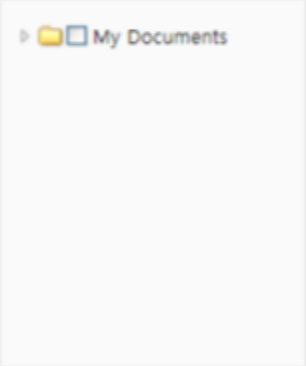
method	source
moveLastChild	tree4.moveLastChild();
result	
 현재 선택된 node의 children 중 마지막 Child Node를 선택합니다.	

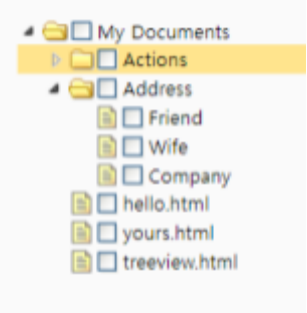
method	source
moveNextChild	tree4.moveNextChild();
result	
 현재 선택된 node의 형제 node 중 자신 다음의 node를 선택합니다.	

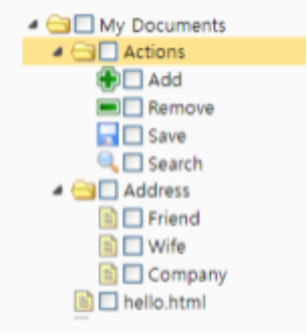
■ Action Methods

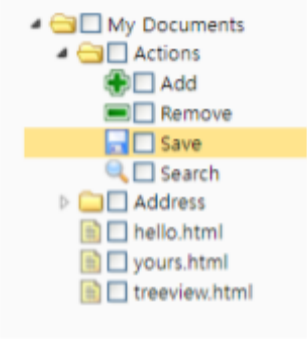
expandAll, collapseAll, collapse, expand, expandTo, toggle, collapseEx, expandEx

method	source
expandAll	tree4.expandAll();
result	
 tree 전체 node를 펼칩니다.	

method	source
collapseAll	tree4. collapseAll();
result	
 tree 전체 node를 접습니다.	

method	source
collapse (target)	<pre>var node = tree4.getSelected(); tree4.collapse(node.target);</pre>
result	
 target의 node를 접습니다.	

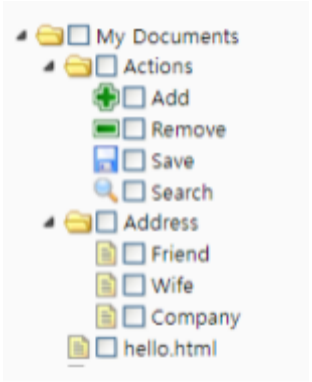
method	source
expand (target)	<pre>tree4.expand();</pre>
result	
 target의 node를 펼칩니다.	

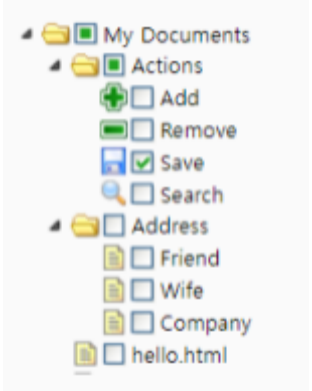
method	source
expandTo (target)	var node =tree4.find(3); tree4.expandTo(node.target);
result	
 임의의 nodeId=3을 가진 save node까지 펼치고 선택합니다.	

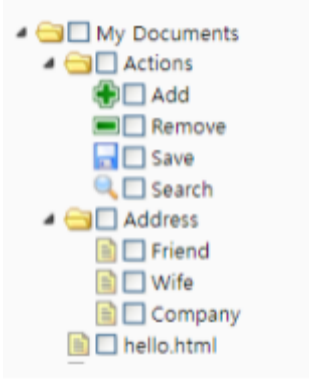
method	source
toggle (target)	var node = tree4.getSelected(); tree4.toggle(node.target);
result	
target을 번갈아 펼치고 접습니다.	

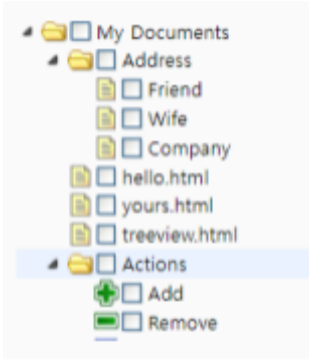
■ Edit Methods

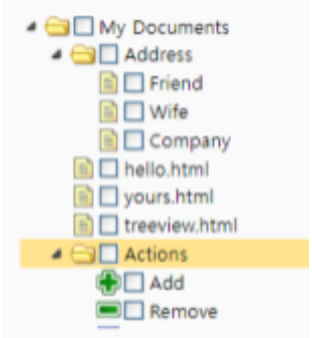
getChecked, check, uncheck, enableDnd, disableDnd, beginEdit

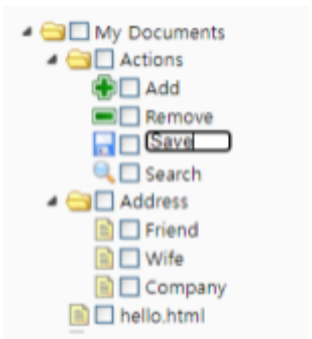
method	source
getChecked	tree4.getChecked(param);
result	
 check된 node를 가져옵니다.	

method	source
check	<pre>var node = tree3.find(3); tree4.uncheck(node.target);</pre>
result	
 target node를 check합니다.	

method	source
unchecked	tree4.getChecked(param);
result	
 target node를 unchecked합니다.	

method	source
enableDnd	tree4.enableDnd();
result	
 drag&drop을 할 수 있습니다.	

method	source
disableDnd	tree4.disableDnd();
result	
 drag&drop을 할 수 없습니다.	

method	source
beginEdit	<pre>var node = tree3.find(3); tree4.beginEdit(node.target);</pre>
result	
 node를 편집합니다.	

■ Events

onLoadSuccess, onLoadError, onContextMenu, onClick, onDbClick, onBeforeExpand, onExpand, onBeforeCollapse, onCollapse, onBeforeSelect, onSelect, onBeforeCheck, onCheck, onBeforeDrag, onStartDrag, onStopDrag, onDragEnter, onDragOver, onDragLeave, onBeforeDrop, onDrop, onBeforeEdit, onAfterEdit, onCancelEdit

```
/*-----  
* BODY  
-----*/  
<treeview id="tree5" url="./testData/tree_data1.json" animate=true style  
="margin:20px;height:260px;"  
    onContextMenu="eventAll(event);"  
    onLoadSuccess="eventAll(event);"  
    onLoadError="eventAll(event);"  
    onClick="eventAll(event);"  
    onDbClick="eventAll(event);"  
    onBeforeExpand="eventAll(event);"  
    onExpand="eventAll(event);"  
    onBeforeCollapse="eventAll(event);"  
    onCollapse="eventAll(event);"  
    onBeforeSelect="eventAll(event);"  
    onSelect="eventAll(event);"  
    onBeforeCheck="eventAll(event);"  
    onCheck="eventAll(event);"  
    onBeforeDrag="eventAll(event);"  
    onStartDrag="eventAll(event);"  
    onStopDrag="eventAll(event);"  
    onDragEnter="eventAll(event);"  
    onDragOver="eventAll(event);"  
    onDragLeave="eventAll(event);"  
    onBeforeDrop="eventAll(event);"  
    onDrop="eventAll(event);"  
    onBeforeEdit="eventAll(event);"  
    onAfterEdit="eventAll(event);"  
    onCancelEdit="eventAll(event);"  
></treeview>  
  
/*-----  
* JAVASCRIPT  
-----*/  
function eventAll(obj) {  
    if(obj.event == "onContextMenu"){  
        console.log('★---- onContextMenu Event');  
        console.log("e : " , obj.args.e , "Wnnode : " , obj.args.node);  
        $('#menu').menu('show',{
```

```

        left: obj.args.e.pageX,
        top: obj.args.e.pageY
    });
}
else if(obj.event == "onLoadSuccess"){
    console.log('★---- onLoadSuccess Event');
    console.log("node : " , obj.args.node , "Wndata : " , obj.args.data);
}
else if(obj.event == "onLoadError"){
    console.log('★---- onLoadError Event');
}
else if(obj.event == "onClick"){
    console.log('★---- onClick Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onDbClick"){
    console.log('★---- onDbClick Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onBeforeExpand"){
    console.log('★---- onBeforeExpand Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onExpand"){
    console.log('★---- onExpand Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onBeforeCollapse"){
    console.log('★---- onBeforeCollapse Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onCollapse"){
    console.log('★---- onCollapse Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onBeforeSelect"){
    console.log('★---- onBeforeSelect Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onSelect"){
    console.log('★---- onSelect Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onBeforeCheck"){
    console.log('★----- onBeforeCheck Event');
    console.log("node : " , obj.args.node);
    console.log("checked : " , obj.args.checked);
}
}

```



```

else if(obj.event == "onCheck"){
    console.log("node : " , obj.args.node);
    console.log("checked : " , obj.args.checked);
}
else if(obj.event == "onBeforeDrag"){
    console.log('★---- onBeforeDrag Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onStartDrag"){
    console.log('★---- onStartDrag Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onStopDrag"){
    console.log('★---- onStopDrag Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onDragEnter"){
    console.log('★---- onDragEnter Event');
    console.log("target : " , obj.args.target);
    console.log("source : " , obj.args.source);
}
else if(obj.event == "onDragOver"){
    console.log('★---- onDragOver Event');
    console.log("target : " , obj.args.target);
    console.log("source : " , obj.args.source);
}
else if(obj.event == "onDragLeave"){
    console.log('★---- onDragLeave Event');
    console.log("target : " , obj.args.target);
    console.log("source : " , obj.args.source);
}
else if(obj.event == "onBeforeDrop"){
    console.log('★---- onBeforeDrop Event');
    console.log("target : " , obj.args.target);
    console.log("source : " , obj.args.source);
    console.log("point : " , obj.args.point);
}
else if(obj.event == "onDrop"){
    console.log('★---- onDrop Event');
    console.log("target : " , obj.args.target);
    console.log("source : " , obj.args.source);
    console.log("point : " , obj.args.point);
}
else if(obj.event == "onBeforeEdit"){
    console.log('★---- onBeforeEdit Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onAfterEdit"){
    console.log('★---- onAfterEdit Event');

```

```
        console.log("node : " , obj.args.node);
    }
    else if(obj.event == "onCancelEdit"){
        console.log('★---- onCancelEdit Event');
        console.log("node : " , obj.args.node);
    }
}

/-----*
* Functions of the menu to configure onContextMenu
-----*/

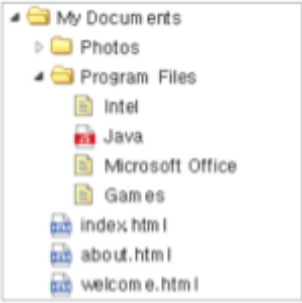
function t5_collapse(){
    var node = tree5.getSelected();
    tree5.collapse(node.target);
}
function t5_expand(){
    var node = tree5.getSelected();
    tree5.expand(node.target);
}
```

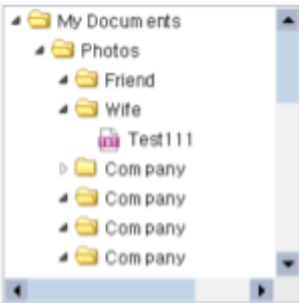
6. Y[wai] Component 예제

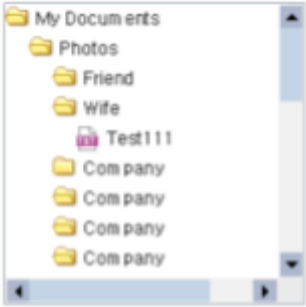
DD. cTree

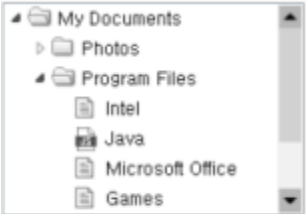
■ Basic Properties

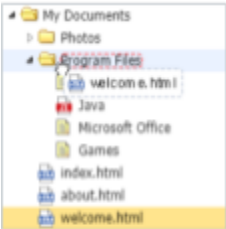
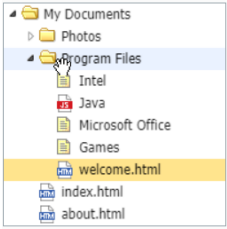
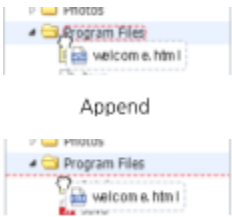
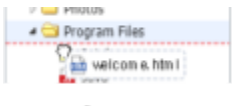
url, data, foldIcon, disabled, drag&drop, style

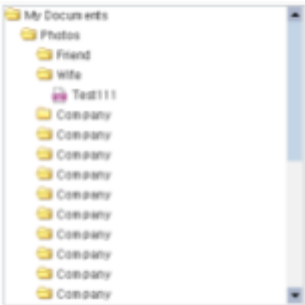
property	source
url	tree1.url='./testData/tree_data1.json';
result	
 해당 url에 있는 json file을 불러 data를 component에 세팅합니다.	

property	source
data	<pre> function dataAttr() { tree1.data = [{ "text": "My Documents", "type": "folder", "state": "open", "children": [{ "text": "Photos", "type": "folder", "state": "open", }, { "text": "Wife", "type": "folder", "state": "open", "children": [{ "id": 111, "text": "Jini", "type": "txt" }] } ]; } </pre>
result	
 json 형태의 data를 component에 세팅합니다.	

property	source
foldIcon	tree1.foldIcon = false;
result	
 삼각형 모양 아이콘이 사라집니다.	

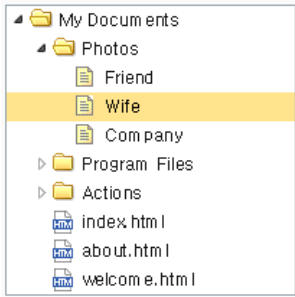
property	source
disabled	<pre>function disabledAttr() { if (tree1.disabled) tree1.disabled = false; else tree1.disabled = true; }</pre>
result	
 객체를 비활성화합니다.	

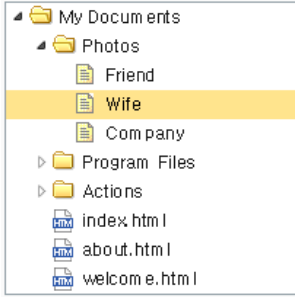
property	source
dnd	tree1.dnd = true;
result	
   Append  Bottom 옮기고 싶은 node를 drag해서 원하는 위치에 놓을 수 있습니다.	

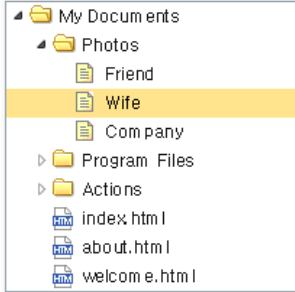
property	source
style	tree1.style = 'width:300px;margin:40px;height:300px;';
result	
 component에 style을 적용합니다.	

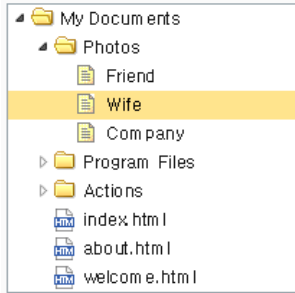
■ Readonly Properties

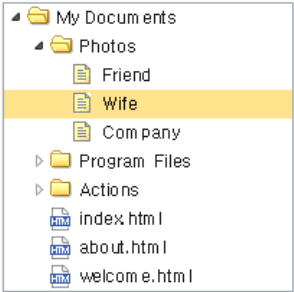
nodeId, nodeLevel, nodeValue, selected, hasChildren, nodeCount

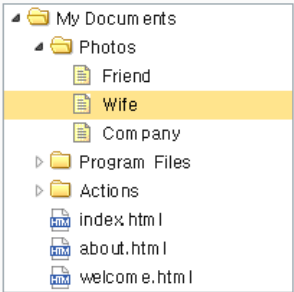
property	source
nodeId	[Source : ID of node selected in tree] tree2.nodeId -->
result	
	

property	source
nodeLevel	[Source : Level of selected node in tree] tree2.nodeLevel -->
result	
	

property	source
nodeValue	[Source : Value of selected node in tree] tree2.nodeValue -->
result	
	

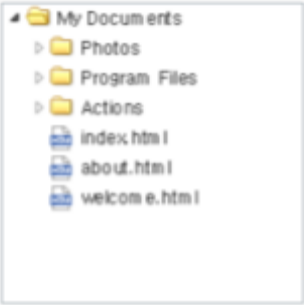
property	source
selected	[Source : Whether node is selected in tree] tree2.selected --> false
result	
	

property	source
hasChildren	[Source : Whether the child node of the selected node in the tree exists] tree2.hasChildren --> false
result	
	

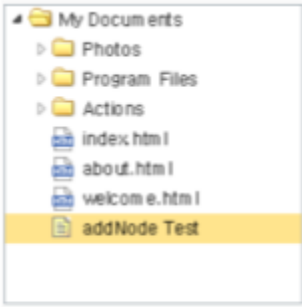
property	source
nodeCount	[Source : total number of nodes in tree] tree2.nodeCount --> 18
result	
	

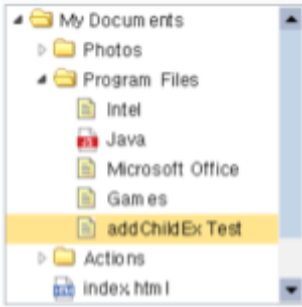
■ Data Methods

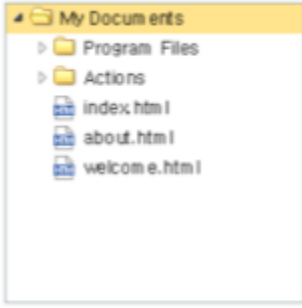
loadData, clear, addNode, addChildEx, delete, deleteChildren, edit

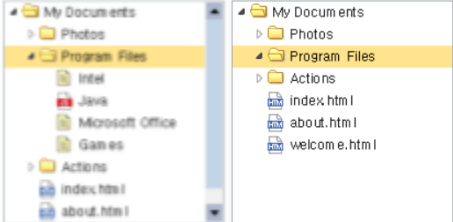
method	source
loadData	<pre>tree3.loadData([{ "id":0,"text":"My Documents","children":[{"id":1,"text":"Photos","state":"closed","children":[{"text":"Friend","_depth":2,"_tld":2,"_parents":"0,1","type":"file"}, {"id":12,"text":"Wife","_depth":2,"_tld":3,"_parents":"0,1","type":"file"}, {"text":"Company","_depth":2,"_tld":4,"_parents":"0,1","type":"file"}]}, {"_depth":1,"_tld":1,"_parents":"0","type":"folder"}, {"id":2,"text":"Program Files","state":"closed","children":[{"text":"Intel","_depth":2,"_tld":6,"_parents":"0,5","type":"file"}, {"text":"Java","_depth":2,"_tld":7,"_parents":"0,5"}, {"text":"Microsoft Office","_depth":2,"_tld":8,"_parents":"0,5","type":"file"}, {"text":"Games","_depth":2,"_tld":9,"_parents":"0,5","type":"file"}]}, {"_depth":1,"_tld":5,"_parents":"0","type":"folder"}, {"id":3,"text":"Actions","state":"closed","children":[{"text":"Add","_depth":2,"_tld":11,"_parents":"0,10","type":"file"}, {"text":"Remove","_depth":2,"_tld":12,"_parents":"0,10","type":"file"}, {"text":"Save","_depth":2,"_tld":13,"_parents":"0,10","type":"file"}, {"text":"Search","_depth":2,"_tld":14,"_parents":"0,10","type":"file"}]}, {"_depth":1,"_tld":10,"_parents":"0","type":"folder"}, {"id":4,"text":"index.html","type":"html","_depth":1,"_tld":15,"_parents":"0"}, {"id":5,"text":"about.html","type":"html","_depth":1,"_tld":16,"_parents":"0"}, {"id":6,"text":"welcome.html","type":"html","_depth":1,"_tld":17,"_parents":"0"}], "_depth":0,"_tld":0,"type":"folder", "state":"open"}]);</pre>
result	
 json 형태의 data를 component에 세팅하여 로드합니다.	

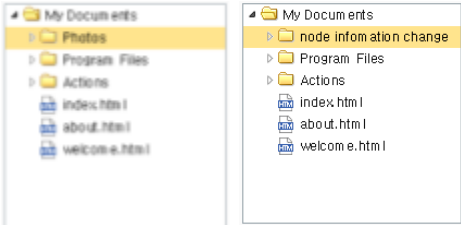
method	source
clear	tree3.clear();
result	
component에 data가 삭제됩니다.	

method	source
addNode (pId,text,type,strData,id)	tree3.addNode(0,"addNode Test"); // -1 : nodeId of My Documents
result	
 임의의 로드 하위 로드를 추가합니다.	

method	source
addChildEx (text,type,strData,id)	tree3.addChildEx("addChildEx Test");
result	
 현재 선택된 로드 하위 로드를 추가합니다.	

method	source
delete (nodeId)	tree3.delete(1);
result	
 Id가 1인 Photos 로드가 삭제됩니다.	

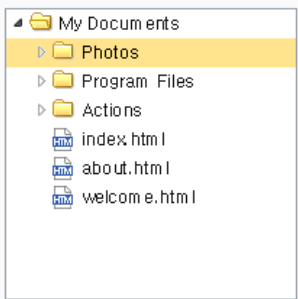
method	source
deleteChildren (nodeId)	tree3.deleteChildren();
result	
 현재 선택된 로드의 하위 로드를 삭제합니다.	

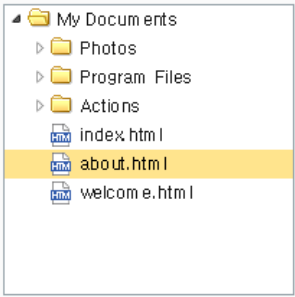
method	source
edit (text, type)	tree3.edit("Modify the node information.");
result	
 현재 선택된 로드의 정보를 변경합니다.	

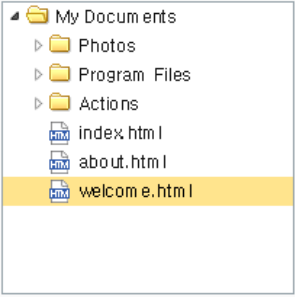
■ Get Methods

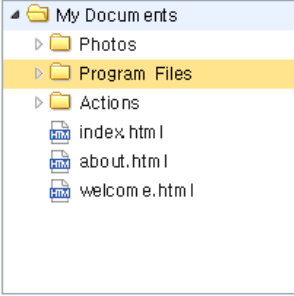
getData, find, getNode, getSelected, isLeaf

method	source
getData	tree3.getData();
result	
cTree의 전체 data node를 가지고 옵니다.	

method	source
find (nodeId)	tree3.find(1);
result	
	

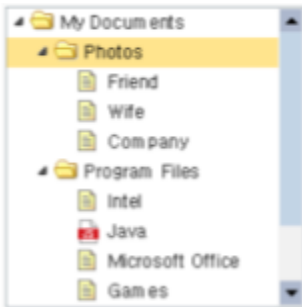
method	source
getNode (nodeId)	tree3.getNode(5);
result	
	

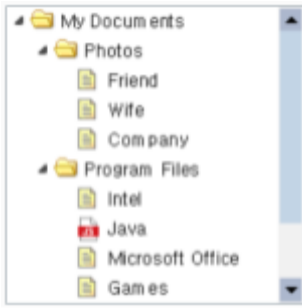
method	source
getSelected	tree3.getSelected();
result	
	

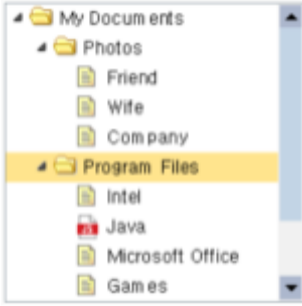
method	source
isLeaf (nodeId)	tree3.isLeaf();
result	
	

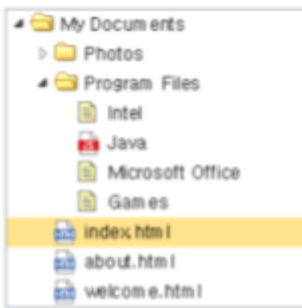
■ Select Methods

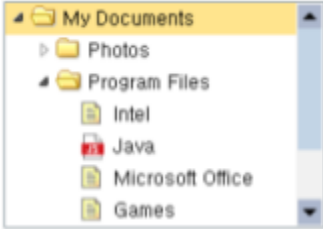
select, unselect, moveNode, moveNodeEx, moveParent, moveFirstChild, moveLastChild, moveNextChild

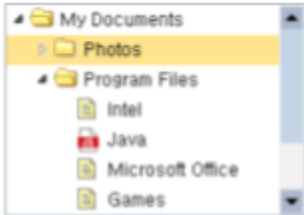
method	source
select (nodeId)	tree4.select(5);
result	
 해당 nodeId를 가진 node를 선택합니다.	

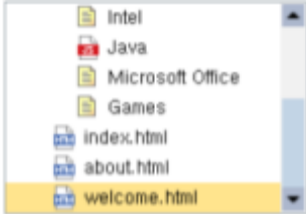
method	source
unselect	tree4.unselect();
result	
 선택된 상태를 해제합니다.	

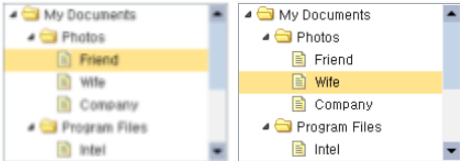
method	source
moveNode (nodeId)	tree4. moveNode(2);
result	
 원하는 node로 이동합니다.	

method	source
moveNodeEx (nodeId)	tree4.moveNodeEx("test_key"); // The nodeId value for Company node is "test_key".
result	
 임의의 id 값이 포함된 node 정보를 가진 node를 선택합니다.	

method	source
moveParent	tree4.moveParent();
result	
 현재 선택된 node의 Parent Node를 선택합니다.	

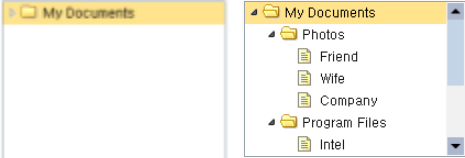
method	source
moveFirstChild	tree4.moveFirstChild();
result	
 현재 선택된 node의 children 중 첫 번째 child node를 선택합니다.	

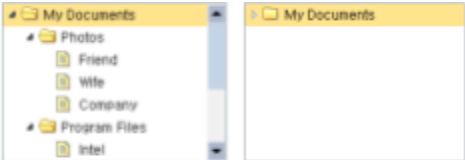
method	source
moveLastChild	tree4. moveLastChild();
result	
 현재 선택된 node의 children 중 마지막 child node를 선택합니다.	

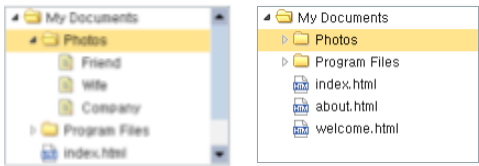
method	source
moveNextChild	tree4.moveNextChild();
result	
 현재 선택된 node의 형제 node 중 자신 다음의 node를 선택합니다.	

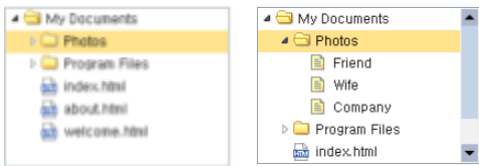
■ Action Methods

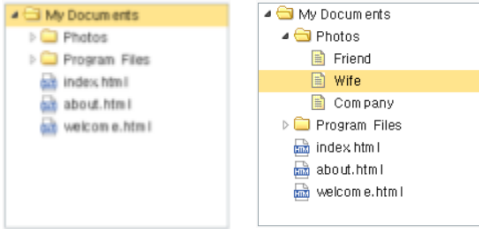
expandAll, collapseAll, collapse, expand, expandTo, toggle, collapseEx, expandEx

method	source
expandAll	tree4.expandAll();
result	
 전체 node를 펼칩니다.	

method	source
collapseAll	tree4.collapseAll();
result	
 전체 node를 접습니다.	

method	source
collapse (nodeId)	tree4. collapse();
result	
 임의의 nodeId를 가진 node를 접습니다. 인자를 넘기지 않으면 현재 선택된 node를 접습니다.	

method	source
expand (nodeId)	tree4. expand();
result	
 임의의 nodeId를 가진 node를 펼칩니다. 인자를 넘기지 않으면 현재 선택된 node를 펼칩니다.	

method	source
expandTo (nodeId)	tree4.expandTo(12);
result	
 임의의 nodeId=12를 가진 Wife node까지 펼치고 선택합니다.	

method	source
toggle (nodeId)	tree4.toggle(nodeId);
result	
임의의 nodeId를 가진 node를 번갈아 open하고 close합니다. 인자를 넘기지 않으면 현재 선택된 node로 수행합니다.	

■ Events

onLoadSuccess, onLoadError, onContextMenu, onClick, onDbClick, onBeforeExpand, onExpand, onBeforeCollapse, onCollapse, onBeforeSelect, onSelect

```
/*-----  
* BODY  
-----*/  
<ctree id="tree5" url="/testData/tree_data1.json" animate=true style="height:260px;"  
    onContextMenu="eventAll(event);"   
    onLoadSuccess="eventAll(event);"   
    onLoadError="eventAll(event);"   
    onClick="eventAll(event);"   
        onDbClick="eventAll(event);"   
    onBeforeExpand="eventAll(event);"   
    onExpand="eventAll(event);"   
    onBeforeCollapse="eventAll(event);"   
    onCollapse="eventAll(event);"   
    onBeforeSelect="eventAll(event);"   
    onSelect="eventAll(event);"   
        onBeforeDrag="eventAll(event);"   
        onStartDrag="eventAll(event);"   
        onStopDrag="eventAll(event);"   
        onDragOver="eventAll(event);"   
        onBeforeDrop="eventAll(event);"   
        onDrop="eventAll(event);"   
></ctree>  
  
/*-----  
* use event  
-----*/  
function eventAll(obj) {  
    if(obj.event == "onContextMenu"){  
        console.log('★---- onContextMenu Event');  
        console.log("e : ", obj.args.e , "Wnnode : " , obj.args.node);  
        $('#menu').menu('show',{  
            left: obj.args.e.pageX,  
            top: obj.args.e.pageY  
        });  
    }  
    else if(obj.event == "onLoadSuccess"){  
        console.log('★---- onLoadSuccess Event');  
        console.log("node : " , obj.args.node , "Wndata : " , obj.args.data);  
    }  
    else if(obj.event == "onLoadError"){  
        console.log('★---- onLoadError Event');  
    }  
}
```



```

else if(obj.event == "onClick"){
    console.log('★---- onClick Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onDbClick"){
    console.log('★---- onDbClick Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onBeforeExpand"){
    console.log('★---- onBeforeExpand Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onExpand"){
    console.log('★---- onExpand Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onBeforeCollapse"){
    console.log('★---- onBeforeCollapse Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onCollapse"){
    console.log('★---- onCollapse Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onBeforeSelect"){
    console.log('★---- onBeforeSelect Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onSelect"){
    console.log('★---- onSelect Event');
    console.log("node : " , obj.args.node);
}
else if(obj.event == "onBeforeDrag"){
    console.log('★---- onBeforeDrag Event');
    console.log("dragNode : " , obj.args.dragNode);
}
else if(obj.event == "onStartDrag"){
    console.log('★---- onStartDrag Event');
    console.log("dragNode : " , obj.args.dragNode);
}
else if(obj.event == "onStopDrag"){
    console.log('★---- onStopDrag Event');
    console.log("dragNode : " , obj.args.dragNode);
}
else if(obj.event == "onDragOver"){
    console.log('★---- onDragOver Event');
    console.log("dragNode : " , obj.args.dragNode);
    console.log("dropTarget : " , obj.args.dropTarget);
    console.log("dropPoint : " , obj.args.dropPoint);
}

```

```
}  
    else if(obj.event == "onBeforeDrop"){  
        console.log('★----- onBeforeDrop Event');  
        console.log("dragNode : " , obj.args.dragNode);  
        console.log("dropTarget : " , obj.args.dropTarget);  
        console.log("dropPoint : " , obj.args.dropPoint);  
    }  
    else if(obj.event == "onDrop"){  
        console.log('★----- onDrop Event');  
        console.log("dragNode : " , obj.args.dragNode);  
        console.log("dropTarget : " , obj.args.dropTarget);  
        console.log("dropPoint : " , obj.args.dropPoint);  
    }  
}
```

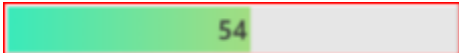
6. Y[wai] Component 예제

EE. Bar Progress

■ Properties

value, style, gradienstStart, gradientEnd, valueText, valueTextColor, emptyFill, lincecap, duration, animationStartValue, unit

property	source
value	bar1.value=60;
result	
 원하는 value를 세팅합니다. (default: 0)	

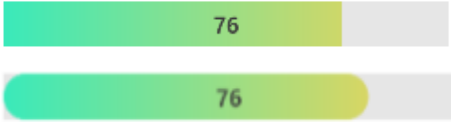
property	source
style	style="width:200px;height:50px; border:1px solid red;"
result	
 원하는 style을 세팅합니다.	

property	source
gradientStart gradientEnd	gradientstart = "blue" gradientend = "skyblue"
result	
 progress를 원하는 그라데이션 색상으로 설정합니다. (gradientstart default: #3aeabb) (gradientend default: #fdd250)	

property	source
valueText	bar1.valueText=true; bar1.valueText=false;
result	
 value를 표시하는 text 표시 여부를 설정합니다. (default: true)	

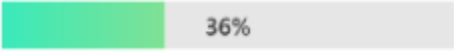
property	source
valueTextColor	bar1.valueTextColor ="blue";
result	
 valueText의 font color를 설정합니다. (default: #000000)	

property	source
emptyFill	bar1.emptyFill =" #ffe1e1";
result	
 progress의 빈 공간의 color를 설정합니다. (default: #e6e6e6)	

property	source
linecap	bar1.linecap ="butt"; bar1.linecap ="round";
result	
 progress의 끝 모양을 설정합니다. (default: butt)	

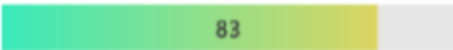
property	source
duration	bar1.duration =1500;
result	
animation의 동작 시간을 설정합니다. (default: 1200)	

property	source
animationStartValue	bar1.animationStartValue=0; bar1.animationStartValue=1;
result	
animation을 표시하는 시작값을 설정합니다. 0: 왼쪽에서 오른쪽으로 progress가 채워짐 1: 오른쪽에서 왼쪽으로 progress가 사라짐	

property	source
unit	bar1.unit="%";
result	
 valueText 표시 단위를 설정합니다.	

■ Methods

getValue, setValue

method	source
getValue	bar1.getValue();
result	
 value를 가져옵니다. (getValue: 83)	

method	source
setValue	bar1.setValue(value);
result	
 value를 설정합니다. (setValue: 48)	

■ Events

onclick, onDrawEnd

```
/*-----  
* BODY  
-----*/  
<bProgress id="barEvent"  
onClick="eventAll(event);"  
onDrawEnd="eventAll(event);"  
>  
</ bProgress>  
  
/*-----  
* SCRIPT - use event  
-----*/  
function eventAll(obj){  
    if (obj.event == 'onDrawEnd') {  
        console.log(obj.event + " : " + obj.srcElement.value);  
    } else {  
        alert(obj.event);  
    }  
}
```


6. Y[wai] Component 예제

FF. Circle Progress

■ Properties

value, style, arcCoef, thickness, gradientStart, gradientEnd, valueText, valueTextColor, emptyFill, reverse, linecap, duration, animationStartValue, unit

property	source
value	circle1.value=60;
result	
	

property	source
style	style="width:120; height:120px;
result	
	

property	source
arcCoef	circle1.arcCoef=7;
result	
 progress의 원 둘레 모양을 설정합니다. (default: 10)	

property	source
thickness	circle1.thickness=60;
result	
 progress의 두께를 설정합니다. (default: 20)	

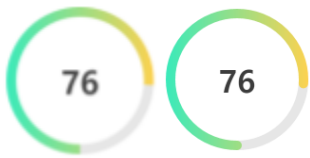
property	source
gradientStart gradientEnd	gradientstart = "blue" gradientend = "skyblue"
result	
 progress를 원하는 그라데이션 색상으로 설정합니다. (gradientstart default: #3aeabb) (gradientend default: #fdd250)	

property	source
valueText	circle1.valueText=true; circle1.valueText=false;
result	
 value를 표시하는 text 표시 여부를 설정합니다. (default: true)	

property	source
valueTextColor	circle1.valueTextColor = "#e6e6e6";
result	
 valueText의 font color를 설정합니다. (default: #000000)	

property	source
emptyFill	circle1.emptyFill = "#ffe1e1";
result	
 progress의 빈 공간 색상을 설정합니다. (default: #e6e6e6)	

property	source
reverse	circle1.reverse=true; circle1.reverse=false;
result	
animation의 시작 방향을 설정합니다. true: 반시계 방향 false: 시계 방향	

property	source
linecap	circle1.linecap ="butt"; circle1.linecap ="round";
result	
 progress의 끝 모양을 설정합니다. (default: butt)	

property	source
duration	circle1.duration =1500;
result	
animation의 동작 시간을 설정합니다. (default: 1200)	

property	source
animationStartValue	circle1.animationStartValue=0; circle1.animationStartValue=1;
result	
animation을 표시하는 시작값을 설정합니다. 0: 시계 방향 1: 반시계 방향	

property	source
unit	circle1.unit="%";
result	
 valueText의 표시 단위를 설정합니다.	

■ Methods

getValue, setValue

method	source
getValue	circle1.getValue();
result	
 value를 가져옵니다. (getValue: 70)	

method	source
setValue	circle1.setValue(value);
result	
 value를 세팅합니다. (setValue: 48)	

■ Events

onclick, onDrawEnd

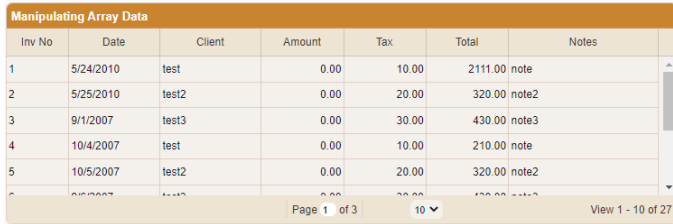
```
/*-----  
* BODY  
-----*/  
<cProgress id="circleEvent"  
onClick="eventAll(event);"  
onDrawEnd="eventAll(event);"  
>  
</ cProgress>  
  
/*-----  
* SCRIPT - use event  
-----*/  
function eventAll(obj){  
    if (obj.event == 'onDrawEnd') {  
        console.log(obj.event + " : " + obj.srcElement.value);  
    } else {  
        alert(obj.event);  
    }  
}
```

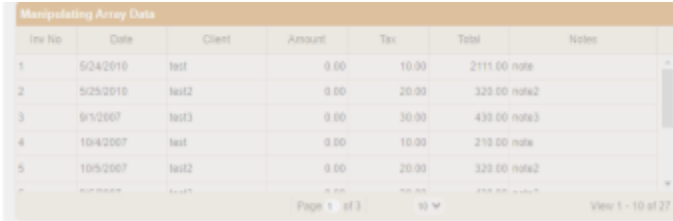

6. Y[wai] Component 예제

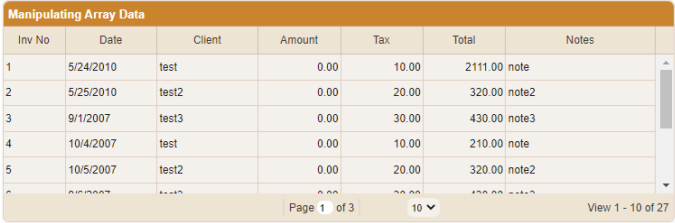
GG. External

■ Properties

name, disabled, readonly, style, createType, elements, customHtml

property	source
name	external1.name = '';
result	
	

property	source
disabled	external1.disabled = true;
result	
 객체를 비활성화합니다.	

property	source
readonly	external1.readonly = true;
result	
	

property	source
style	external2.style = 'width:200px;height:30px;';
result	
 원하는 style을 세팅합니다.	

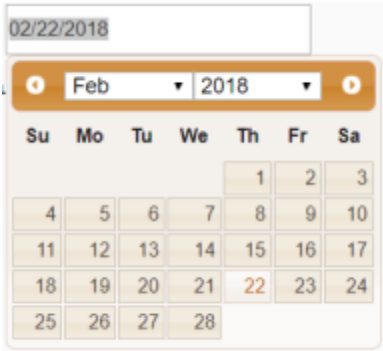
property	source
createType	external2.createType="elements"; external2.createType="customHtml";
result	
elements와 customHtml을 사용하여 객체를 생성할 수 있습니다.	

property	source
elements	<pre> <external id="external2" createType="elements"> <elem tag="div" id="cc1"></elem> </external> </pre>
result	
createType이 elements일 경우 객체를 <elem> </elem> 태그로 정의합니다. elem 태그 속성은 tag, id가 있습니다.	

property	source
customHtml	<pre> <external id="external2" createType="customHtml" <custHtml> <div id="cc1"></div> </custHtml> </external> </pre>
result	
createType이 customHtml일 경우 객체를 <custHtml> </custHtml> 태그로 정의합니다. custHtml 태그 안에 객체를 생성하여 정의합니다.	

■ Methods

focus, clear, reload

method	source
focus	external4.focus();
result	
 value를 가져옵니다.	

method	source
clear	external4.clear();
result	
library를 적용한 객체가 clear됩니다.	

method	source
reload	external4.reload();
result	
객체가 다시 load됩니다.	

■ Events

oninit

```
/*-----  
* BODY  
-----*/  
<external id="external1"  
    name="TestName1"  
    title="Test Title !!!"  
    oninit="load_jqGrid(event)"  
    createType="elements"  
    style="position:absolute;left: 120px;top:350px ;width: 700px; height:  
300px;">  
  
    <elem tag="table">  
    </elem>  
    <elem tag="div" id="plist47">  
    </elem>  
  
</external>  
  
/*-----  
* SCRIPT - use event  
-----*/  
function load_jqGrid(e) {  
    var mydata = [  
        {id:"1",invdate:"2010-05-24",name:"test",note:"note",tax:"10.00",total:"2111.00"}  
    ,  
  
    {id:"2",invdate:"2010-05-25",name:"test2",note:"note2",tax:"20.00",total:"320.00"},  
  
    {id:"3",invdate:"2007-09-01",name:"test3",note:"note3",tax:"30.00",total:"430.00"},  
        .....(skip)....  
    ];  
    jQuery(external1.elems[0]).jqGrid({  
        data: mydata,  
        datatype: "local",  
        height: 150,  
        rowNum: 10,  
        rowList: [10,20,30],  
        colNames:['Inv No','Date', 'Client', 'Amount','Tax','Total','Notes'],  
        colModel:[  
            {name:'id',index:'id', width:60, sorttype:"int"},  
            {name:'invdate',index:'invdate', width:90, sorttype:"date",  
formatter:"date"},
```

```
        {name:'name',index:'name', width:100},
        {name:'amount',index:'amount', width:80, align:"right",sorttype:"float",
formatter:"number"},
        {name:'tax',index:'tax', width:80, align:"right",sorttype:"float"},
        {name:'total',index:'total', width:80,align:"right",sorttype:"float"},
        {name:'note',index:'note', width:150, sortable:false}
    ],
    pager: "#plist47",
    viewrecords: true,
    caption: "Manipulating Array Data"
});
}
```

6. Y[wai] Component 예제

HH. SubFrame

■ Properties

screen, window

property	source
screen	subFrame1.screen = “/biz/dp/DP10.html”;
result	
subFrame 컴포넌트에 screen 속성값에 해당하는 화면을 호출합니다.	

property	source
window	console.log(subFrame1.window);
result	
subFrame 컴포넌트에 호출된 HTML 페이지의 contentWindow 객체를 반환합니다.	

6. Y[wai] Component 예제

N. Search Edit

■ Properties

disabled, readonly, hideClearButton, hideSearchButton, searchPosition, clearButtonImage, searchButtonImage, buttonFocusable

property	source
disabled	searchEdit1.disabled = true;
result	
	

property	source
readonly	searchEdit1.readonly = true;
result	
	

property	source
hideClearButton	searchEdit1.hideClearButton = true;
result	
	

property	source
hideSearchButton	searchEdit1.hideSearchButton = true;
result	
searchEdit Test 	

property	source
searchPositon	searchEdit1.searchPositon = left;
result	
searchEdit Test  	

property	source
clearButtonImage	searchEdit1.clearButtonImage = "/css/images/bpDash.png";
result	
searchEdit Test  	

property	source
searchButtonImage	searchEdit1.searchButtonImage = "/css/images/validatebox_warning.png";
result	
searchEdit Test  	

property	source
buttonFocusable	searchEdit1.buttonFocusable = false;
result	
	

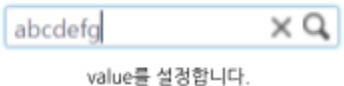
■ Methods

focus, getValue, unmaskedValue, setValue, clear, getEditComponent, search

method	source
focus	searchEdit1.focus();
result	
	

method	source
getValue	searchEdit1.getValue();
result	
현재 value 값을 가져옵니다.	

method	source
unmaskedValue	searchEdit1.unmaskedValue();
result	
마스킹을 제거한 value를 가져옵니다.	

method	source
setValue	searchEdit1.setValue("abcdefg");
result	
	

method	source
clear	searchEdit1.clear();
result	
서치인풋의 value를 빈 값으로 설정합니다.	

method	source
getEditComponent	searchEdit1.getEditComponent();
result	
searchEdit에 포함된 Y[wai] component인 edit 객체를 가져옵니다. 해당 객체로 edit component의 모든 속성과 메소드를 사용할 수 있습니다.	

method	source
search	searchEdit1.search();
result	
search 버튼을 누르는 이벤트가 발생합니다.	

■ Events

onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmouseup, onmouseover, onblur, onfocus, onchange, oncontextmenu, oncomplete, onincomplete, oncleared, onSearch

```
/* -----
* body source
* ----- */
onclick      : <searchEdit id="clickvt1" onclick="eventAll(event);"></searchEdit>
ondblclick    : <searchEdit id="dblclickvt1" ondblclick="eventAll(event);"></searchEdit>
onkeydown     : <searchEdit id="keydownvt1" onkeydown="eventAll(event);"></searchEdit>
onkeypress    : <searchEdit id="keypresvt1" onkeypress="eventAll(event);"></searchEdit>
onkeyup       : <searchEdit id="keyupevt1" onkeyup="eventAll(event);"></searchEdit>
onmousedown   : <searchEdit id="mousedownvt1"
onmousedown="eventAll(event);"></searchEdit>
onmouseup     : <searchEdit id="mouseupevt1" onmouseup="eventAll(event);"></searchEdit>
onmouseover   : <searchEdit id="mouseovervt1" onmouseover="eventAll(event);"></searchEdit>
onblur        : <searchEdit id="blurevt1" onblur="eventAll(event);"></searchEdit>
onfocus      : <searchEdit id="focusevt1" onfocus="eventAll(event);"></searchEdit>
onchange      : <searchEdit id="changevt1" onchange="eventAll(event);"></searchEdit>
oncontextmenu : <searchEdit id="ctmenuevt1" oncontextmenu="eventAll(event);"></searchEdit>

oncomplete    : <searchEdit id="oncompletevt1" oncomplete="eventAll(event);"></searchEdit>
onincomplete   : <searchEdit id="onincompletevt1"
onincomplete="eventAll(event);"></searchEdit>
oncleared      : <searchEdit id="oncleared1evt" oncleared="eventAll(event);"></searchEdit>

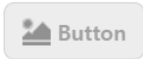
onSearch : <searchEdit id="onsearchvt1" onSearch="eventAll(event);"></searchEdit>
/* -----
* script source
* ----- */
function eventAll(obj){
    if (obj.event == 'onChange') {
        alert("Event : " + obj.event + "WnoldValue: " + obj.args.oldValue + "WnnewValue: " +
obj.args.newValue);
    } else {
        alert("Event : " + obj.event);
    }
}
```

6. Y[wai] Component 예제

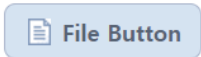
O. Image Button

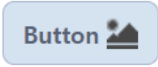
■ Properties

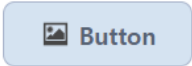
disabled, readonly, image, imageAlign, imageSize

property	source
disabled	<code>imgButton1.disabled = true;</code>
result	
	

property	source
readonly	<code>imgButton1.readonly = true;</code>
result	
	

property	source
image	<code>imgButton1.image = '/css/images/tree_file.png';</code>
result	
	

property	source
imageAlign	imgButton1.imageAlign = 'right';
result	
	

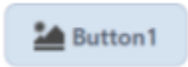
property	source
imageSize	imgButton1.imageSize = '13px';
result	
	

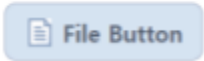
■ Methods

focus, getValue, unmaskedValue, setValue, clear, getEditComponent, search

method	source
focus	imgButton1.focus();
result	
focus를 이동합니다.	

method	source
click	imgButton1.click();
result	
버튼을 누르는 이벤트가 발생합니다.	

method	source
setText	imgButton1.setText('Button1');
result	
 버튼의 text 영역이 해당 값으로 변경됩니다.	

method	source
setImage	<code>imgButton1.setImage('/css/images/tree_file.png');</code>
result	
 버튼의 이미지 영역이 해당 url의 이미지 값으로 변경됩니다.	

■ Events

onfocus, onclick, ondblclick, onmousedown, onmouseover, onmouseup, onkeydown, onkeypress, onkeyup, oncontextmenu

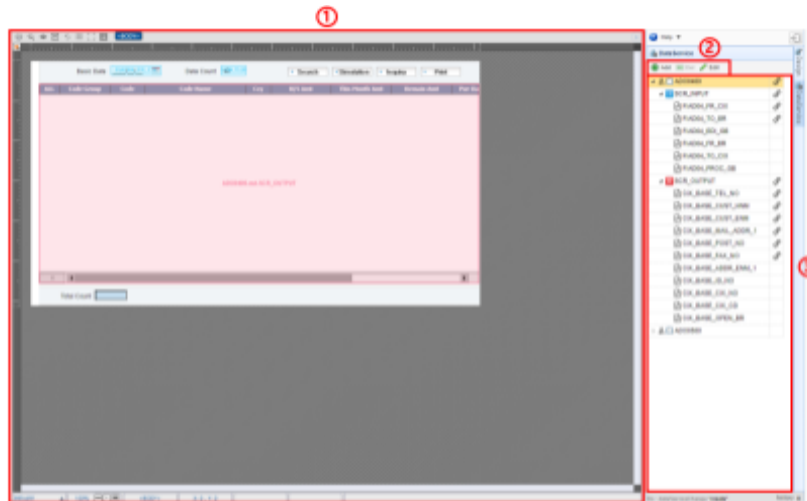
```
/* -----  
* body source  
* ----- */  
onclick      : <imgButton id="clikevt1" onclick="eventAll(event);"></imgButton>  
ondblclick   : <imgButton id="dblclikevt1" ondblclick="eventAll(event);"></imgButton>  
onkeydown    : <imgButton id="keydownvt1" onkeydown="eventAll(event);"></imgButton>  
onkeypress   : <imgButton id="keypressevt1" onkeypress="eventAll(event);"></imgButton>  
onkeyup      : <imgButton id="keyupevt1" onkeyup="eventAll(event);"></imgButton>  
onmousedown  : <imgButton id="mousedownvt1"  
onmousedown="eventAll(event);"></imgButton>  
onmouseup    : <imgButton id="mouseupevt1" onmouseup="eventAll(event);"></imgButton>  
onmouseover  : <imgButton id="mouseoverevt1" onmouseover="eventAll(event);"></imgButton>  
onfocus     : <imgButton id="focusevt1" onfocus="eventAll(event);"></imgButton>  
oncontextmenu : <imgButton id="ctmenuvt1" oncontextmenu="eventAll(event);"></imgButton>  
  
/* -----  
* script source  
* ----- */  
function eventAll(obj){  
    if (obj.event == onclick ) {  
        alert("Event : " + obj.event);  
    } else {  
        alert("Event : " + obj.event);  
    }  
}
```

7. waiServer 연동 기능

A. DataService Mode

waiServer의 Data I/O를 이용한 서비스 호출 기능입니다.

■ 메인 화면



메인 화면

① Object Status

우측 DataService 리스트에서 행 선택 시 DataField에 바인딩된 컴포넌트 객체를 표시합니다.
입력 Dataset에 포함된 필드는 파란색, 출력 Dataset에 포함된 필드는 빨간색으로 표시됩니다.

② Edit Tool

DataService를 추가, 삭제, 편집하는 도구입니다.

- ✓ Add : DataService 선택 팝업창을 호출하여 DataService를 추가합니다.
- ✓ Del : DataService 리스트에 체크된 DataService를 삭제합니다.
- ✓ Edit : DataService 관리 팝업창을 호출하여 DataService를 생성, 수정, 삭제합니다.

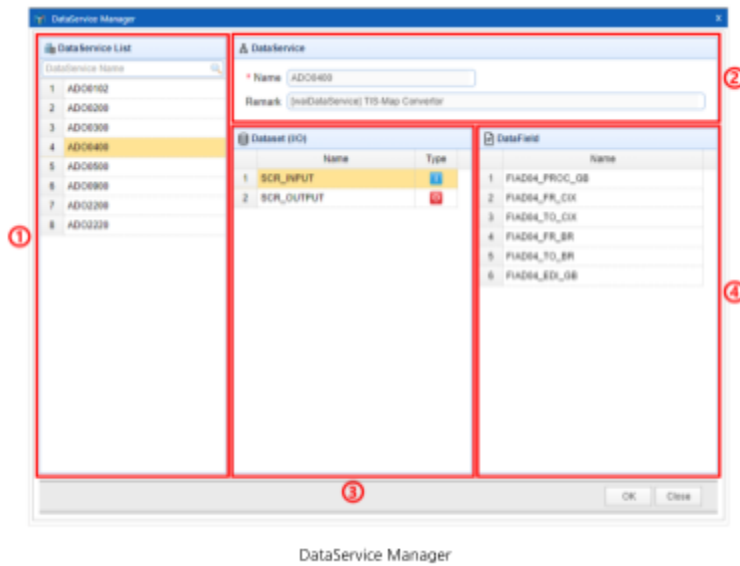
③ DataService List

현재 작업하고 있는 html 파일에 추가된 DataService 목록을 표시합니다.

Tree 구조로 표현되며 DataService - Dataset - DataField로 구성됩니다.

우측에 위치한 아이콘은 바인딩 여부를 표시합니다.

■ DataService Manager (Select)



① DataService List

waiServer에 등록된 DataService 목록이 조회됩니다.

상단에 위치한 검색창에 DataService명을 입력하여 특정 DataService를 조회할 수 있습니다.

목록에서 DataService를 선택하고 [OK] 버튼을 눌러 DataService를 추가합니다.

② DataService Information

DataService List에서 선택한 DataService의 정보가 표시됩니다.

③ Dataset List

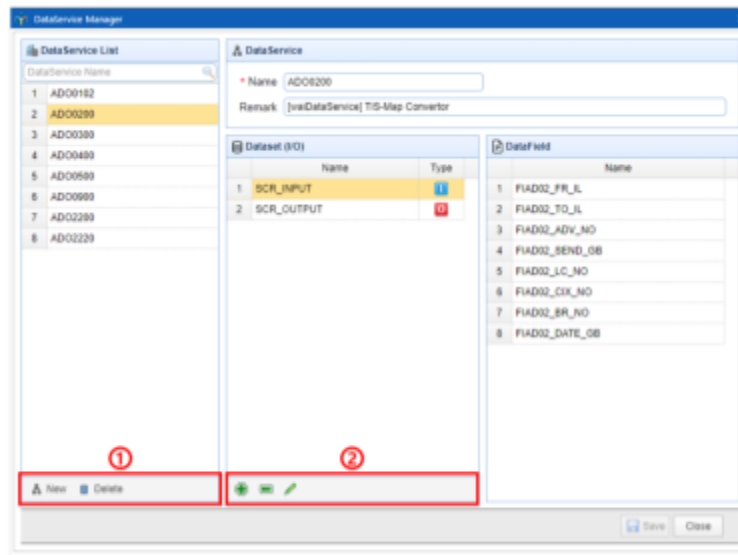
선택한 DataService의 Dataset 목록이 조회됩니다.

우측에 위치한 아이콘은 입출력 구분을 표시합니다.

④ DataField List

선택한 Dataset의 DataField 목록이 조회됩니다.

■ DataService Manager (Manage)



DataService Manager

① DataService Edit Tool

waiServer에 DataService를 등록, 삭제하는 도구입니다.

✓ New : DataService를 waiServer에 신규로 등록합니다.

DataService 정보를 입력하고 Dataset을 추가한 뒤 [Save] 버튼을 눌러 등록합니다.

✓ Delete : 목록에서 선택한 DataService를 waiServer에서 삭제합니다.

② Dataset Edit Tool

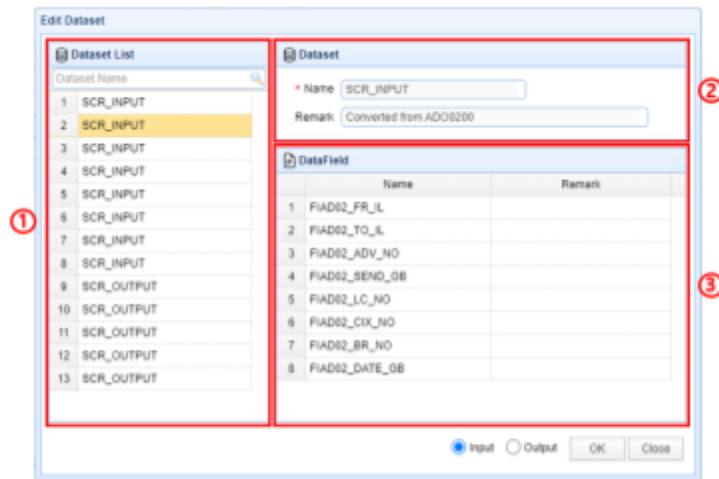
Dataset을 추가, 삭제, 편집하는 도구입니다.

✓ Add : Dataset 선택 팝업창을 호출하여 Dataset을 추가합니다.

✓ Del : Dataset List에서 선택한 Dataset을 삭제합니다.

✓ Edit : Dataset 관리 팝업창을 호출하여 Dataset을 생성, 수정, 삭제합니다.

■ Dataset Manager (Select)



Dataset Manager

① Dataset List

waiServer에 등록된 Dataset 목록이 조회됩니다.

상단에 위치한 검색창에 Dataset명을 입력하여 특정 Dataset을 조회할 수 있습니다.

목록에서 Dataset을 선택하고 Input 또는 Output 체크박스에 체크합니다.

[OK] 버튼을 눌러 Dataset을 추가합니다.

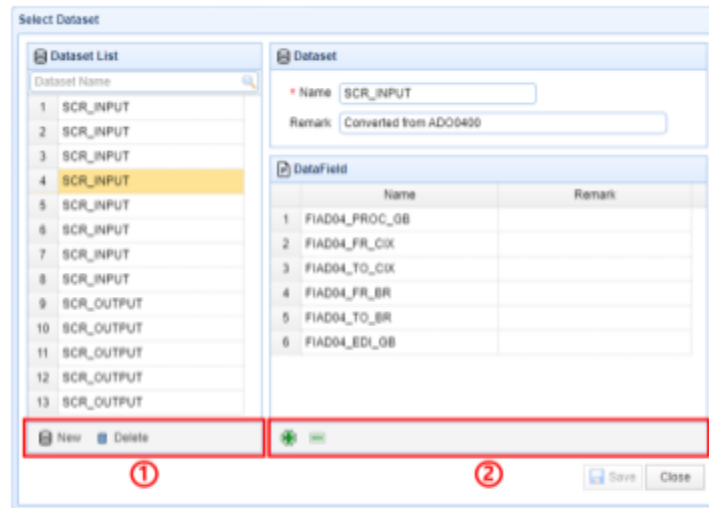
② Dataset Information

Dataset List에서 선택한 Dataset의 정보가 표시됩니다.

③ DataField List

선택한 Dataset의 DataField 목록이 조회됩니다.

■ Dataset Manager (Manage)



Dataset Manager

① Dataset Edit Tool

waiServer에 Dataset을 등록 및 삭제하는 도구입니다.

✓ New : Dataset을 waiServer에 신규로 등록합니다.

Dataset 정보를 입력하고 DataField를 추가한 뒤 [Save] 버튼을 눌러 등록합니다.

✓ Delete : 목록에서 선택한 Dataset을 waiServer에서 삭제합니다.

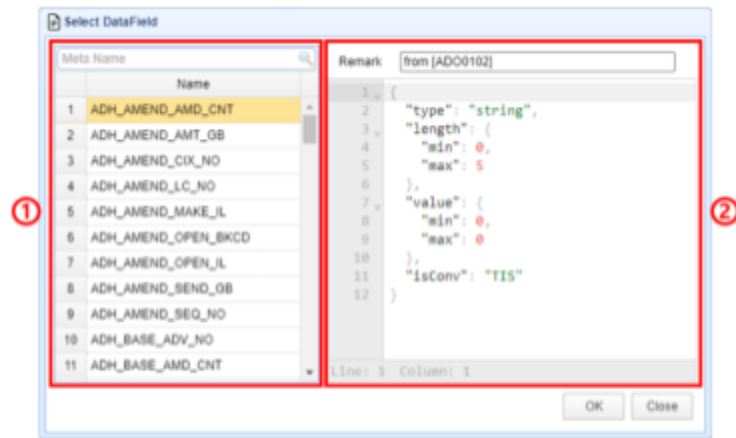
② DataField Edit Tool

DataField를 추가 및 삭제하는 도구입니다.

✓ Add : DataField 선택 팝업창을 호출하여 DataField를 추가합니다.

✓ Del : DataField List에서 선택한 DataField를 삭제합니다.

■ DataField Manager (Select)



DataField Manager

① DataField(Data Meta) List

waiServer에 등록된 DataField(Data Meta) 목록이 조회됩니다.

상단에 위치한 검색창에 DataField(Data Meta)명을 입력하여 특정 DataField(Data Meta)를 조회할 수 있습니다.

목록에서 DataField(Data Meta)를 선택하고 [OK] 버튼을 눌러 DataField(Data Meta)를 추가합니다.

② DataField(Data Meta) Information

DataField(Data Meta) List에서 선택한 DataField(Data Meta)의 정보가 표시됩니다.

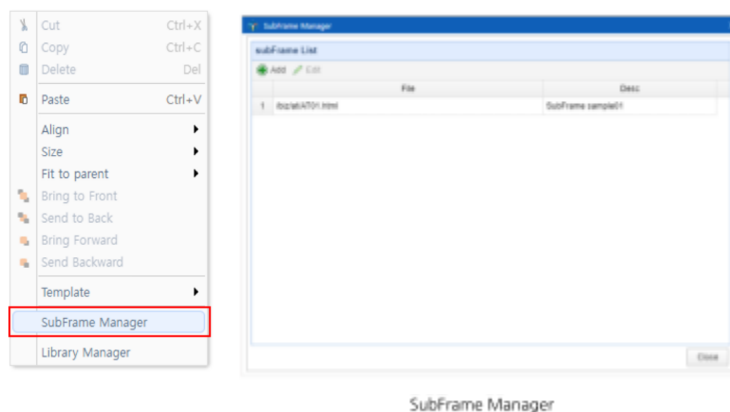
7. waiServer 연동 기능

B. SubFrame

컴포넌트 내부에 다른 HTML 페이지를 호출하는 기능입니다.

SubFrame Manager를 이용하여 HTML페이지를 등록하고 SubFrame 컴포넌트에서 호출할 수 있습니다.

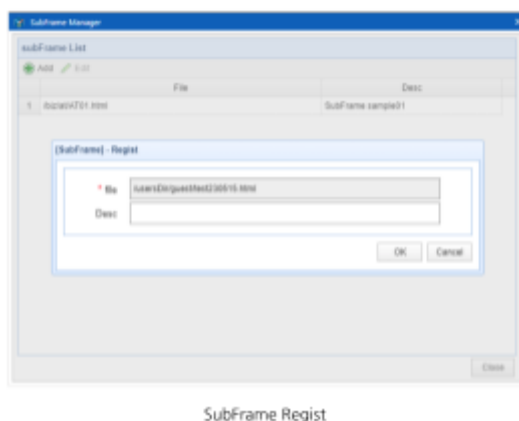
■ SubFrame 등록



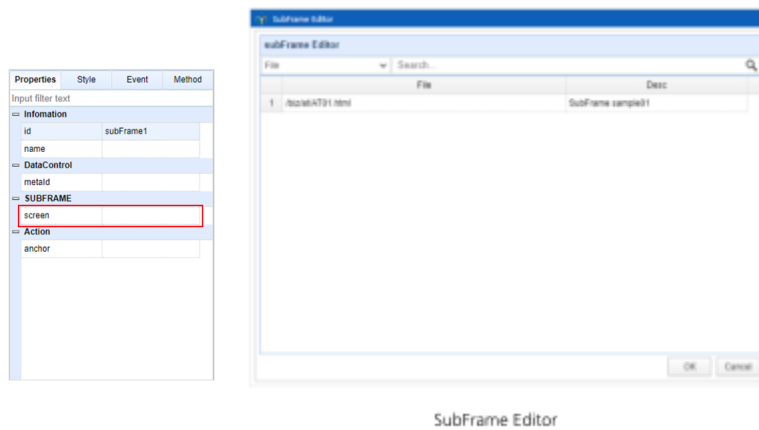
디자이너의 최상위 영역에 마우스 우클릭을 하면 우클릭 메뉴에 SubFrame Manager 메뉴가 활성화됩니다.

SubFrame Manager에서 [Add] 버튼을 클릭하면 SubFrame 등록 팝업창이 호출됩니다.

현재 작업 중인 파일이 등록 대상이 됩니다. 등록된 정보는 waiServer에서 확인할 수 있습니다.



■ SubFrame 호출



디자이너에 SubFrame 컴포넌트를 생성하고 Property Editor에서 screen 속성을 편집하여 SubFrame Editor 팝업창을 호출합니다.

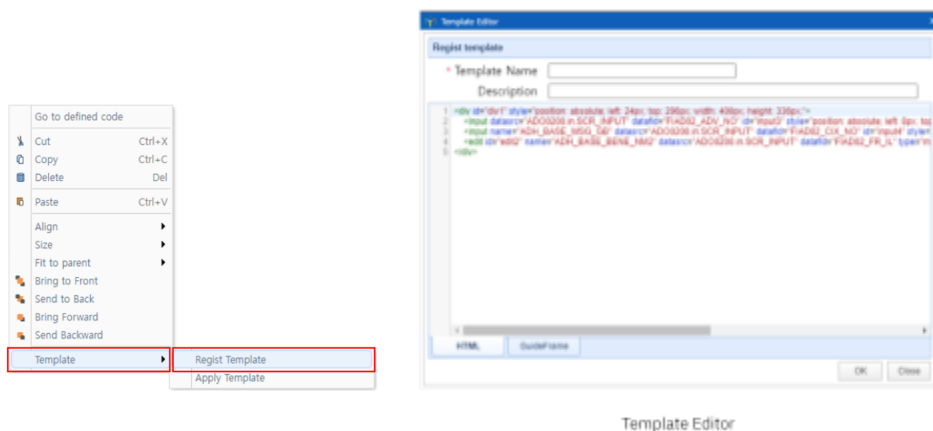
호출하려는 HTML 페이지를 선택하고 [OK] 버튼을 누르면 해당 HTML 페이지가 SubFrame 컴포넌트 내부에 호출된 것을 확인할 수 있습니다.

7. waiServer 연동 기능

C. Template

자주 사용하는 소스 코드를 Template으로 등록해서 사용하는 기능입니다.

■ Template 등록

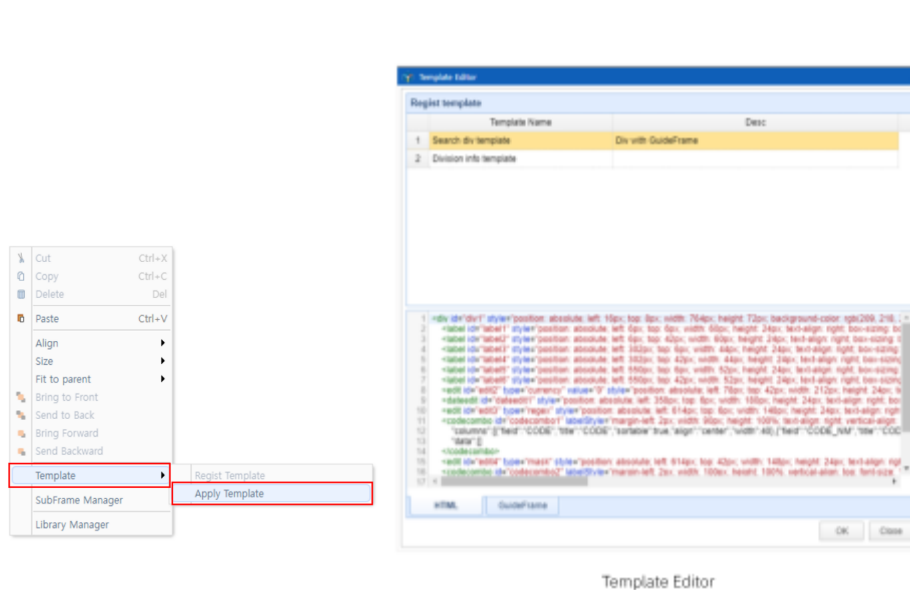


Template으로 등록하고자 하는 컴포넌트에 마우스 우클릭을 하면 Template 메뉴가 활성화됩니다.
Regist Template 메뉴를 클릭하여 Template 등록 팝업창을 호출합니다.

등록할 컴포넌트의 자식 컴포넌트를 포함한 전체 소스가 등록 대상이 됩니다. GuideFrame이 적용된 컴포넌트의 경우 하단 GuideFrame 탭에 GuideFrame 정보가 함께 포함됩니다.

정보를 입력하고 [OK] 버튼을 클릭하여 등록합니다. 등록된 정보는 waiServer에서 확인할 수 있습니다.

■ Template 호출



디자이너에서 Template을 불러올 위치에 우클릭을 합니다.

Template 메뉴에서 Apply Template 메뉴를 선택하면 Template 선택창이 호출됩니다.

(Apply Template 메뉴는 자식 컴포넌트를 가질 수 있는 위치에서만 활성화됩니다.)

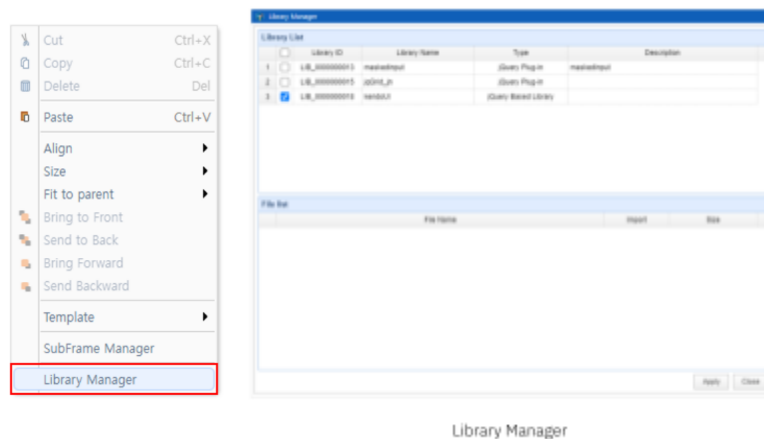
Template을 선택하고 [OK] 버튼을 클릭하여 Template을 불러옵니다.

(화면에 존재하는 ID가 Template에 포함되어 중복되는 경우에는 Template을 불러올 수 없습니다.)

7. waiServer 연동 기능

D. 라이브러리 호출

■ 적용 방법



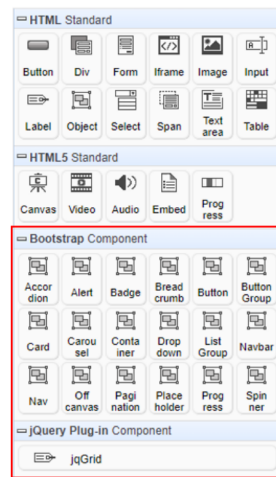
디자이너의 최상위 영역에 마우스 우클릭을 하면 우클릭 메뉴에 Library Manager 메뉴가 활성화됩니다.

Library Manager 팝업창에서 waiServer에 등록된 라이브러리들이 조회됩니다.

적용하려는 라이브러리를 그리드 좌측 체크박스에 체크하고 [Apply] 버튼을 클릭하여 적용합니다.
소스 코드에 라이브러리 정보가 입력된 것을 확인할 수 있습니다.

```
<head>
  <meta charset="UTF-8">
  <script type="text/javascript" src="/wai/wai.js">
    {
      { "type": "library", "id": "LIB_0000000002", "name": "jqGrid-5.7.0-Trial" }
    }
  </script>
  <link href="/wai/default.css" rel="stylesheet" type="text/css">
</head>
```

■ 동적 컴포넌트



waiServer에서 라이브러리를 등록할 때 jQuery Plug-in 또는 jQuery Based Library 타입으로 컴포넌트 목록을 구성한 경우 라이브러리를 적용하면 컴포넌트 팔레트에 해당 컴포넌트 목록이 나타납니다.

디자이너에 컴포넌트를 그리면 각 컴포넌트별로 구성된 정보를 바탕으로 컴포넌트가 추가됩니다.

8. 고객 지원 및 제품 공급자 정보

A. 고객 지원 정보

■ 제품 및 기술 문의

✓ (주)코아뱅크 기술연구소

✓ Tel: 070-4774-5391

8. 고객 지원 및 제품 공급자 정보

B. 제품 공급자 정보

■ 공급자 정보

✓ 회사명

(주)코아뱅크

✓ 주소

서울시 송파구 백제고분로 63 위너스빌딩 A동 610호

✓ 대표 번호

02-425-6200

✓ Fax

02-425-6209

✓ 홈페이지

<https://www.corebank.co.kr/corebank/index.html>

✓ 체험 사이트

<https://uitool.org/>