**DEVELOPER GUIDE**

**AngularJS Controller**

AngularJS adalah framework yang kami gunakan dalam membuat controller pada sisi client (java script), tujuannya adalah untuk menyederhanakan dan memisahkan antara koding yang digunakan untuk membuat User Interface (UI – Widget) dengan koding yg digunakan untuk menghandle fungsi fungsi yang berhubungan dengan manipulasi data.

Dan agar dapat bekerja pada widget yang sudah ada (SDMS Widget), kami juga menambahkan beberapa attribute yang berfungsi sebagai penghubung antara Widget dengan controller, dan selain itu dengan penggunaan attribute-attribute tersebut dapat mengeliminasi penggunaan selector, sehingga penulisan kode program dapat lebih sederhana.

Berikut ini adalah **attribute-attribute** baru yang kami define:

1. **model**: adalah attribute yang digunakan sebagai data-binding antara object/komponen widget dengan AngularJS controller. Nilai masukan untuk attribute ini berupa string untuk nama field yang akan di binding, dan jika attribute ini tidak di define (deklarasikan), maka secara otomatis object tersebut akan di binding ke default model berdasarkan nama dari object tersebut.

{ … , **model**: “data.CompanyName”, … }

1. **show**: adalah attribute yang berfungsi untuk mengendalikan visibility dari object/komponen UI. Nilai masukan untuk attribute ini dapat berupa variable ataupun fungsi dari controller, yg mana jika variable atau fungsi tersebut bernilai TRUE, maka object/komponen UI tersebut akan tampil an jika bernilai FALSE, maka object tersebut akan tidak tampil (Hide).

{ … , **show**: “isLoadData”, … } *// Visibility berdasarkan nilai dari variable isLoadData*

1. **disable**: adalah attribute yang digunakan untuk mengatur aktif tidaknya fungsi input atau object Widget. Nilai masukan untuk attribute ini dapat berupa variable atau fungsi dari controller, yang mana jika variable atau fungsi tersebut bernilai TRUE, maka object secara otomatis akan disabled (tidak aktif/readonly), dan jika bernilai FALSE maka object tersebut akan aktif.

{ …, **disable**: "IsEditing()" , … } *// Disabled berdasarkan nilai dari fungsi IsEditing*

1. **options**: adalah attribute yang dipakai pada object SELECT yang berfungsi untuk menginisialisasikan nilai pilihan dari object SELECT tersebut. Nilai masukan untuk attribute ini berupa string yang mirip dengan statement linq.

{ …, **options**: "x.text for x in dataOption track by x.value", … }

1. **click**: adalah attribute yang digunakan untuk menghandle event click pada object/item widget tersebut. Attribute ini biasa dipakai pada object button atau hyperlink. Nilai masukan untuk attribute ini berupa nama fungsi yang tersedia pada controller.

{ …, **click**: “Process()”,… }

1. **change**: adalah attribute yang digunakan untuk menghandle event change pada object/item tersebut. Nilai masukan untuk attribute ini berupa nama fungsi yang tersedia pada controller.

{ …, **change**: “calculate()”, … }

1. **event**: adalah attribute yang digunakan untuk menghandle custom event dari object/item widget tersebut, misalnya event blur. Nilai masukan untuk attribute tersebut berupa nilai JSON yang mana fieldnya merupakan event yang ingin dicapture, sedangkan valuenya merupakan nama fungsi yang tersedia di controller.

{ …, **event**: “{ blur: ‘OnTxtPriceBlur($event)’ }”, …}

1. **keypress**: adalah attribute yang digunakan untuk menghandle event keypress dari object/item widget tersebut. Nilai masukan untuk attribute tersebut berupa nilai JSON yang mana fieldnya merupakan keycode yang ingin dicapture, sedangkan valuenya merupakan nama fungsi yang tersedia di controller.

{ …, **keypress**: “{ 13: ‘keypressCallback($event)’ }”, …}

**Global Variable**

Global variable adalah standard variable global yang digunakan dalam pengembangan antar muka aplikasi klien (UI widget) yang sudah di binding dengan standard controller (base controller).

Adapun global variable tersebut adalah:

1. **Wx**: adalah variable yang berfungsi untuk menampung instance dari object SimDMS.Widget
2. **WxButtons**: adalah variable atau bisa juga disebut sebagai konstanta yang merupakan standard dari button-button yang digunakan pada toolbar disetiap form aplikasi. Dengan demikian variable ini dapat digunakan untuk mendefine object toolbar items dari form standard dan juga sudah terintegrasi dengan base controller, baik secara fungsi maupun visibility nya.

**Base Controller**

Base controller adalah induk/template untuk pembentukan controller-controller yang akan digunakan dalam pengembangan aplikasi antar muka (UI Widget) , yang mana sebagian fungsi-fungsinya sudah diintegrasikan dengan fungsi-fungsi umum yang ada pada widget.

function **BaseController**($scope)

{

// Untuk mempermudah penulisan kode program, define variable me sebagai $scope (Alias)

var **me** = $**scope**;

// Define object untuk menyimpan/menampung data (baik dari interaksi dgn user (editing) mapupun load dari database)

**me**.data = {};

// define local variable hasChange, fungsinya sebagai flag indikator perubahan data

**me**.hasChanged = false;

// define local variable isLoadData, fungsinya sebagai flag indikator load data dari browse function

**me**.isLoadData = false;

// define local variable isEditable, fungsinya sebagai flag indikator untuk memulai proses edit data (disabled primary/composite key)

**me**.isEditable = false;

// define local variable isSave, fungsinya sebagai flag indikator untuk menunjukan adanya perubahan data yg harus disimpan

**me**.isSave = false;

// Aktifkan fungsi $watch untuk memonitor perubahan data yang terjadi pada object 'data'

**me**.$**watch**('data', function(newValue, oldValue)

{

// check apakah perubahan data tersebut memiliki nilai atau object kosong (empty object)

if (!(\_.isEmpty(newValue)))

{

if(!me.hasChanged && !me.isLoadData)

{

me.hasChanged = true;

me.isLoadData = false;

} else if (!me.isSave)

{

me.isSave = true;

}

} else {

me.hasChanged = false;

me.isSave = false;

}

}, true);

**me**.**init** = function()

{

**me.data** = {};

if (**Wx** !== undefined) {

**Wx.reset**();

}

**me.hasChanged** = false;

**me.isLoadData** = false;

**me.isEditable** = false;

**me.isSave** = false;

$("#btnCancel").html("<i class='icon icon-undo'></i>Cancel");

if (this.hasOwnProperty("clear")) {

**me.clear.call**();

}

}

// fungsi ini digunakan untuk menset flag indicator,

// agar state dari input/button berubah (show/hide/enable/disable)

**me**.**allowEdit** = function()

{

**me.isLoadData** = false;

**me.isEditable** = true;

**me.hasChanged** = true;

$("#btnCancel").html("<i class='icon icon-undo'></i>Cancel");

}

// fungsi bernilai TRUE, jika data dalam proses editing

**me.IsEditing** = function()

{

return **me.isEditable** || **me.isLoadData**;

}

// fungsi ini digunakan untuk mengembalikan nilai object dari index element pada input selection

**me**.**GetObjectById** = function(o,s)

{

if (s === undefined || s == null ) return null;

var pos = $.map(o, function(obj, index) {

if(obj.value == $.trim(s.toString())) {

return obj;

}

});

if (pos != null )

return pos[0];

else

return null;

}

// change button & control state by flag indicator

**me**.**startEditing** = function()

{

**me.isEditable** = true;

**me.isSave** = false;

*// change cancel button caption to "Close"*

$("#btnCancel").html("<i class='icon icon-hand-right'></i>Close");

}

// save data, posting data from user cache to the server

**me**.**save** = function()

{

if (Wx !== undefined)

{

**Wx.submit**();

}

}

**me**.**delete** = function()

{

logIt("action save call on base controller");

}

**me**.**OnResize** = function(width, height)

{

}

**me**.**lookupAfterSelect** = function(value)

{

**me**.**isLoadData** = true;

**me**.**data** = value;

**me**.**hasChanged** = false;

**me**.**startEditing**();

}

**me**.**Apply** = function()

{

**$scope.$apply**();

}

// start up

**me**.**init**();

}

**Cara Menggunakan AngularJS controller**

Berikut ini adalah langkah-langkah yang diperlukan dalam mengembakan aplikasi dengan SimDMS.Widget dan AngularJS:

1. Define UI (SimDMS.Widget)
2. Define AngularJS controller

Contoh UI untuk Moving Code (Master Sparepart ):

**$(document).ready(function () {**

var options = {

title: "Moving Code",

xtype: "panels",

toolbars: **WxButtons**,

panels: [ {

name: "pnlMC", title: "Moving Code",

items: [

{ name: "MovingCode", text: "Moving Code", cls: "span3", **disable: "IsEditing()",** **validasi:"required"** },

{ name: "MovingCodeName", text: "MovingCode Name", cls: "span3", validasi:"required" },

{

text: "Parameter", type: "controls",

items: [

{ name: "Param1", cls: "col-x-2", placeHolder: "0"},

{ name: "Sign1", type: "select", cls: "col-x-3", model:"sign1", **change:"updateOption(1)",** **options: "o.text for o in dataOption track by o.value"** },

{ name: "Variable", cls: "col-x-2", placeHolder: "x",

readonly:true, style:"text-align:center; font-size:24px; line-height:34px; padding-top:0px; padding-bottom:5px;" },{ name: "Sign2", type: "select", cls: "col-x-3", model:"sign2" ,

options: "o.text for o in dataOption track by o.value" , change:"updateOption(2)"},

{ name: "Param2", cls: "col-x-2", placeHolder: "0"},

]

}

]

},

]

};

Wx = new SimDms.Widget(options);

Wx.default = {};

Wx.render(init);

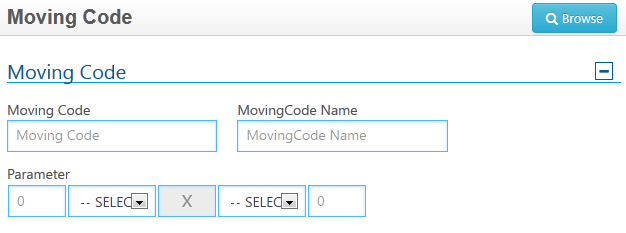
function init() {

**SimDms.Angular**("SpMovingController");

}

**});**

**spMovingController** adalah nama Angular controller (class/fungsi pada widget Moving Code)



Dari hasil render diatas, kita melihat bahwa hanya terdapat 1 button saja yang tampil pada toolbar, padahal pada list **WxButtons** kita memiliki 5 item button, yaitu: Button Browse, Edit, Delete, Save dan Cancel.

var **WxButtons** = [

{ name: "btnBrowse", text: "Browse", cls:"btn btn-info", icon: "icon-search", show: "!hasChanged" , click: "browse()" },

{ name: "btnEdit", text: "Edit", cls:"btn btn-primary", icon: "icon-edit", show: "isLoadData", click: "allowEdit()" },

{ name: "btnDelete", text: "Delete", cls:"btn btn-danger", icon: "icon-remove", show: "isLoadData", click: "delete()" },

{ name: "btnSave", text: "Save", cls:"btn btn-success", icon: "icon-save", show: "hasChanged", click: "save()", disable: "!isSave" },

{ name: "btnCancel", text: "Cancel", cls:"btn btn-warning", icon: "icon-remove", show: "hasChanged || isLoadData", click: "init()" },

Berikut ini penjelasan mengenai script diatas:

1. Button Search, akan tampil jika nilai ***hasChange*** = FALSE, itulah sebabnya hanya button search yang tampil pada saat pertama kali Widget di load.
2. Button Edit dan Button Delete akan tampil jika nilai isLoadData = **TRUE**, dan variable ***isLoadData*** hanya akan di set = **TRUE** pada saat user memilih data dari popup dialog yang ditampilkan oleh fungsi browse.
3. Button Save akan tampil jika ada perubahan data atau nilai ***hasChange*** = **TRUE**.
4. Button Cancel akan tampil jika nilai ***hasChange*** = **TRUE** atau nilai ***IsLoadData*** = **TRUE**.

**Angular Controller untuk Widget Moving Code:**

**function** SpMovingController($scope, $http, $injector) {

*// Untuk mempermudah penulisan kode program, define variable me sebagai $scope (Alias)*

var **me** = **$scope**;

*// Inheritance // Metode pewarisan pada object AngularJS Controller*

$injector.*invoke*(BaseController, this, { $scope: me});

**me**.**clear** = function()

*{*

*// fungsi ini digunakan untuk inisialisasi ataupun reset form widget*

}

// show popup dialog to select existing record

**me.browse** = function()

{

var lookup = Wx.**blookup**({

name: "movingcodeBrowse",

title: "Moving Code Browse",

manager: **spManager**,

query: "**movingcode**",

defaultSort: "MovingCode asc",

columns: [

{ field: 'MovingCode', title: 'Moving Code' },

{ field: 'MovingCodeName', title: 'Moving Code Name'},

{ field: 'Param1', title: 'Param1', type: "number" },

{ field: 'Sign1', title: 'Sign1' },

{ field: 'Param2', title: 'Param2', type: "number" },

{ field: 'Sign2', title: 'Sign2' },

],

});

*// fungsi untuk menghandle double click event pada k-grid / select button*

lookup.dblClick(function (data) {

if (data != null) {

**me**.sign1 = **me**.**GetObjectById**(me.dataOption, data.Sign1);

**me**.sign2 = **me**.**GetObjectById**(me.dataOption, data.Sign2);

**me**.**lookupAfterSelect**(data);

**me**.**Apply**();

**me**.isSave = false;

}

});

}

*// save data, posting data from user cache to the server*

**me.saveData** = function(e, param)

{

*// Using $http provider from angular to posting data to the server*

**$http**.**post**('sp.api/movingcode/save', **me.data**).**success**(function(data, status, headers, config) {

if (data.success){

**Wx**.Success("Data saved...");

**me**.**startEditing**();

} else {

*// show an error message*

**MsgBox**(data.message, **MSG\_ERROR**);

}

}).**error**(function(data, status, headers, config) {

**MsgBox**("Connection to the server failed...", **MSG\_ERROR**);

});

}

**me.delete** = function()

{

*// show confirmation dialog to the user*

*// check respond, if true, notify to the server to remove a current record*

**MsgConfirm**("Are you sure to delete current record?", function(result)

{

if (result)

{

*// call web api by $http provider*

**$http**.**post**('/sp.api/movingcode/delete', **me.data**)

.**success**(function(**v**, status, headers, config){

if(**v.success**)

{

**Wx.Info**("Record has been deleted...");

**me.init**(); *// reset to default state*

} else { *// show an error message*

**MsgBox**(v.message, **MSG\_ERROR**);

}

}).**error**(function(e, status, headers, config) {

**MsgBox**("Connection to the server failed...", **MSG\_ERROR**);

});

}

})

}

*// define object untuk inisialisasi data selection*

**me.dataOption** = [{ value: '<', text: '<' },

{ value: '<=', text: '<=' },

{ value: '=' , text: '=' },

{ value: '>' , text: '>' },

{ value: '>=', text: '>=' }];

*// fungsi untuk menset nilai data.sign1 berdasarkan event change dari selection control*

**me.updateOption** = function (i)

{

switch (i)

{

case 1: me.data.Sign1 = me.sign1 != null ? me.sign1.value : null; break;

case 2: me.data.Sign2 = me.sign2 != null ? me.sign2.value : null; break;

}

}

*// start up point*

if (**Wx** !== undefined)

{

**Wx.OnValidation** ( **me.saveData** );

}

}