

SedonaAPI and SedonaWeb 2.0 - 1.41.0 - 3/9/2022

01/05/2024 12:02 pm EST

Special Upgrade / Install Instructions:

If your company uses the SedonaCloud API, IT will update the SedonaCloud version at the same time as your SedonaOffice upgrade. This is to ensure compatibility with the all modules utilizing SedonaCloud.

Supported Environments:

Version Support

** This version of SedonaCloud requires SedonaOffice version 6.2.0.4 or above **

Minimum System Requirements:

- Server is on Microsoft .Net 4.6.1

Features:

New Endpoints created:

Retrieve OPT Avalara tax group data to support Avalara Tax Lookups.

- `api/optavalarataxgroup/{id}`
- `api/optavalarataxgroup/search`

Return all AR_RMR_Tracking records.

`api/CustomerRecurring/GetRMRTrackingHistory`

Utilizes the capture function in SedonaOffice, populating it with Forte Payments.

`/api/paymentmerchant/captureauthorizedtransaction`



See [technical documentation](#) for further information.

Move Reporting from FSU Tech Tool to SedonaCloud

Added endpoint POST `/api/fsu/servicereport`. This asks the API to generate the requested service report and add it to the documents of the specified entities. The body of the post must contain the following parameters:

- `DispatchId` - This is the `SV_Service_Ticket_Dispatch.Dispatch_Id` of the appointment on which the technician is

working.

- DocumentTypes - This is a bit-coded value that specifies to which entities the document will be added. The desired values are added together. For example, to add to customer site (2) and system (4), set DocumentTypes to 6. At least one entity must be specified.
 - Bit 0 (value of 1) - Add to the service ticket documents.
 - Bit 1 (value of 2) - Add to the customer site documents.
 - Bit 2 (value of 4) - Add to the customer system documents.
- ReportType - This indicates which report to generate. Currently, the only valid value is 1, which is the Service Ticket Invoice report.
- ServiceTicketId - This is the SV_Service_Ticket.Service_Ticket_Id for which the report will be generated.
- TaxBreakout - This specifies whether or not the tax amounts of individual items should be shown on the invoice.



[See technical documentation for further information.](#)

Added Branch Branding

Provided the ability to add branding to branches for companies that want to customize their branding for each branch. If there is at least one branch in a company, selecting Preferences from the menu will show company/branch preferences. If there are no branches, this screen will only show "Preferences" for the company).

Note: Only a company-level web user will be able to see/edit branch-level preferences.



[See technical documentation for further information.](#)

Updated GET documents by ID to include the document file data.

Updated all the GET /api/documents/<ownertype>/bydocumentid/{id} endpoints to include the FileData Contents.

Bugs:

Note: Development Item Number in Parenthesis (), Customer Case Number in brackets []

Error received on service ticket if the system on the service ticket originated from Sales Automation (20711) [Internal]

Issue: When a new system is created from a proposal in SA and the proposal is completed, the system is pushed to the SO database with AR_Customer_System.InspectionNotes = NULL. In SedonaWeb, when a Service ticket is created with the system generated from the SA proposal, the service ticket is created, but opening the Service Ticket from SedonaOffice an error is received, "An error occurred opening the service ticket dialog".

Solution: Added a check to InspectionNotes property on CustomerSystem object to return Empty String if null is detected. Writes to the database as an empty string.

201 Created returned on /api/CustomerBill when record is not being created. (21507) [24046]

Issue: Using the /api/customerbill, the post will go through and return a result of 201 Created when the record failed to create.

Solution: Found that the transaction commit flag was not being set to true, so the transaction was never committed to the database. This has been corrected.

Additional corrections to /api/CustomerBill:

Also found that the PUT endpoint would allow updating the Is_Primary flag from Y to N. One of the customer's bill-to records must be marked as primary, and this would cause none of them to be primary.

- If there is only one bill-to, the update is not allowed.
- If there are two bill-to accounts, it will set the one being updated to N and the other to Y.
- If there are more than two bill-to accounts, the update is not allowed. The only way to properly change that is to update the record that *should* become the primary (change from N to Y). This also ensures the previous primary gets set to N.

Also found that updating a bill-to would always cause all invoice print/email checkboxes to be selected even if they were not prior to the update. This happened because those fields were not in the CustomerBill model and were always getting set to Y in the database upon any update via the API. Added those properties to the model so they can be updated and will maintain the desired settings.

Also added validation that an email address has been supplied if any of the 4 "invoice emailed" fields are true.

Also added validation of email address format (<something>@<something>.<something>).

Also found that the last bill-to account of a customer could be deleted. This should not be allowed and has been corrected.

Also found that deleting the primary bill-to did not set another as the primary. This has been corrected by setting the customer's lowest remaining customer_bill_id as the primary.

Wrong Data Displayed - showing Customer ID not Customer Number (21578) [18149]

Issue: Settings/Manage Customer Logins shows the customer_id and not the customer_number this is not helpful to the customer, this account_id is not visible to the Sedona user in the SedonaOffice application.

Solution: Changed to display Customer Number instead of Customer ID in Manage Customer Logins and in Sub Account screen. Change Customer # heading to Customer Number in Sub Account screen.

Job API POST allows job to be created without a JobCode. (21994) [20261, 34107]

Issue: Using Job API to POST new record it will allow a record to be created without sending the JobCode.

Solution: Found that the API was kind of trying to get the next job number when the passed-in job code was "0", but did not work. Corrected the functionality so it actually gets the next job number when a job code of blank or "0" is passed in.



See [technical documentation](#) for further information.

API/Dispatch/Search and API/Appointment/Search endpoints do not appear to use StartTime and EndTime (22683) [32021]

Issue: Api/Dispatch/Search & Api/Appointment/Search endpoints do not appear to use the StartTime and EndTime passed in the DispatchSearch model to limit the returned dataset

Solution: Implemented the use of start and end times passes in search parameters to only retrieve records in that range.

API Issue with PUT in api/documents (22768) [28593]

Issue: PUT /{companPrefix}/api/documents/{id} does not update the document

Solution: PUT (update) was only implemented for vendor documents and was expecting the document itself in the PUT request. Updating a document is now implemented for all document owners but can only update the UserDescription, SecurityLevel, and Document Type (similar to editing in the SO client), and requires the owning entity (SedonaDocumentEnum) and DocumentId to be updated to be included in the request. The remaining

incoming properties are ignored.



See [technical documentation](#) for further information.

API Get Payments failed Error: read ECONNRESET (22769) [25014]

Issue: Get Payments, failed repeatedly with either a 500 error (Unhandled Exception) or we got no response from the API, meaning something somewhere closed the connection before it was able to respond.

Solution: Corrected .../app/payments endpoint to return results instead of "connection reset" error.

API Part Sync Request is not returning full parts/panels list in a single request. (24375) [Internal]

Issue: Sales Automation parts sync on a clean database requires two complete sync executions to fully sync all parts from the SedonaOffice database. The first pass is only bringing in SO panels and the second pass is bringing in the parts.

Solution: The PartsController.Syncable() request needed a filter.Count == 0 condition added to the logic to determine if the query needed to gather all parts or just part ids specified in the filter.

JobDocuments not showing in SedonaOffice (24543) [Internal]

Issue: When saving a Job Document using the SedonaCloud API, the SystemID is being left null in the table. When this happens, SedonaOffice is not loading the document correctly.

Solution: When saving Job Documents the SystemId will now be set to 1.

Sedona API Post Job Parts does not have validation on customer. (25908) [34580]

Issue: The Post will be accepted with there is an incorrect customerID on the record.

Solution: Added validation on the Customer ID.

Sedona-X Mobile/API adding parts returns Error the WIPRegisterID 0 is invalid. (26042) [41239]

Also corrects: Sedona-X Mobile bug - Adding part in Sedona-X mobile corrupts part in SedonaOffice (26424) [38616]

Issue: The WIPRegisterId 0 is not valid is returned when trying to POST Service Ticket Parts records in Sedona-X mobile or through the API.

Solution: Corrected parameters that were incorrect.

For POST or changing the part by PUT, PartId and/or PartCode must be supplied. PartId takes precedence.

- If PartId > 1, it must reference an existing part.
 - POST: If PartCode is also supplied, the part code of the part referenced by PartId must match the supplied PartCode.
 - PUT: The incoming PartCode is ignored.
- If PartId is empty or < 2, PartCode must be supplied and reference an existing part.

When updating an existing entry (via PUT), incoming properties with null values will retain the existing value currently in the database. To blank out an existing value (Location, for example), an empty string must be supplied.

Sedona-X Mobile/API adding parts returns error; "The InterCompanyRegisterID 0 is invalid. (26249) [41401]

Issue: The InterCompanyRegisterID 0 is not valid is returned when trying to POST Service Ticket Parts records in SedonaX or through the API.

Solution: Same as solution for bug #26042.