

DataController

Kind: global class

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

- [DataController](#)
 - [new DataController\(tree\)](#)
 - *instance*
 - [.getWorkflowStructure](#)
 - [new getWorkflowStructure\(wf_id, inout\)](#)
 - [.viewController](#) : Doosan
 - [.tree](#) : Tree
 - [.chartRenderer](#)
 - [.aras](#) : aras
 - [.thisItem](#) : Object
 - [.wfid](#) : String
 - [.stdYN](#) : String
 - [.projectId](#) : String
 - [.inn](#) : null
 - [.prjMaxDepth](#) : number
 - [.stdMaxDepth](#) : number
 - [.TYPE](#) : Object
 - [.iExmL2jsobj\(node\)](#) ⇒ Object
 - [.getHtmlParameter\(val\)](#) ⇒ String
 - [.initResize\(\)](#)
 - [.init\(\)](#)
 - [.createBody\(params\)](#) ⇒ string
 - [.applyMethod\(methodName, body\)](#) ⇒ Object
 - [.getUserIdentity\(\)](#) ⇒ String
 - [.getUserId\(\)](#) ⇒ String
 - [.getUserTeam\(\)](#) ⇒ string
 - [.getWorkflowData\(wf_id\)](#) ⇒ Object
 - [.getWorkflowHeader\(wf_id\)](#) ⇒ * | 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\(type, id, data, view\)](#)
 - [.sortActivities\(activityIds\)](#)
 - [.checkMaxCreateNumber\(view, requestType\)](#) ⇒ 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\(wfId\)](#)
 - [.refreshAll\(\)](#)
 - [.refreshMyWorkFlow\(\)](#)
 - [.refreshOutFolder\(data, view\)](#)
 - [.syncExpandDataWithTree\(data\)](#)
 - [.getEDBTypeList\(\)](#) ⇒ * | Object
 - [.getRelOtherPropsForEditor\(\)](#) ⇒ * | Array
 - [.getProjectMember\(\)](#)
 - [.updatePromote\(checkedList\)](#)
 - [.updateOwner\(checkedList, identityId\)](#)
 - [.updateName\(data, view, name\)](#)
 - [.convertMethodResultToJsonArray\(result\)](#) ⇒ Array
 - [.checkPM\(\)](#) ⇒ *
 - [.getProjectMapData\(\)](#)
 - [.getKeyActivityList\(\)](#) ⇒ * | Object
 - [.getEngFuncCodeList\(\)](#) ⇒ * | Object
 - [.getObjActivityList\(\)](#) ⇒ * | Object
 - [.saveMapData\(mapData|json\)](#) ⇒ * | Object
 - [.getChartData\(\)](#) ⇒ Object
- *inner*
- [~state|son](#)
 - [~chartState|son](#)

new DataController(tree)

Aras data Handler

Param	Description
tree	오픈그래프 트리 라이브러리

dataController.getWorkflowStructure

Kind: instance class of [DataController](#)

new getWorkflowStructure(wf_id, inout)

WF 하위의 액티비티, 폴더 및 ED 조회

Returns: Object – Aras Item

Param Description

wf_id

inout IN/OUT

dataController.viewController : Doosan

Doosan view controller

Kind: instance property of [DataController](#)

dataController.tree : Tree

Renderer 객체

Kind: instance property of [DataController](#)

dataController.chartRenderer

차트 렌더러

Kind: instance property of [DataController](#)

dataController.aras : aras

부모페이지의 aras 객체

Kind: instance property of [DataController](#)

dataController.thisItem : Object

현재의 워크플로우 아이템

Kind: instance property of [DataController](#)

dataController.wfId : String

현재의 워크플로우 아이디

Kind: instance property of [DataController](#)

dataController.stdYN : String

현재 워크플로우의 스탠다드 모드 여부

Kind: instance property of [DataController](#)

dataController.projectId : String

현재 워크플로우의 프로젝트 아이디

Kind: instance property of [DataController](#)

dataController.inn : null

newIOMInnovator 리턴 객체

Kind: instance property of [DataController](#)

dataController.prjMaxDepth : number

프로젝트, 폴더 5레벨 까지 enable, ed 는 6레벨까지 가능

Kind: instance property of [DataController](#)

dataController.stdMaxDepth : number

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

Kind: instance property of [DataController](#)

dataController.TYPE : Object

아이템 타입 Contants

Kind: instance property of [DataController](#)

dataController.iExmL2jsobj(node) ⇒ Object

IE 에서 아라스 아이템을 json 으로 변환한다.

Kind: instance method of [DataController](#)

Returns: Object – json

Param Description

node Aras Item

dataController.getHtmlParameter(val) ⇒ String

URL 에서 지정된 파라미터의 get 프로퍼티를 가져온다.

Kind: instance method of [DataController](#)

Returns: String – parameter value

Param Description

val url 파라미터

dataController.initResize()

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

Kind: instance method of [DataController](#)

dataController.init()

아라스 객체를 얻기 위한 init 메소드. 최초 한번 실행한다.

Kind: instance method of [DataController](#)

dataController.createBody(params) ⇒ string

key value 오브젝트로부터 xml 바디 스트링을 만든다

Kind: instance method of [DataController](#)

Returns: string – body string

Param Description

params Object

dataController.applyMethod(methodName, body) ⇒ Object

주어진 메소드 이름과 body 스트링으로 아라스의 applyMethod 를 호출한다.

Kind: instance method of [DataController](#)

Returns: Object – Aras Item

Param

methodName

body

dataController.getUserIdentity() ⇒ String

현재 접속중인 사용자의 IdentityId 를 반환한다.

Kind: instance method of [DataController](#)

Returns: String – identityId

dataController.getUserId() ⇒ String

현재 접속중인 사용자의 아이디를 반환한다.

Kind: instance method of [DataController](#)

Returns: String – userId

dataController.getUserTeam() ⇒ string

로그인한 유저의 팀을 알아온다.

Kind: instance method of [DataController](#)

dataController.getWorkflowData(wf_id) ⇒ Object

워크플로우 데이터를 반환한다.

Kind: instance method of [DataController](#)

Returns: Object – Aras Item

Param Description

wf_id 워크플로우 아이디

dataController.getWorkflowHeader(wf_id) ⇒ * | Object

워크플로우 헤더정보를 얻어온다.

Kind: instance method of [DataController](#)

Param

wf_id

dataController.getItemById(type, id) ⇒ Object

타입과 아이디와 매칭된 데이터를 반환한다.

Kind: instance method of [DataController](#)

Returns: Object – Aras Item

Param Description

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

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

하나의 폴더 또는 Activity 를 기준으로 하위 폴더와 ED 조회

Kind: instance method of [DataController](#)

Returns: Object – Aras Item

Param Description

activity_id

folder_yn Y/N

folder_id

inout IN / OUT

dataController.getEdParentList(id, ed_yn) ⇒ Array

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

Kind: instance method of [DataController](#)

Returns: Array – json array

Param Description

id

ed_yn Y/N

dataController.getPickEd(ed_number, ed_name) ⇒ Array

PICK ED 에 대한 조회 리스트(Project 에서만 필요)

Kind: instance method of [DataController](#)

Returns: Array – json array

Param

ed_number

ed_name

dataController.nodeToJson(xmlNode) ⇒ Object

Aras Item 노드를 Json 으로 변환한다.

Kind: instance method of [DataController](#)

Returns: Object – json

Param

xmlNode

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

아더 워크플로우의 셀렉트 박스 리스트의 내용을 구한다.

Kind: instance method of [DataController](#)

Param

kind
discLine
discSpec
bg
REL_WF_ID
callback

dataController.getCurrentItemId(itemType, id) ⇒ Object

주어진 아이템의 아이디로 현재 아이템의 아이디를 리턴한다.

Kind: instance method of [DataController](#)

Returns: Object – Aras Item

Param

itemType
id

dataController.getItemType(type) ⇒ String

주어진 타입으로 아라스 아이템 타입을 구한다.

Kind: instance method of [DataController](#)

Returns: String – itemType

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

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

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

Kind: instance method of [DataController](#)

Returns: String – itemType

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

dataController.showPropertyWindow(type, id, data, view)

주어진 타입과 아이디로 아라스의 아이템 상세정보창을 띄운다

Kind: instance method of [DataController](#)

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

type "workflow","activity","folder","ed"
id
data optional
view optional

dataController.sortActivities(activityIds)

주어진 액티비티 아이디 배열에 따라 아라스 액티비티 아이템을 소팅한다.

Kind: instance method of [DataController](#)

Param	Description
activityIds	Array of Aras Activity id

dataController.checkMaxCreateNumber(view, requestType) ⇒ boolean

주어진 아이템의 depth 로 추가 생성이 가능한지 여부를 반환한다.

Kind: instance method of [DataController](#)

Param
view
requestType

dataController.createFolder(data, view)

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

Kind: instance method of [DataController](#)

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

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

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

Kind: instance method of [DataController](#)

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

dataController.createEd(data, view, edType)

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

Kind: instance method of [DataController](#)

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

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

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

Kind: instance method of [DataController](#)

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

dataController.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 [DataController](#)

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

dataController.deleteOutItem(data, view)

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

Kind: instance method of [DataController](#)

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

dataController.createActivity()

현재 워크플로우에 액티비티를 생성하는 Aras 팝업을 띄운다.

Kind: instance method of [DataController](#)

dataController.addInRel(source, target, selectedTargetList)

지정된 마이 워크플로우의 소스와 아더 워크플로우의 타겟간에 릴레이션을 생성한다.

Kind: instance method of [DataController](#)

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

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

주어진 마이워크플로우 소스아이디와 아더워크플로우 타겟 아이디간의 릴레이션을 삭제한다.

Kind: instance method of [DataController](#)

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

dataController.currentSortOrder() ⇒ Object

화면의 소트 메뉴로부터 현재 소트 지정값을 알아온다.

Kind: instance method of [DataController](#)

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

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

Aras 의 메소드 리턴값을 오픈그래프 트리 데이터로 변환한다.

Kind: instance method of [DataController](#)

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

dataController.createMyWorkFlowData(resultNodeList, inout) ⇒ Array

Aras 메소드 리턴값을 오픈그래프 마이워크플로우 데이터로 변환한다.

Kind: instance method of [DataController](#)

Returns: Array - json

Param	Description
resultNodeList	Aras 메소드 리턴값의 nodeList
inout	인아웃 여부

dataController.createOtherWorkFlowData(resultNodeList) ⇒ Array

Aras 메소드 리턴값을 오픈그래프 아더워크플로우 데이터로 변환한다.

Kind: instance method of [DataController](#)

Returns: Array - json

Param	Description
resultNodeList	Aras 메소드 리턴값의 nodeList

dataController.refreshOtherWorkflow(wfld)

주어진 아더 워크플로우 아이디로 화면의 아더 워크플로우 트리를 갱신한다.

Kind: instance method of [DataController](#)

Param	Description
wfld	아더 워크플로우 아이디

dataController.refreshAll()

현재 화면의 아더, 마이 워크플로우를 모두 갱신한다.

Kind: instance method of [DataController](#)

dataController.refreshMyWorkFlow()

현재 화면의 마이 워크플로우 트리를 갱신한다.

Kind: instance method of [DataController](#)

dataController.refreshOutFolder(data, view)

주어진 마이 워크플로우의 폴더의 하위 요소를 갱신한다.

Kind: instance method of [DataController](#)

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

dataController.syncExpandDataWithTree(data)

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

Kind: instance method of [DataController](#)

Param	Description
data	OG-Tree array data

dataController.getEDBTypeList() ⇒ * | Object

EDB 타입 리스트를 불러온다.

Kind: instance method of [DataController](#)

dataController.getRelOtherPropsForEditor() ⇒ * | Array

현재 워크플로우와 연계된 아더워크플로우 목록을 가져온다.

Kind: instance method of [DataController](#)

dataController.getProjectMember()

현재 워크플로우의 프로젝트 멤버 목록을 가져온다.

Kind: instance method of [DataController](#)

dataController.updatePromote(checkedList)

체크된 리스트에 대해 promote 처리한다.

Kind: instance method of [DataController](#)

Param

checkedList

dataController.updateOwner(checkedList, identityId)

체크된 리스트에 대하여 담당자를 변경한다.

Kind: instance method of [DataController](#)

Param

checkedList

identityId

dataController.updateName(data, view, name)

오브젝트의 이름을 변경한다.

Kind: instance method of [DataController](#)

Param

data

view

name

dataController.convertMethodResultToJsonArray(result) ⇒ Array

메소드 리설트를 JSON 어레이 형식으로 변환한다.

Kind: instance method of [DataController](#)

Param

result

dataController.checkPM() ⇒ *

PM 자격을 조회한다.

Kind: instance method of [DataController](#)

dataController.getProjectMapData()

차크 맵데이터를 가져온다.

Kind: instance method of [DataController](#)

dataController.getKeyActivityList() ⇒ * | Object

차트 헤더 리스트를 가져온다.

Kind: instance method of [DataController](#)

dataController.getEngFuncCodeList() ⇒ * | Object

차트 row 리스트를 가져온다.

Kind: instance method of [DataController](#)

dataController.getObjActivityList() ⇒ * | Object

차트 액티비티 리스트를 가져온다.

Kind: instance method of [DataController](#)

dataController.saveMapData(mapDataJson) ⇒ * | Object

차트 맵 데이터를 저장한다.

Kind: instance method of [DataController](#)

Param

mapDataJson

dataController.getChartData() ⇒ Object

차트맵, 헤더, 로우, 액티비티를 가져와 조합하여 렌더러에서 사용하는 데이터 형식으로 돌려준다.

Kind: instance method of [DataController](#)

DataController~stateJson

스태이트 정의가 저장되어 있는 파일을 불러온다.

Kind: inner property of [DataController](#)

DataController~chartStateJson

차트 스테이트 정의가 저장되어 있는 파일을 불러온다.

Kind: inner property of [DataController](#)