

# Tree

Kind: global class

Requires: module:OG.\*

Author: [Seungpil Park \(mailto:sppark@uengine.org\)](mailto:sppark@uengine.org)

- [Tree](#)
  - [new Tree\(container\)](#)
  - [.init\(\)](#)
  - [.getScale\(\)](#) ⇒ Number
  - [.setScale\(scale\)](#)
  - [.setShowLabel\(show\)](#)
  - [.drawArea\(\)](#)
  - [.clear\(\)](#)
  - [.loadViewData\(\)](#) ⇒ Array
  - [.load\(\)](#) ⇒ Array
  - [.loadByFilter\(filterData\)](#) ⇒ Array
  - [.removeDataByFilter\(filterData\)](#)
  - [.clearData\(preventRender\)](#)
  - [.sortData\(prop, positions, desc, preventRender\)](#)
  - [.updateData\(data, preventRender\)](#)
  - [.render\(\)](#)
  - [.createViewData\(\)](#) ⇒ Object
    - [~getViewData\(object, depth, parentView, childFromParent\)](#)
  - [.createStandaloneViewData\(mapping, targetActivityView\)](#) ⇒ Object
  - [.renderViews\(\)](#)
  - [.labelSubstring\(label\)](#) ⇒ String
  - [.updateImageShapeStatus\(view, element\)](#)
    - [~applyPathStyle\(\\$svg, color, stroke\)](#)
  - [.drawMappingLabel\(view\)](#)
  - [.updateMappingLabel\(view, element, customData\)](#)
  - [.updateActivity\(view, element\)](#)
  - [.drawActivity\(view\)](#)
  - [.updateFolder\(view, element\)](#)
  - [.drawFolder\(view\)](#)
  - [.updateEd\(view, element\)](#)
  - [.drawEd\(view\)](#)
  - [.drawMappingLine\(view\)](#)
  - [.updateExpanderLine\(view, element\)](#)
  - [.drawExpanderLine\(view\)](#)
  - [.updateActivityRelLine\(view, element\)](#)
  - [.drawActivityRelLine\(view\)](#)
  - [.updateExpander\(view, element\)](#)
  - [.drawExpander\(view\)](#)
  - [.getExpanderCenterX\(position, depth, standardX\)](#) ⇒ Number
  - [.getShapeCenterX\(position, depth, standardX\)](#) ⇒ Number

- [.getMappingEdgeVertices\(depth, parentY, myY, pStandardX, myStandardX, hasChild\)](#) ⇒ Array
- [.getActivityRelVertices\(position, depth, standardX, parentY, myY\)](#) ⇒ Array
- [.getExpanderToVertices\(position, depth, standardX, parentY, myY\)](#) ⇒ Array
- [.getExpanderFromVertices\(position, depth, standardX, parentY, myY\)](#) ⇒ Array
- [.dividedViewsByPosition\(displayViews\)](#) ⇒ Object
- [.reRangeAreaSize\(viewData\)](#)
- [\[.fitToBoundary\(element, offsetupper,low,left,right\)](#) ⇒ element
- [.selectActivityByPosition\(position\)](#) ⇒ Array
- [.selectNextActivity\(id\)](#) ⇒ Object
- [.selectPrevActivity\(id\)](#) ⇒ Object
- [.selectNextActivities\(id\)](#) ⇒ Array
- [.selectChildById\(id\)](#) ⇒ Array
- [.selectChildMapping\(sourceld, targetId\)](#) ⇒ Array
- [.selectRecursiveChildMapping\(sourceld, targetId\)](#) ⇒ Array
- [.selectParentById\(id\)](#) ⇒ Object
- [.selectParentMapping\(sourceld, targetId\)](#) ⇒ Object
- [.selectById\(id\)](#) ⇒ Object
- [.selectBySourceTarget\(sourceld, targetId\)](#) ⇒ Object
- [.selectMappings\(\)](#) ⇒ Array
- [.selectRootActivityById\(id\)](#) ⇒ Object
- [.selectRootMapping\(sourceld, targetId\)](#) ⇒ Object
- [.selectRecursiveParentById\(id\)](#) ⇒ Array
- [.selectRecursiveChildById\(id\)](#) ⇒ Array
- [.selectRecursiveLastChildById\(id\)](#) ⇒ Array
- [.selectViewById\(viewData, id\)](#) ⇒ Object
- [.selectViewByFilter\(viewData, filterData\)](#) ⇒ Array
- [.selectRecursiveChildViewsById\(viewData, id\)](#) ⇒ Array
- [.selectMaxyFromViews\(views\)](#) ⇒ number
- [.selectMaxDepthFromViews\(views\)](#) ⇒ number
- [.selectMaxBottomFromViews\(views\)](#) ⇒ number
- [.emptyString\(value\)](#) ⇒ boolean
- [.getElementByPoint\(point\)](#) ⇒ Element
- [.uuid\(\)](#) ⇒ string
- [.bindEvent\(\)](#)
- [.bindTooltip\(element\)](#)
- [.bindDbClickEvent\(element\)](#)
- [.bindMappingHighLight\(element\)](#)
- [.bindActivityMove\(\)](#)
- [.onBeforeActivityMove\(activities\)](#)
- [.onActivityMove\(activities\)](#)
- [.bindMappingEvent\(\)](#)
- [.deleteMapping\(data, view\)](#)
- [.enableShapeContextMenu\(\)](#)
- [.makeShowProperties\(\)](#) ⇒ Object
- [.makeFolder\(\)](#) ⇒ Object
- [.makeEd\(\)](#) ⇒ Object
- [.makePickEd\(\)](#) ⇒ Object
- [.makeDelete\(\)](#) ⇒ Object

- [.makeListRelation\(\)](#) ⇒ Object
- [.makeDeleteRelation\(\)](#) ⇒ Object
- [.onBeforeMapping\(source, target, selectedTargetList\)](#) ⇒ boolean
- [.onMapping\(source, target, selectedTargetList\)](#) ⇒ boolean
- [.onBeforeDeleteMapping\(sourceId, sourceType, targetId, targetType\)](#) ⇒ boolean
- [.onDeleteMapping\(sourceId, sourceType, targetId, targetType\)](#) ⇒ boolean

## new Tree(container)

Open graph Tree Library (OG-Tree)

Param	Type	Description
container	String	Dom Element Id

## tree.init()

캔버스를 초기 빌드한다. 최초 1번만 실행된다.

**Kind:** instance method of [Tree](#)

## tree.getScale() ⇒ Number

Scale 을 반환한다. (리얼 사이즈 : Scale = 1)

**Kind:** instance method of [Tree](#)

**Returns:** Number – 스케일값

## tree.setScale(scale)

Scale 을 설정한다. (기본 사이즈 : Scale = 1)

**Kind:** instance method of [Tree](#)

Param	Type	Description
scale	Number	스케일값

## tree.setShowLabel(show)

라벨을 숨김/ 보임 처리한다.

**Kind:** instance method of [Tree](#)

Param	Description
show	보임 여부

## tree.drawArea()

기본 Area 를 생성한다.

lAc,lOut,rIn,rAc,rOut,Canvas

**Kind:** instance method of [Tree](#)

## tree.clear()

캔버스의 모든 화면요소를 삭제한다.

**Kind:** instance method of [Tree](#)

**tree.loadViewData()** ⇒ Array

뷰 데이터를 불러온다.

**Kind:** instance method of [Tree](#)

**Returns:** Array - OG-Tree view data

**tree.load()** ⇒ Array

노드 데이터를 불러온다.

**Kind:** instance method of [Tree](#)

**Returns:** Array - OG-Tree data

**tree.loadByFilter(filterData)** ⇒ Array

노드 데이터를 필터링하여 불러온다.

**Kind:** instance method of [Tree](#)

**Returns:** Array - OG-Tree data

Param	Description
-------	-------------

filterData	json
------------	------

**tree.removeDataByFilter(filterData)**

노드 데이터를 필터링 하여 삭제한다.

**Kind:** instance method of [Tree](#)

Param	Description
-------	-------------

filterData	json
------------	------

**tree.clearData(preventRender)**

노드 데이터를 모두 삭제한다.

**Kind:** instance method of [Tree](#)

Param	Description
-------	-------------

preventRender	화면 리로드 여부
---------------	-----------

**tree.sortData(prop, positions, desc, preventRender)**

트리의 데이터를 주어진 prop 로 sorts한다.

**Kind:** instance method of [Tree](#)

Param	Description
-------	-------------

prop	sort 키
------	--------

positions	Array of Area position
desc	역순 여부
preventRender	화면 리로드 여부

## **tree.updateData(data, preventRender)**

데이터를 업데이트한다.

**Kind:** instance method of [Tree](#)

Param	Description
data	OG-Tree data
preventRender	화면 리로드 여부

## **tree.render()**

스토리지의 데이터를 기반으로 화면에 렌더링한다.

**Kind:** instance method of [Tree](#)

## **tree.createViewData() ⇒ Object**

스토리지의 데이터를 기반으로 화면에 표현되어야 하는 각 객체의 y 좌표를 생성한 ViewData 를 반환한다.

**Kind:** instance method of [Tree](#)

## **createViewData~getViewData(object, depth, parentView, childFromParent)**

주어진 객체의 좌표를 생성하여 viewData 에 저장하고, 객체에 자식이 있다면 함수를 반복수행한다.

**Kind:** inner method of [createViewData](#)

Param
object
depth
parentView
childFromParent

## **tree.createStandaloneViewData(mapping, targetActivityView) ⇒ Object**

매핑 시킬 아더워크플로우가 없는 인 데이터들로 viewData 를 구성한다.

**Kind:** instance method of [Tree](#)

Param	Description
mapping	OG-Tree data
targetActivityView	OG-Tree view data

## **tree.renderViews()**

viewData 중에서 실제로 화면에 표현되어야 할 객체를 선정하고 각 x 좌표를 책정한다.  
선정된 객체들을 화면에 드로잉한다.

**Kind:** instance method of [Tree](#)

## **tree.labelSubstring(label) ⇒ String**

주어진 라벨이 최대 표기 숫자를 넘길 경우 텍스트를 줄인다.

**Kind:** instance method of [Tree](#)

**Returns:** String – fixed label

### **Param Description**

label    라벨

## **tree.updateImageShapeStatus(view, element)**

이미지 Shape 의 컬러와 스트로크를 스테이터스에 따라 변경한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data
element	OG-Tree Dom Element

## **updateImageShapeStatus~applyPathStyle(\$svg, color, stroke)**

svg 의 path 들에 컬러와 stroke 를 적용시킨다.

**Kind:** inner method of [updateImageShapeStatus](#)

Param
\$svg
color
stroke

## **tree.drawMappingLabel(view)**

매핑시 선택된 된 아이템에 S 마크를 붙인다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data

## **tree.updateMappingLabel(view, element, customData)**

매핑시 선택된 된 아이템의 S 마크를 업데이트 한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data
element	OG-Tree Dom Element
customData	OG-Tree data

## **tree.updateActivity(view, element)**

액티비티 아이템을 업데이트 한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data
element	OG-Tree Dom Element

## **tree.drawActivity(view)**

액티비티 아이템을 드로잉한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data

## **tree.updateFolder(view, element)**

폴더 아이템을 업데이트한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data
element	OG-Tree Dom Element

## **tree.drawFolder(view)**

폴더 아이템을 드로잉한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data

## **tree.updateEd(view, element)**

ED 아이템을 업데이트 한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data
element	OG-Tree Dom Element

## **tree.drawEd(view)**

ED 아이템을 드로잉한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data

## **tree.drawMappingLine(view)**

매핑 연결선을 드로잉한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data

## **tree.updateExpanderLine(view, element)**

expander 선연결을 업데이트한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data
element	OG-Tree Dom Element

## **tree.drawExpanderLine(view)**

expander 선연결을 생성한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data

## **tree.updateActivityRelLine(view, element)**

액티비티간의 연결선을 업데이트한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data
element	OG-Tree Dom Element

## **tree.drawActivityRelLine(view)**

액티비티간의 연결선을 드로잉한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data

## **tree.updateExpander(view, element)**

expander 를 업데이트한다.



**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data
element	OG-Tree Dom Element

### **tree.drawExpander(view)**

expander 를 드로잉한다.

**Kind:** instance method of [Tree](#)

Param	Description
view	OG-Tree view data

### **tree.getExpanderCenterX(position, depth, standardX) ⇒ Number**

expander 의 센터를 구한다.

**Kind:** instance method of [Tree](#)

**Returns:** Number – center X 좌표

Param	Description
position	Area position
depth	아이템 depth
standardX	Area X 좌표

### **tree.getShapeCenterX(position, depth, standardX) ⇒ Number**

액티비티, 폴더, Ed 의 센터를 구한다.

**Kind:** instance method of [Tree](#)

**Returns:** Number – center X 좌표

Param	Description
position	Area position
depth	아이템 depth
standardX	Area X 좌표

### **tree.getMappingEdgeVertices(depth, parentY, myY, pStandardX, myStandardX, hasChild) ⇒ Array**

매핑 연결선의 vertices 를 구한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – vertices

Param	Description
depth	아이템 depth
parentY	매핑 대상 액티비티 Y 좌표
myY	자신의 Y 좌표
pStandardX	매핑 대상 액티비티 Area X 좌표

myStandardX 자신의 Area X 좌표  
hasChild      자식이 있는지 여부

**tree.getActivityRelVertices(position, depth, standardX, parentY, myY) ⇒ Array**

액티비티간의 연결선의 vertices 를 구한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – vertices

Param	Description
position	Area position
depth	아이템 depth
standardX	Area X 좌표
parentY	연결대상 액티비티 Y 좌표
myY	자신의 Y 좌표

**tree.getExpanderToVertices(position, depth, standardX, parentY, myY) ⇒ Array**

Expander To 선의 vertices 를 구한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – vertices

Param	Description
position	Area position
depth	아이템 depth
standardX	Area X 좌표
parentY	부모 아이템의 Y 좌표
myY	자신의 Y 좌표

**tree.getExpanderFromVertices(position, depth, standardX, parentY, myY) ⇒ Array**

Expander From 선의 vertices 를 구한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – vertices

Param	Description
position	Area position
depth	아이템 depth
standardX	Area X 좌표
parentY	부모 아이템의 Y 좌표
myY	자신의 Y 좌표

**tree.dividedViewsByPosition(displayViews) ⇒ Object**

주어진 views 를 포지션별로 분류하여 리턴한다.

**Kind:** instance method of [Tree](#)

**Returns:** Object – HashMap of OG-Tree view data

Param	Description
displayViews	Array of OG-Tree view data

### **tree.reRangeAreaSize(viewData)**

각 Area 의 크기를 책정하고 redraw 한다.  
캔버스의 사이즈를 재조정한다.

**Kind:** instance method of [Tree](#)

Param	Description
viewData	HashMap of OG-Tree view data

### **tree.fitToBoundary(element, offset[upper,low,left,right]) ⇒ element**

주어진 Boundary 영역 안으로 공간 기하 객체를 적용한다.(이동 & 리사이즈)

**Kind:** instance method of [Tree](#)

**Returns:** element – OG-Tree Dom Element

Param	Description
element	OG-Tree Dom Element
offset[upper,low,left,right]	

### **tree.selectActivityByPosition(position) ⇒ Array**

주어진 에어리어에 해당하는 액티비티 정보를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree data

Param	Description
position	Area position

### **tree.selectNextActivity(id) ⇒ Object**

주어진 id 의 액티비티의 next 액티비티를 구한다.

**Kind:** instance method of [Tree](#)

**Returns:** Object – OG-Tree data

Param	Description
id	액티비티 id

### **tree.selectPrevActivity(id) ⇒ Object**

주어진 id 의 prev 액티비티를 구한다.

**Kind:** instance method of [Tree](#)

**Returns:** Object – OG-Tree data

#### Param Description

id      액티비티 id

### **tree.selectNextActivities(id) ⇒ Array**

주어진 id 의 next 액티비티들을 구한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree data

#### Param Description

id      액티비티 id

### **tree.selectChildById(id) ⇒ Array**

주어진 아이디의 자식 데이터를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree data

#### Param Description

id      OG-Tree data id

### **tree.selectChildMapping(sourceId, targetId) ⇒ Array**

주어진 소스와 타겟 아이디를 가지는 매핑 데이터의 자식을 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree data

#### Param Description

sourceId OG-Tree data id

targetId OG-Tree data id

### **tree.selectRecursiveChildMapping(sourceId, targetId) ⇒ Array**

주어진 소스와 타겟 아이디를 가지는 매핑 데이터의 자식을 재귀호출하여 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree data

#### Param Description

sourceId OG-Tree data id

targetId OG-Tree data id

### **tree.selectParentById(id) ⇒ Object**

주어진 아이디의 부모정보를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Object – OG-Tree data

#### Param Description

id      OG-Tree data id

**tree.selectParentMapping(sourceId, targetId) ⇒ Object**

매핑 데이터의 부모 매핑 데이터를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Object – OG-Tree data

Param	Description
sourceId	OG-Tree data id
targetId	OG-Tree data id

**tree.selectById(id) ⇒ Object**

주어진 아이디의 정보를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Object – OG-Tree data

Param	Description
id	OG-Tree data id

**tree.selectBySourceTarget(sourceId, targetId) ⇒ Object**

주어진 소스아이디와 타겟아이디와 일치하는 OG-Tree 데이터를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Object – OG-Tree data

Param	Description
sourceId	OG-Tree data id
targetId	OG-Tree data id

**tree.selectMappings() ⇒ Array**

매핑 데이터를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree data

**tree.selectRootActivityById(id) ⇒ Object**

주어진 아이디의 루트 액티비티 정보를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Object – OG-Tree data

Param	Description
id	OG-Tree data id

**tree.selectRootMapping(sourceId, targetId) ⇒ Object**

매핑 데이터의 루트를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Object – OG-Tree data

Param	Description
sourceId	OG-Tree data id
targetId	OG-Tree data id

**tree.selectRecursiveParentById(id) ⇒ Array**

주어진 아이디의 부모 일람을 재귀호출하여 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree data

Param	Description
id	OG-Tree data id

**tree.selectRecursiveChildById(id) ⇒ Array**

주어진 아이디의 자식 데이터를 재귀호출하여 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree data

Param	Description
id	OG-Tree data id

**tree.selectRecursiveLastChildById(id) ⇒ Array**

주어진 아이디의 자식 데이터를 재귀호출하여, 더이상 자식이 없는 마지막 데이터일 경우의 리스트를 반환한다.  
(자기 자신이 마지막 데이터일 경우 자기 자신을 포함하여)

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree data

Param	Description
id	OG-Tree data id

**tree.selectViewById(viewData, id) ⇒ Object**

주어진 아이디에 해당하는 뷰 데이터를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Object – OG-Tree view data id

Param	Description
viewData	HashMap of OG-Tree view data
id	OG-Tree view data id

**tree.selectViewByFilter(viewData, filterData) ⇒ Array**

주어진 필터 조건에 따라 뷰데이터를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree view data

Param	Description
viewData	HashMap of OG-Tree view data
filterData	HashMap filter data

**tree.selectRecursiveChildViewsById(viewData, id) ⇒ Array**

주어진 아이디의 자식 뷰 데이터를 재귀호출하여 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Array – Array of OG-Tree view data

Param	Description
viewData	HashMap of OG-Tree view data
id	OG-Tree view data id

**tree.selectMaxyFromViews(views) ⇒ number**

주어진 views 중 가장 큰 y 를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** number – max Y

Param	Description
views	Array of OG-Tree view data

**tree.selectMaxDepthFromViews(views) ⇒ number**

주어진 views 중 가장 큰 depth 를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** number – max depth

Param	Description
views	Array of OG-Tree view data

**tree.selectMaxBottomFromViews(views) ⇒ number**

주어진 views 중 가장 큰 bottom 을 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** number – max bottom

Param	Description
views	Array of OG-Tree view data

**tree.emptyString(value) ⇒ boolean**

주어진 스트링이 빈값인지를 확인한다.

**Kind:** instance method of [Tree](#)

**Returns:** boolean - 빈 값 여부

**Param Description**

value String

**tree.getElementByPoint(point) ⇒ Element**

좌표값을 포함하는 가장 앞단의 엘리먼트를 반환한다.

**Kind:** instance method of [Tree](#)

**Returns:** Element - OG-Tree Dom Element

**Param Description**

point [x,y] 좌표

**tree.uuid() ⇒ string**

무작위 랜덤 아이디 생성

**Kind:** instance method of [Tree](#)

**Returns:** string - 랜덤 아이디

**tree.bindEvent()**

캔버스가 처음 렌더링 될 시 필요한 이벤트들을 바인딩한다.

**Kind:** instance method of [Tree](#)

**tree.bindTooltip(element)**

툴팁 이벤트를 바인딩한다.

**Kind:** instance method of [Tree](#)

Param	Description
-------	-------------

element	OG-Tree Dom Element
---------	---------------------

**tree.bindDbClickEvent(element)**

더블클릭 이벤트를 바인딩한다.

**Kind:** instance method of [Tree](#)

Param	Description
-------	-------------

element	OG-Tree Dom Element
---------	---------------------

**tree.bindMappingHighLight(element)**

매핑 연결선의 하이라이트 이벤트를 바인딩한다.

**Kind:** instance method of [Tree](#)

Param	Description
-------	-------------



element OG-Tree Dom Element

### **tree.bindActivityMove()**

액티비티의 이동 이벤트를 바인딩한다.

**Kind:** instance method of [Tree](#)

### **tree.onBeforeActivityMove(activities)**

액티비티가 이동되기 전 이벤트

**Kind:** instance method of [Tree](#)

Param	Description
activities	Array of OG-Tree data

### **tree.onActivityMove(activities)**

액티비티가 이동 된 후 이벤트

**Kind:** instance method of [Tree](#)

Param	Description
activities	Array of OG-Tree data

### **tree.bindMappingEvent()**

매핑이 이루어졌을 때의 이벤트를 처리한다.

**Kind:** instance method of [Tree](#)

### **tree.deleteMapping(data, view)**

매핑을 해제한다.

**Kind:** instance method of [Tree](#)

Param	Description
data	OG-Tree data
view	OG-Tree view

### **tree.enableShapeContextMenu()**

OG Tree Dom Element 에 마우스 우클릭 메뉴를 가능하게 한다.

**Kind:** instance method of [Tree](#)

### **tree.makeShowProperties() ⇒ Object**

프로퍼티 보기 컨텍스트 메뉴를 생성한다.

**Kind:** instance method of [Tree](#)

**tree.makeFolder() ⇒ Object**

폴더 생성 컨텍스트 메뉴를 생성한다.

**Kind:** instance method of [Tree](#)

**tree.makeEd() ⇒ Object**

ED 생성 컨텍스트 메뉴를 생성한다.

**Kind:** instance method of [Tree](#)

**tree.makePickEd() ⇒ Object**

Pick ED 컨텍스트 메뉴를 생성한다.

**Kind:** instance method of [Tree](#)

**tree.makeDelete() ⇒ Object**

삭제 컨텍스트 메뉴를 생성한다.

**Kind:** instance method of [Tree](#)

**tree.makeListRelation() ⇒ Object**

List Relation 컨텍스트 메뉴를 생성한다.

**Kind:** instance method of [Tree](#)

**tree.makeDeleteRelation() ⇒ Object**

매핑 삭제 컨텍스트 메뉴를 생성한다.

**Kind:** instance method of [Tree](#)

**tree.onBeforeMapping(source, target, selectedTargetList) ⇒ boolean**

매핑이 이루어지기 전 이벤트

**Kind:** instance method of [Tree](#)

Param	Description
source	OG-Tree data 드래그 한 대상
target	OG-Tree data 드랍 한 대상
selectedTargetList	Array of OG-Tree data 드래그 대상의 하위 요소들

**tree.onMapping(source, target, selectedTargetList) ⇒ boolean**

매핑이 이루어졌을 때의 이벤트

**Kind:** instance method of [Tree](#)

Param	Description
-------	-------------

source                    OG-Tree data 드래그 한 대상  
target                    OG-Tree data 드랍 한 대상  
selectedTargetList Array of OG-Tree data 드래그 대상의 하위 요소들

**tree.onBeforeDeleteMapping(sourceId, sourceType, targetId, targetType) ⇒ boolean**

매핑을 삭제하기 전 이벤트

**Kind:** instance method of [Tree](#)

Param	Description
sourceId	OG-Tree data id 드래그 한 대상
sourceType	"workflow","activity","folder","ed"
targetId	OG-Tree data id 드랍 한 대상
targetType	"workflow","activity","folder","ed"

**tree.onDeleteMapping(sourceId, sourceType, targetId, targetType) ⇒ boolean**

매핑을 삭제한 후 이벤트

**Kind:** instance method of [Tree](#)

Param	Description
sourceId	OG-Tree data id 드래그 한 대상
sourceType	"workflow","activity","folder","ed"
targetId	OG-Tree data id 드랍 한 대상
targetType	"workflow","activity","folder","ed"