Sage 300 Web Screens SDK

Grid Json Configuration file definition

August 2022

The MIT License (MIT)

Copyright © 2021-2022 The Sage Group plc or its licensors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contents

[1. Introduction 4](#_Toc65220952)

[**2. JOSN configuration file class definition 4**](#_Toc65220953)

[**3. Sample Json file 4**](#_Toc65220954)

1. Introduction

This document is intended to provide information on Grid Json configuration file.

The Json file is placed in the same place as screen razor view partial file. It’s used to defined the sg.viewList grid and columns.

2. JOSN configuration file class definition

namespace Sage.CA.SBS.ERP.Sage300.Common.Models

{

/// <summary>

/// Column alignment definition

/// </summary>

public enum ColumnAlignment

{

/// <summary>

/// align left

/// </summary>

Left,

/// <summary>

/// align center

/// </summary>

Center,

/// <summary>

/// align right

/// </summary>

Right

}

/// <summary>

/// Column definition

/// </summary>

public class ColumnDefinition

{

/// <summary>

/// Column View ID. Column field binding from other view, not from grid associated view

/// </summary>

public string ViewID { get; set; }

/// <summary>

/// Column header in the grid

/// </summary>

public string ColumnName { get; set; }

/// <summary>

/// Corresponding ACCPAC view fieldname. If blank, it is a calculated field

/// </summary>

public string FieldName { get; set; }

/// <summary>

/// Corresponding ACCPAC view field length

/// </summary>

public short FieldSize { get; set; }

/// <summary>

/// Column width

/// </summary>

public int Width {get; set;}

/// <summary>

/// Alignment

/// </summary>

public ColumnAlignment Alignment { get; set; }

/// <summary>

/// Determine whether the column is read only or not

/// </summary>

public bool IsEditable { get; set; } = true;

/// <summary>

/// Column foreign keys to join other view, key is comma delimited

/// </summary>

public string ForeignKeys { get; set; }

/// <summary>

/// Column Keys to join other view, key is comma delimited in keys string

/// </summary>

public string PrimaryKeys { get; set; }

/// <summary>

/// Reference Field name, this field value editor is depend on referene field

/// </summary>

public string ReferenceField { get; set; }

/// <summary>

/// Determine whether the column is hidden or not

/// </summary>

public bool IsHidden { get; set; }

/// <summary>

/// Determine whether the column is line number column, the grid display value is auto generated

/// </summary>

public bool IsLineNumber { get; set; }

/// <summary>

/// Determine whether the column is optional field column

/// </summary>

public bool IsOptionalField { get; set; }

/// <summary>

/// Determine whether the column is popup column

/// </summary>

public bool IsPopup { get; set; }

/// <summary>

/// Determine whether the field is in business view or not

/// </summary>

public bool IsVirtualField { get; set; }

/// <summary>

/// Determine whether the field is a primary key field

/// </summary>

public bool IsPrimaryKeyField { get; set; }

/// <summary>

/// Kendo data type. This is a calculated field and programmer doesn't need to set it

/// </summary>

public string DataType { get; set; }

/// <summary>

/// Column field numeric value decimal place

/// </summary>

public int Precision { get; set; }

/// <summary>

/// Grid Column finder definition

/// </summary>

public FinderDefinition Finder { get; set; }

/// <summary>

/// Presentation List

/// </summary>

public IEnumerable<CustomSelectList> PresentationList { get; set; }

/// <summary>

/// Presentation Mask

/// </summary>

public string PresentationMask { get; set; }

/// <summary>

/// Column drill down url //todo: set controller, action and area

/// </summary>

public string DrillDownUrl { get; set; }

/// <summary>

/// Drill down Url parameters

/// </summary>

public List<Parameter> Parameters { get; set; }

/// <summary>

/// custom define column level call back functions

/// </summary>

public IEnumerable<IDictionary<string, string>> CustomFunctions { get; set; }

}

/// <summary>

/// Grid definition

/// </summary>

public class GridDefinition

{

/// <summary>

/// Custom grid data service controller implements CRUD operations

/// </summary>

public string GridDataServiceController { get; set; }

/// <summary>

/// View ID associated with the grid

/// </summary>

public string ViewID { get; set; }

/// <summary>

/// ViewOrder, default 0

/// </summary>

public int ViewOrder { get; set; } = 0;

/// <summary>

/// Grid type: 0-Standard Grid, 1-OptionalField Grid, 2-Other type Grid

/// </summary>

public int GridType { get; set; } = 0;

/// <summary>

/// Page using record key instead of index and total. Increases performance for tables with many records

/// </summary>

public bool PageByKey { get; set; }

/// <summary>

/// Create columns from ColumnDefinitions instead of fields from view. Increases performance for views with many fields

/// </summary>

public bool ColumnsFromConfig { get; set; }

/// <summary>

/// Read only grid

/// </summary>

public bool ReadOnly { get; set; } = false;

/// <summary>

/// Grid page size

/// </summary>

public int PageSize { get; set; } = 10;

/// <summary>

/// Grid height

/// </summary>

public int Height { get; set; }

/// <summary>

/// GUID for the accpac view associated with the grid

/// </summary>

public string AccpacEntityGUID { get; set; }

/// <summary>

/// Grid Json definition file path

/// </summary>

public string GridJsonFilePath { get; set; }

/// <summary>

/// Determine whether it is a sequence or ordered revision list, reserved for future use

/// </summary>

public bool IsSequenceRevisionList { get; set; }

/// <summary>

/// Define all the columns in the grid

/// </summary>

public List<ColumnDefinition> ColumnDefinitions { get; set; }

/// <summary>

/// grid column preference settings

/// </summary>

public string GridColumnSettings { get; set; }

/// <summary>

/// grid preferences Guid

/// </summary>

public string UserPreferencesUniqueId { get; set; }

/// <summary>

/// custom define grid level call back functions

/// </summary>

public IEnumerable<IDictionary<string, string>> CustomFunctions { get; set; }

/// <summary>

/// Get and Set css class for the grid

/// </summary>

public string CssClass { get; set; }

}

/// <summary>

/// Finder definition

/// </summary>

public class FinderDefinition

{

/// <summary>

/// View ID

/// </summary>

public string ViewID { get; set; }

/// <summary>

/// ViewOrder, default 0

/// </summary>

public int ViewOrder { get; set; } = 0;

/// <summary>

/// Display filed names in UI grid

/// </summary>

public string[] DisplayFieldNames { get; set; }

/// <summary>

/// Finder select return fileds name in UI grid

/// </summary>

public string[] ReturnFieldNames { get; set; }

/// <summary>

/// Grid fields name for finder keys, used set filter and initKeyValues

/// </summary>

public string[] InitKeyFieldNames { get; set; }

/// <summary>

/// Filter for the finder. A complete browse filter is created in conjunction of the column filter

/// </summary>

public string Filter { get; set; }

/// <summary>

/// Custom defined finder source

/// </summary>

public string CustomFinderProperties { get; set; }

}

}

3. Sample Json file

{

"GridType": 0,

"ReadOnly": false,

"PageSize": 10,

"ViewID":"IC0580",

"CustomFunctions": [

{ "gridChanged": "receiptUI.customGridChanged" },

{ "gridAfterSetActiveRecord": "receiptUI.customGridAfterSetActiveRecord" },

{ "gridBeforeDelete": "receiptUI.customGridBeforeDelete" },

{ "gridAfterDelete": "receiptUI.customGridAfterDelete" },

{ "gridBeforeCreate": "receiptUI.customGridBeforeCreate" },

{ "gridAfterCreate": "receiptUI.customGridAfterCreate" },

{ "gridAfterInsert": "receiptUI.customGridAfterInsert" }

],

"ColumnDefinitions": [

{

"FieldName": "LINENO",

"IsEditable": false,

"IsLineNumber": true

},

{

"FieldName": "SEQUENCENO"

},

{

"FieldName": "ITEMNO",

"Precision": 0,

"IsEditable": true,

"Finder": {

"ViewID": "IC0310",

"ViewOrder": 0,

"DisplayFieldNames": [

"ITEMNO",

"DESC",

"INACTIVE",

"ITEMBRKID",

"CATEGORY",

"CNTLACCT",

"STOCKUNIT",

"DEFPRICLST"

],

"ReturnFieldNames": [

"ITEMNO"

],

"InitKeyFieldNames": [

"UNFMTITMNO"

]

}

},

{

"FieldName": "ITEMDESC",

"IsEditable": false,

"DrillDownUrl": "IC/Item/Index",

"Parameters": [

{

"Name": "id",

"Field": "ITEMNO"

}

]

},

{

"ViewID": "IC0310",

"FieldName": "CNTLACCT",

"ForeignKeys": "ITEMNO",

"PrimaryKeys": "ITEMNO",

"IsEditable": false

},

{

"FieldName": "CATEGORY",

"CustomFunctions": [

{ "columnBeforeFinder": "receiptUI.updateFinderFilter" },

{ "columnBeforeEdit": "receiptUI.customColumnBeforeEdit" }

]

},

{

"FieldName": "RECPTYPE"

},

{

"FieldName": "LOCATION",

"IsEditable": true,

"Precision": 0,

"CustomFunctions": [

{ "columnBeforeFinder": "receiptUI.updateFinderFilter" }

],

"Finder": {

"ViewID": "IC0372",

"ViewOrder": 0,

"Filter": "ITEMNO=UNFMTITMNO",

"DisplayFieldNames": [

"ITEMNO",

"LOCATION",

"DESC",

"ACTIVE",

"USED",

"QTYONHAND",

"QTYONORDER",

"QTYSALORDR"

],

"ReturnFieldNames": [

"LOCATION"

],

"InitKeyFieldNames": [

"UNFMTITMNO",

"LOCATION"

]

}

},

{

"FieldName": "RECPQTY",

"IsEditable": true,

"CustomFunctions": [

{ "columnBeforeDisplay": "receiptUI.showGridTimeColumn" }

]

},

{

"FieldName": "RECPUNIT",

"IsEditable": true,

"Finder": {

"ViewID": "IC0750",

"ViewOrder": 0,

"Filter": "ITEMNO=UNFMTITMNO",

"DisplayFieldNames": [

"ITEMNO",

"UNIT",

"FACTOR"

],

"ReturnFieldNames": [

"UNIT"

],

"InitKeyFieldNames": [

"UNFMTITMNO",

"RECPUNIT"

]

}

},

{

"FieldName": "UNITCOST",

"IsEditable": true

},

{

"FieldName": "RECPCOST",

"IsEditable": true

},

{

"FieldName": "LABELS",

"IsEditable": true

},

{

"FieldName": "COMMENTS",

"IsEditable": false,

"CustomFunctions": [

{ "columnBeforeDisplay": "receiptUI.showGridCommentColumn" }

]

},

{

"FieldName": "RETURNQTY",

"IsEditable": true,

"IsHidden": true

},

{

"FieldName": "RETURNCOST",

"IsEditable": true,

"IsHidden": true

},

{

"FieldName": "ADJUNITCST",

"IsEditable": true,

"IsHidden": true

},

{

"FieldName": "ADJCOST",

"IsEditable": true,

"IsHidden": true

},

{

"FieldName": "MANITEMNO",

"IsEditable": true

},

{

"IsOptionalField": true,

"FieldName": "OptionalField",

"ColumnName": "Optional Field",

"IsEditable": true,

"IsVirtualField": true,

"CustomFunctions": [

{ "OptionalField": "receiptUI.showDetailOptionalField" }

]

}

]

}