SedonaWeb/SedonaAPI 2.0 Release Notes

October 2024

Version 1.48.0

Contents

SedonaAPI 2.0 Enhancements/Features

How Sales Automation Synced Part Costs are Calculated

NOTE: This change is required for AlarmBiller version 4.50.0.

The SedonaAPI 2.0 part sync endpoints used by Sales Automation have been updated to return a Part Cost value based on each synced part's costing method instead of the Purchase Cost. This is how part costs are calculated for Sales Automation synchronized parts:

For parts in SedonaOffice identified as using Standard Cost:

- SA Sync Part Cost = Inventory Standard Cost when the part has a Warehouse ID
  - If for some reason, an inventory record is not found for a particular part, the SA Sync Part Cost is set to the Part Purchase Cost

For parts in SedonaOffice identified with a costing method other than Standard Cost:

- SA Sync Part Cost = Part Supplier Cost / Part Supplier Quantity where the Part Id = Supplier Part Id and the Part Supplier is the primary supplier
  - If for some reason, a part supplier record is not found for a particular part, the SA Sync Part Cost is set to the Part Purchase Cost

Ensure the Vendor API Calls use the same locking as SedonaOffice Client for Vendor Concurrent Users

Vendor API Calls follow the same logic for locking as SedonaOffice. Refer to the SedonaOffice release notes enhancement "Vendor Concurrent User Activity Request".

If another user is editing the information attempting to be updated, the API call response will indicate that a data lock could not be obtained, the information could not be modified, and the call can be tried again later.

Multiple users with the proper permissions to edit and save associated vendor information (Bills, Credits, Payments, POs, Receipts, Returns, GL Journal, GL Accounts, Taxes, Custom Fields, Notes) can edit a single vendor at the same time.

We also made these changes:

- Marked all three existing LockTable endpoints as obsolete in swagger.

- LockTable/Check still returns lock information.
- The other two endpoints have been changed to do nothing and simply return "success"

Note: All three endpoints may be removed in a future version. These should not be used in new development and should be removed from existing development as soon as possible.

SedonaAPI 2.0 Application Corrections

Slow "get single Id" endpoints [110780]

We made these changes to resolve this:

- Increased Entity Framework global command timeout from 30 seconds to 5 minutes. This gives longer running queries more time to finish before timing out.
- Created new endpoint GET /api/CustomerSite/simple/{id} to add the ability to retrieve a much simpler customer site model. This endpoint will execute much faster than CustomerSite/{id}.
- Changed PUT /api/CustomerSite to return CustomerSiteSimple model.

Special Upgrade/Installation Instructions

If your company uses SedonaWeb 2.0/SedonaAPI 2.0, IT will update the version at the same time as your SedonaOffice upgrade. This is to ensure compatibility with all modules using SedonaWeb 2.0/SedonaAPI 2.0.

.NET Framework 4.8 automatically uses TLS 1.2. TLS 1.2 must be the only TLS version Enabled in the Registry. TLS 1.0 and TLS 1.1 must be disabled. Verify they are disabled, and TLS 1.2 is enabled.

Supported Environments

- This version of SedonaWeb 2.0/SedonaAPI 2.0 requires SedonaOffice version 6.2.0.18 or above.
- Server (where SedonaWeb 2.0 is installed) has Microsoft .NET 4.8 installed.