Sage 300 Web Screens SDK

Grid Server Interface

May 2019

The MIT License (MIT)

Copyright © 2019 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](#_Toc66692919)

[2. Customization 5](#_Toc66692920)

[3. Methods 6](#_Toc66692921)

[3.1 Read 6](#_Toc66692922)

[3.2 Create 7](#_Toc66692923)

[3.3 Insert 7](#_Toc66692924)

[3.4 Update 8](#_Toc66692925)

[3.5 Refresh 8](#_Toc66692926)

[3.6 MoveTo 8](#_Toc66692927)

[3.7 Delete 9](#_Toc66692928)

[3.8 RefreshCurrentRecord 9](#_Toc66692929)

1. Introduction

This document is intended to provide information on the server-side grid interface.

There will be three types of grids for the web screens:

* View List
  + sg.viewList
  + Counterpart to the desktop’s **AccpacViewList**
  + Introduced in 2019.2

The only grid to support server-side customization

* Optional Field Control
  + sg.optionalFieldControl
  + Counterpart to the desktop’s **OFCtrl**
  + Introduced in 2019.2

Only supports EDIT mode currently (not SETUP mode)

* Grid Control
  + sg.gridControl
  + Counterpart to the desktop’s **AccpacGrid**
  + Supports a virtual table (i.e. Tax Authorities)

Not yet implemented

1. Customization

By default, server requests for the grid are routed to ***GridDataServiceController*** which is responsible for all CRUD operations against the underlying ACCPAC business view.

When a 3rd party wants to participate in the CRUD operations, it will need to:

1. Create a controller inherited from ***GridDataServiceController*** (i.e.   
   ***MyGridDataServiceController***.)
2. Override the default CRUD operation as needed for the customization
3. Register ***MyGridDataServiceController*** in the 3rd party bootstrapper
4. Define the **GridDataServiceController** property in the grid’s json configuration file (i.e. ***GridDataServiceController***: ***MyGridDataServiceController***)
5. Methods

|  |  |
| --- | --- |
| Name | Description |
| **Read** | Read a page of records starting at a specified position |
| **Create** | Create a new record |
| **Insert** | Create a new record |
| **Update** | Update a record |
| **Refresh** | Put a value into a field and refresh all the fields |
| **MoveTo** | Move the current record to a specified position |
| **Delete** | Delete the current record |
| **RefreshCurrentRecord** | Get grid current row data from view |

* 1. Read

|  |  |  |
| --- | --- | --- |
| ***ActionResult*** ***Read***(***string*** ***viewID***, ***int*** ***take***, ***int*** ***skip***, ***string*** ***filter, bool pageByKey = false, GridPageType pageType = GridPageType.FirstPage, IDictionary<string, string> record = null, bool columnsFromConfig = false, List<ColumnDefinition> columns = null***) | | |
| **Description** | Read a page of records starting at a specified position | |
| **Parameters** | ***viewID*** | The guid to the ACCPAC business entity |
|  | ***take*** | The number of records to read |
|  | ***skip*** | The starting point |
|  | ***filter*** | The filter for the ACCPAC business entity |
|  | ***pageByKey*** | Optional parameter, whether using PageByKey setting |
|  | ***pageType*** | Optional parameter, determines fetch direction when using PageByKey setting |
|  | ***record*** | Optional parameter, determines starting record when using PageByKey setting |
|  | ***columnsFromConfig*** | Optional parameter, whether using ColumnsFromConfig setting |
|  | ***columns*** | Optional parameter, list of columns defined in grid config used by ColumnsFromConfig setting |
| **Returns** | A JSON payload with the following properties:  **Total** Total number of records in this payload  **Data** An array of dictionaries for the records fetched from the business entity. Each dictionary contains all the fields in the ACCPAC business entity and their corresponding values. In addition, the dictionary must contain a constructed key named **KendoGridAccpacViewPrimaryKey** which contains a hashed string value from the primary key fields of the records  **IsFirstPage** Is current page the first page. Used by PageByKey setting to enable/disable paging buttons after page loads  **IsLastPage** Is current page the last page. Used by PageByKey setting to enable/disable paging buttons after page loads | |
| **Example** |  | |

* 1. Create

|  |  |  |
| --- | --- | --- |
| ***ActionResult*** ***Create***(***string*** ***viewID***, ***IDictionary****<****string****,* ***string****>* ***record***) | | |
| **Description** | Create a new record | |
| **Parameters** | ***viewID*** | The guid to the ACCPAC business entity |
|  | ***record*** | For sequence revision list, the record must be initialized with the row after which the new record will be inserted. For ordered revision list, record can be NULL |
| **Returns** | A JSON payload with the following properties:  **Data** A dictionary containing the fields in the ACCPAC business entity and the corresponding values. In addition, the dictionary must contain a constructed key named **KendoGridAccpacViewPrimaryKey** which contains a hashed string value from the primary key fields of the new record.  **UserMessage** Contains the message(s) returned from the business entity | |
| **Example** |  | |

* 1. Insert

|  |  |  |
| --- | --- | --- |
| ***ActionResult*** ***Insert***(***string*** ***viewID***, ***IDictionary****<****string****,* ***string****>* ***record***) | | |
| **Description** | Create a new record | |
| **Parameters** | ***viewID*** | The guid to the ACCPAC business entity |
|  | ***record*** | For sequence revision list, the record must be initialized with the row after which the new record will be inserted. For ordered revision list, record can be NULL |
| **Returns** | A JSON payload with the following properties:  **Data** A dictionary containing the fields in the ACCPAC business entity and the corresponding values. In addition, the dictionary must contain a constructed key named **KendoGridAccpacViewPrimaryKey** which contains a hashed string value from the primary key fields of the new record.  **UserMessage** Contains the message(s) returned from the business entity | |
| **Example** |  | |

* 1. Update

|  |  |  |
| --- | --- | --- |
| ***ActionResult*** ***Update***(***string*** ***viewID***, ***IDictionary****<****string****,* ***string****>* ***record***) | | |
| **Description** | Update a record | |
| **Parameters** | ***viewID*** | The guid to the ACCPAC business entity |
|  | ***record*** | The active record |
| **Returns** | None | |
| **Example** |  | |

* 1. Refresh

|  |  |  |
| --- | --- | --- |
| ***ActionResult*** ***Refresh***(***string*** ***viewID***, ***IDictionary****<****string****,* ***string****>* ***record, string fieldname, bool isNewRecord***) | | |
| **Description** | Put a value into a field and refresh all the fields | |
| **Parameters** | ***viewID*** | The guid to the ACCPAC business entity |
|  | ***record*** | The active record |
|  | ***fieldName*** | The field that has changed |
|  | ***isNewRecord*** | Indicate whether this is a new record |
| **Returns** | A JSON payload with the following properties:  **Data** A dictionary containing the fields in the ACCPAC business entity and the corresponding values. In addition, the dictionary must contain a constructed key named **KendoGridAccpacViewPrimaryKey** which contains a hashed string value from the primary key fields of the record.  **UserMessage** Contains the message(s) returned from the business entity | |
| **Example** |  | |

* 1. MoveTo

|  |  |  |
| --- | --- | --- |
| ***ActionResult*** ***MoveTo***(***string*** ***viewID***, ***IDictionary****<****string****,* ***string****>* ***record***) | | |
| **Description** | Move the record to a specified position | |
| **Parameters** | ***viewID*** | The guid to the ACCPAC business entity |
|  | ***record*** | The active record |
| **Returns** | A JSON payload with the following properties:  **Data** A dictionary containing the fields in the ACCPAC business entity and the corresponding values. In addition, the dictionary must contain a constructed key named **KendoGridAccpacViewPrimaryKey** which contains a hashed string value from the primary key fields of the record.  **UserMessage** Contains the message(s) returned from the business entity | |
| **Example** |  | |

* 1. Delete

|  |  |  |
| --- | --- | --- |
| ***ActionResult*** ***Delete***(***string*** ***viewID***, ***IDictionary****<****string****,* ***string****>* ***record***) | | |
| **Description** | Delete a record | |
| **Parameters** | ***viewID*** | The guid to the ACCPAC business entity |
|  | ***record*** | The active record |
| **Returns** | None | |
| **Example** |  | |

* 1. RefreshCurrentRecord

|  |  |  |
| --- | --- | --- |
| ***ActionResult*** ***RefreshCurrentRecord***(***string*** ***viewID)*** | | |
| **Description** | Retrieve the grid current record in the view and return to the client | |
| **Parameters** | ***viewID*** | The guid to the ACCPAC business entity |
| **Returns** | The record for grid current selected row | |
| **Example** |  | |