

Aras

Kind: global class

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

- [Aras](#)
 - [new Aras\(tree\)](#)
 - *instance*
 - [.getWorkflowStructure](#)
 - [new getWorkflowStructure\(wf_id, inout\)](#)
 - [.tree](#) : Tree
 - [.aras](#) : aras
 - [.thisItem](#) : Object
 - [.wfId](#) : String
 - [.stdYN](#) : String
 - [.projectId](#) : String
 - [.inn](#) : null
 - [.prjMaxDepth](#) : number
 - [.stdMaxDepth](#) : number
 - [.TYPE](#) : Object
 - [.iExmL2jsobj\(node\)](#) ⇒ Object
 - [.getHtmlParameter\(val\)](#) ⇒ String
 - [.init\(\)](#)
 - [.createBody\(params\)](#) ⇒ string
 - [.applyMethod\(methodName, body\)](#) ⇒ Object
 - [.getUserIdentity\(\)](#) ⇒ String
 - [.getUserId\(\)](#) ⇒ String
 - [.getWorkflowData\(wf_id\)](#) ⇒ Object
 - [.getItemById\(type, id\)](#) ⇒ Object
 - [.getActivityStructure\(activity_id, folder_yn, folder_id, inout\)](#) ⇒ Object
 - [.getEdParentList\(id, ed_yn\)](#) ⇒ Array
 - [.getPickEd\(ed_number, ed_name\)](#) ⇒ Array
 - [.nodeToJson\(xmlNode\)](#) ⇒ Object
 - [.getSchCombo\(kind, discLine, discSpec, bq, REL_WF_ID, callback\)](#)
 - [.getCurrentItemId\(itemType, id\)](#) ⇒ Object
 - [.getItemType\(type\)](#) ⇒ String
 - [.getRelType\(sourceType, targetType, inout\)](#) ⇒ String
 - [.showPropertyWindow\(type, id\)](#)
 - [.sortActivities\(activityIds\)](#)
 - [.checkMaxCreateNumber\(depth\)](#) ⇒ boolean
 - [.createFolder\(data, view\)](#)
 - [.addFolderOutRelation\(parentData, parentView, newItem, parentItem, parentType, parentId\)](#)
 - [.createEd\(data, view, edType\)](#)
 - [.addFolderEDOutRelation\(edItem, parentItem, data, view\)](#)
 - [.addPickEDOutRelation\(edItems, parentItem, data, view\)](#)

- [.deleteOutItem\(data, view\)](#)
 - [.createActivity\(\)](#)
 - [.addInRel\(source, target, selectedTargetList\)](#)
 - [.deleteInRel\(sourceId, sourceType, targetId, targetType\)](#)
 - [.currentSortOrder\(\)](#) ⇒ Object
 - [.createWorkFlowData\(resultNodeList, who, inout\)](#) ⇒ Array
 - [~getStateColorAndStroke\(type, state, isDelay\)](#) ⇒ Object
 - [~convertDate\(dateStr\)](#) ⇒ *
 - [~checkDelay\(node\)](#) ⇒ boolean
 - [.createMyWorkFlowData\(resultNodeList, inout\)](#) ⇒ Array
 - [.createOtherWorkFlowData\(resultNodeList\)](#) ⇒ Array
 - [.refreshOtherWorkflow\(wfId\)](#)
 - [.refreshMyWorkFlow\(\)](#)
 - [.refreshOutFolder\(data, view\)](#)
 - [.syncExpandDataWithTree\(data\)](#)
- *inner*
- [~state|son](#)

new Aras(tree)

Aras data Handler

Param	Type	Description
tree	Tree	오픈그래프 트리 라이브러리

aras.getWorkflowStructure

Kind: instance class of [Aras](#)

new getWorkflowStructure(wf_id, inout)

WF 하위의 액티비티, 폴더 및 ED 조회

Returns: Object – Aras Item

Param Description

wf_id
inout IN/OUT

aras.tree : Tree

OG-Tree 객체

Kind: instance property of [Aras](#)

aras.aras : aras

부모페이지의 aras 객체

Kind: instance property of [Aras](#)

aras.thisItem : Object

현재의 워크플로우 아이템

Kind: instance property of [Aras](#)

aras.wfld : String

현재의 워크플로우 아이디

Kind: instance property of [Aras](#)

aras.stdYN : String

현재 워크플로우의 스탠다드 모드 여부

Kind: instance property of [Aras](#)

aras.projectId : String

현재 워크플로우의 프로젝트 아이디

Kind: instance property of [Aras](#)

aras.inn : null

newIOMInnovator 리턴 객체

Kind: instance property of [Aras](#)

aras.prjMaxDepth : number

프로젝트, 폴더 5레벨 까지 enable, ed 는 6레벨까지 가능

Kind: instance property of [Aras](#)

aras.stdMaxDepth : number

스탠다드, 폴더 3레벨 까지 enable, 3레벨 일 경우는 ed 생성 불가

Kind: instance property of [Aras](#)

aras.TYPE : Object

아이템 타입 Contants

Kind: instance property of [Aras](#)

aras.iExmL2jsobj(node) ⇒ Object

IE 에서 아라스 아이템을 json 으로 변환한다.

Kind: instance method of [Aras](#)

Returns: Object – json

Param Description

node Aras Item

aras.getHtmlParameter(val) ⇒ String

URL 에서 지정된 파라미터의 get 프로퍼티를 가져온다.

Kind: instance method of [Aras](#)

Returns: String – parameter value

Param Description

val url 파라미터

aras.init()

아라스 객체를 얻기 위한 init 메소드. 최초 한번 실행한다.

부모 페이지로부터 워크플로우 아이템을 받아와서 적용하고, 부모페이지의 리사이즈 이벤트에 반응해 페이지를 레이아웃을 재구성한다.

Kind: instance method of [Aras](#)

aras.createBody(params) ⇒ string

key value 오브젝트로부터 xml 바디 스트링을 만든다

Kind: instance method of [Aras](#)

Returns: string – body string

Param Description

params Object

aras.applyMethod(methodName, body) ⇒ Object

주어진 메소드 이름과 body 스트링으로 아라스의 applyMethod 를 호출한다.

Kind: instance method of [Aras](#)

Returns: Object – Aras Item

Param

methodName

body

aras.getUserIdentity() ⇒ String

현재 접속중인 사용자의 IdentityId 를 반환한다.

Kind: instance method of [Aras](#)

Returns: String – identityId

aras.getUserId() ⇒ String

현재 접속중인 사용자의 아이디를 반환한다.

Kind: instance method of [Aras](#)

Returns: String - userId

aras.getWorkflowData(wf_id) ⇒ Object

워크플로우 데이터를 반환한다.

Kind: instance method of [Aras](#)

Returns: Object - Aras Item

Param Description

wf_id 워크플로우 아이디

aras.getItemById(type, id) ⇒ Object

타입과 아이디와 매칭된 데이터를 반환한다.

Kind: instance method of [Aras](#)

Returns: Object - Aras Item

Param

type

id

aras.getActivityStructure(activity_id, folder_yn, folder_id, inout) ⇒ Object

하나의 폴더 또는 Activity 를 기준으로 하위 폴더와 ED 조회

Kind: instance method of [Aras](#)

Returns: Object - Aras Item

Param Description

activity_id

folder_yn Y/N

folder_id

inout IN / OUT

aras.getEdParentList(id, ed_yn) ⇒ Array

선택한 폴더 또는 ED 를 input 으로 쓰는 워크플로우 - Activity 리스트 조회

Kind: instance method of [Aras](#)

Returns: Array - json array

Param Description

id

ed_yn Y/N

aras.getPickEd(ed_number, ed_name) ⇒ Array

PICK ED 에 대한 조회 리스트(Project 에서만 필요)

Kind: instance method of [Aras](#)

Returns: Array – json array

Param

ed_number

ed_name

aras.nodeToJson(xmlNode) ⇒ Object

Aras Item 노드를 Json 으로 변환한다.

Kind: instance method of [Aras](#)

Returns: Object – json

Param

xmlNode

aras.getSchCombo(kind, discLine, discSpec, bg, REL_WF_ID, callback)

아더 워크플로우의 셀렉트 박스 리스트의 내용을 구한다.

Kind: instance method of [Aras](#)

Param

kind

discLine

discSpec

bg

REL_WF_ID

callback

aras.getCurrentItemId(itemType, id) ⇒ Object

주어진 아이템의 아이디로 현재 아이템 상태를 조회한다.

Kind: instance method of [Aras](#)

Returns: Object – Aras Item

Param

itemType

id

aras.getItemType(type) ⇒ String

주어진 타입으로 아라스 아이템 타입을 구한다.

Kind: instance method of [Aras](#)

Returns: String – itemType

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

type "workflow","activity","folder","ed"

aras.getRelType(sourceType, targetType, inout) ⇒ String

주어진 소스의 타입, 타겟의 타입, 인아웃 으로 아라스 릴레이션 아이템 타입을 구한다.

Kind: instance method of [Aras](#)

Returns: String – itemType

Param	Description
sourceType	"workflow","activity","folder","ed"
targetType	"workflow","activity","folder","ed"
inout	"in","out"

aras.showPropertyWindow(type, id)

주어진 타입과 아이디로 아라스의 아이템 상세정보창을 띄운다

Kind: instance method of [Aras](#)

Param	Description
type	"workflow","activity","folder","ed"
id	

aras.sortActivities(activityIds)

주어진 액티비티 아이디 배열에 따라 아라스 액티비티 아이템을 소팅한다.

Kind: instance method of [Aras](#)

Param	Description
activityIds	Array of Aras Activity id

aras.checkMaxCreateNumber(depth) ⇒ boolean

주어진 아이템의 depth 로 추가 생성이 가능한지 여부를 반환한다.

Kind: instance method of [Aras](#)

Returns: boolean – enable

Param
depth

aras.createFolder(data, view)

주어진 OG-Tree 폴더 하위에 신규 아라스 폴더를 생성하는 팝업창을 띄운다.

Kind: instance method of [Aras](#)

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

aras.addFolderOutRelation(parentData, parentView, newItem, parentItem, parentType, parentId)

주어진 부모 아이템과 자식 아이템(폴더) 사이에 릴레이션을 생성한다.

Kind: instance method of [Aras](#)

Param	Description
parentData	OG-Tree parent data
parentView	OG-Tree parent view data
newItem	Aras created folder item
parentItem	Aras parent item
parentType	"workflow", "activity", "folder", "ed"
parentId	Aras parent id

aras.createEd(data, view, edType)

주어진 OG-Tree 폴더 하위에 신규 아라스 ED를 생성하는 팝업창을 띄운다.

Kind: instance method of [Aras](#)

Param	Description
data	OG-Tree data
view	OG-Tree view data
edType	'CAD','DHI_C3D_OUTPUT','Document','DHI_IntelliSheet','DHI_ED_KDM'

aras.addFolderEDOutRelation(edItem, parentItem, data, view)

주어진 부모 아이템(폴더)과 신규 생성한 자식 아이템(ED) 사이에 릴레이션을 생성한다.

Kind: instance method of [Aras](#)

Param	Description
edItem	Aras created ED item
parentItem	Aras parent Folder item
data	OG-Tree parent data
view	OG-Tree parent view data

aras.addPickEDOutRelation(edItems, parentItem, data, view)

주어진 부모 아이템(폴더)과 Pick 된 아이템(ED)들 사이에 DDCL 체크 후 릴레이션을 생성한다.

Kind: instance method of [Aras](#)

Param	Description
edItems	Array of Picked ED Items
parentItem	Aras parent Folder item
data	OG-Tree parent data
view	OG-Tree parent view data

aras.deleteOutItem(data, view)

지정된 아웃데이터 아이템(액티비티, 폴더, ED) 과 그 부모간의 릴레이션을 삭제한다.

Kind: instance method of [Aras](#)

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

aras.createActivity()

현재 워크플로우에 액티비티를 생성하는 Aras 팝업을 띄운다.

Kind: instance method of [Aras](#)

aras.addInRel(source, target, selectedTargetList)

지정된 마이 워크플로우의 소스와 아더 워크플로우의 타겟간에 릴레이션을 생성한다.

Kind: instance method of [Aras](#)

Param	Description
source	OG-Tree data (My workflow)
target	OG-Tree data (Other workflow)
selectedTargetList	Array OG-Tree data (Other workflow) target 의 하위 아이템 데이터

aras.deleteInRel(sourceId, sourceType, targetId, targetType)

주어진 마이워크플로우 소스아이디와 아더워크플로우 타겟 아이디간의 릴레이션을 삭제한다.

Kind: instance method of [Aras](#)

Param	Description
sourceId	Aras Item id (My workflow)
sourceType	"workflow","activity","folder","ed"
targetId	Aras Item id (Other workflow)
targetType	"workflow","activity","folder","ed"

aras.currentSortOrder() ⇒ Object

화면의 소트 메뉴로부터 현재 소트 지정값을 알아온다.

Kind: instance method of [Aras](#)

Returns: Object – key : 소트 키, order : asc/desc

aras.createWorkFlowData(resultNodeList, who, inout) ⇒ Array

Aras 의 메소드 리턴값을 오픈그래프 트리 데이터로 변환한다.

Kind: instance method of [Aras](#)

Returns: Array – json

Param	Description
resultNodeList	Aras 메소드 리턴값의 nodeList

who 마이,아더 워크플로우 여부 other/my
inout 인 데이터, 아웃데이터 여부 in/out

- [.createWorkFlowData\(resultNodeList, who, inout\)](#) ⇒ Array
 - [~getStateColorAndStroke\(type, state, isDelay\)](#) ⇒ Object
 - [~convertDate\(dateStr\)](#) ⇒ *
 - [~checkDelay\(node\)](#) ⇒ boolean

createWorkFlowData~getStateColorAndStroke(type, state, isDelay) ⇒ Object

스테이트 컬러를 반환한다. 딜레이 값이 있을 경우 stroke 를 더한다.

Kind: inner method of [createWorkFlowData](#)

Param

type
state
isDelay

createWorkFlowData~convertDate(dateStr) ⇒ *

월/일/년 형식의 데이터를 년월일 형식으로 교체한다.(ex) 20160901)

Kind: inner method of [createWorkFlowData](#)

Param

dateStr

createWorkFlowData~checkDelay(node) ⇒ boolean

딜레이 여부를 반환한다.

Kind: inner method of [createWorkFlowData](#)

Param

node

aras.createMyWorkFlowData(resultNodeList, inout) ⇒ Array

Aras 메소드 리턴값을 오픈그래프 마이워크플로우 데이터로 변환한다.

Kind: instance method of [Aras](#)

Returns: Array – json

Param

Description

resultNodeList	Aras 메소드 리턴값의 nodeList
inout	인아웃 여부

aras.createOtherWorkFlowData(resultNodeList) ⇒ Array

Aras 메소드 리턴값을 오픈그래프 아더워크플로우 데이터로 변환한다.

Kind: instance method of [Aras](#)

Returns: Array – json

Param	Description
resultNodeList	Aras 메소드 리턴값의 nodeList

aras.refreshOtherWorkflow(wfId)

주어진 아더 워크플로우 아이디로 화면의 아더 워크플로우 트리를 갱신한다.

Kind: instance method of [Aras](#)

Param	Description
wfId	아더 워크플로우 아이디

aras.refreshMyWorkFlow()

현재 화면의 마이 워크플로우 트리를 갱신한다.

Kind: instance method of [Aras](#)

aras.refreshOutFolder(data, view)

주어진 마이 워크플로우의 폴더의 하위 요소를 갱신한다.

Kind: instance method of [Aras](#)

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

aras.syncExpandDataWithTree(data)

주어진 오픈그래프 트리 데이터를 화면에 적용시키기 전에, 폴더의 열고 닫음 상태를 화면과 동기화시킨다.

Kind: instance method of [Aras](#)

Param	Description
data	OG-Tree array data

Aras~stateJson

스테이트 정의가 저장되어 있는 파일을 불러온다.

Kind: inner property of [Aras](#)