
Sugar Professional 8.0.0 Release Notes

Sugar Professional 8.0.0 Release Notes	3
Overview	3
Administrator and End User	3
Data Privacy Enhancements	3
Additional Enhancements	4
Feature Enhancements When Upgrading From 7.9	5
Data Privacy Enhancements	5
Additional Enhancements	6
Fixed Issues When Upgrading From Winter '18 (7.11)	7
Fixed Issues When Upgrading From 7.9	9
Known Issues	12
Developer	21
Developer Notes for Upgrading From Winter '18 (7.11)	21
Developer Notes for Upgrading From 7.9	30
Supported Platforms	70
Upgrade Paths	71

Sugar Professional 8.0.0 Release Notes

Overview

This document describes the changes and functionality available in Sugar Professional 8.0.0. Sugar 8.0.0 is available for both SugarCloud (Sugar-hosted SaaS) and on-site deployments.

Customers who are upgrading to 8.0.0 from Winter '18 (7.11) should refer to the following sections:

- [Feature Enhancements When Upgrading From Winter '18 \(7.11\)](#)
- [Fixed Issues When Upgrading From Winter '18 \(7.11\)](#)
- [Developer Notes for Upgrading From Winter '18 \(7.11\)](#)
- [Known Issues](#)

Customers who are upgrading to 8.0.0 from 7.9 should refer to the following sections:

- [Feature Enhancements When Upgrading From 7.9](#)
- [Fixed Issues When Upgrading From 7.9](#)
- [Developer Notes for Upgrading From 7.9](#)
- [Known Issues](#)

For more information on the changes in this release, please refer to the [What to Expect When Upgrading to 8.0](#) article.

Administrator and End User

Feature Enhancements When Upgrading From Winter '18 (7.11)

The following feature enhancements are available in version 8.0.0 when upgrading from the Winter '18 release, 7.11:

Data Privacy Enhancements

New features to support data privacy practices, including adherence to the EU's GDPR, have been added in this release as follows:

- [Data Privacy module](#) : The new Data Privacy module allows users to record and track data privacy requests (e.g. data subject request to be forgotten) and events (e.g. gaining consent from the data subject).

-
- [Data Privacy Manager role](#) : A new out-of-the-box role provides regular users the ability to close data privacy records and permanently erase personal information.
 - [Personal information fields](#) : Fields can now be designated as personally identifying information which enables them to be permanently erased and included in the record's View Personal Info option.
 - [Permanent erasure](#) : Users in the Data Privacy Manager role now have the ability to immediately and permanently erase personal information fields from their Sugar database via an erasure request in the Data Privacy module.
 - [View Personal Info screen](#) : The new View Personal Info option displays the record's personal information fields along with the current field value, source of the value, and date the value was set.
 - [Consent tracking](#): The gaining or loss of consent can be tracked via the Data Privacy module. Users can record the business purposes to which an individual has consented in the new Business Purposes Consented For field on person-type modules (e.g contacts).
 - [Email opt-out default](#) : Using the new option available via Admin > System Email Settings, admins can configure whether new email addresses added to Sugar records default to opted-in or opted-out.
 - [Opt-in via web-to-lead forms](#) : An opt-in checkbox now gets added to web-to-lead forms containing email address fields to allow visitors to explicitly indicate that they wish to receive marketing emails.
 - [Disabling activity streams](#) : Admins now have the ability to stop activity streams from displaying or storing new entries via Admin > System Settings.
 - [Record change log enhancements](#) : Sugar's change log is now called "Audit Log" and includes the following improvements:
 - **Logging from creation** : Audit logs now capture the setting of fields starting at record creation.
 - **Source column** : Audit log now captures the source (e.g. Sugar user, Web to Lead) from which each field change originated.
 - [Email address audit log](#) : Email addresses now have a separate audit log which records when the address is marked as opted-out or invalid.
 - [Audit Log for the Targets module](#) : Users can now view a history of changes to audited fields for target records.
 - **Audit log header enhancement** : The audit log header now displays the name of the current record to better identify which record's audit log you are viewing.

Additional Enhancements

Enhancements to other areas of the application have been made as follows:

-
- **Product Catalog dashlet enhancements** : The following improvements have been made for the Product Catalog dashlet:
 - **ProductCatalog dashlet available in other views** : The Product Catalog dashlet can now be added to other Sidecar modules, as well as the home page, and will be read-only.
 - **Usability improvements** : For a better user experience, enhancements have been made to the dashlet including the addition of a scroll bar (for Chrome), a cursor that changes when hovering over actionable elements, and an updated font style to match Sugar's new look.
 - [Saved Reports Chart dashlet enhancement](#) : The Select a Report field now allows users to search and select their desired report when adding/editing the Saved Reports Chart dashlet.
 - [Configure API Platforms user interface](#) : Administrators now have the ability to add or remove custom API platform identifiers via Admin > Configure API Platforms.

Feature Enhancements When Upgrading From 7.9

The following feature enhancements are new in version 8.0.0 when you are upgrading from version 7.9:

Data Privacy Enhancements

New features to support data privacy practices, including adherence to the EU's GDPR, have been added in this release as follows:

- [Data Privacy module](#) : The new Data Privacy module allows users to record and track data privacy requests (e.g. data subject request to be forgotten) and events (e.g. gaining consent from the data subject).
- [Data Privacy Manager role](#) : A new out-of-the-box role provides regular users the ability to close data privacy records and permanently erase personal information.
- [Personal information fields](#) : Fields can now be designated as personally identifying information which enables them to be permanently erased and included in the record's View Personal Info option.
- [Permanent erasure](#) : Users in the Data Privacy Manager role now have the ability to immediately and permanently erase personal information fields from their Sugar database via an erasure request in the Data Privacy module.
- [View Personal Info screen](#) : The new View Personal Info option displays the record's personal information fields along with the current field value, source of the value, and date the value was set.

-
- [Consent tracking](#) : The gaining or loss of consent can be tracked via the Data Privacy module. Users can record the business purposes to which an individual has consented in the new Business Purposes Consented For field on person-type modules (e.g contacts).
 - [Email opt-out default](#) : Using the new option available via Admin > System Email Settings, admins can configure whether new email addresses added to Sugar records default to opted-in or opted-out.
 - [Opt-in via web-to-lead forms](#) : An opt-in checkbox now gets added to web-to-lead forms containing email address fields to allow visitors to explicitly indicate that they wish to receive marketing emails.
 - [Disabling activity streams](#) : Admins now have the ability to stop activity streams from displaying or storing new entries via Admin > System Settings.
 - [Record change log enhancements](#) : Sugar's change log is now called "Audit Log" and includes the following improvements:
 - **Logging from creation** : Audit logs now capture the setting of fields starting at record creation.
 - **Source column** : Audit log now captures the source (e.g. Sugar user, Web to Lead) from which each field change originated.
 - [Email address audit log](#) : Email addresses now have a separate audit log which records when the address is marked as opted-out or invalid.
 - [Audit Log for the Targets module](#) : Users can now view a history of changes to audited fields for target records.
 - **Audit log header enhancement** : The audit log header now displays the name of the current record to better identify which record's audit log you are viewing.

Additional Enhancements

Enhancements to other areas of the application have been made as follows:

- [New look and feel](#) : Updated colors and shapes appear throughout Sugar's Sidecar user interface.
- **Preview user interface** : Record previews in the intelligence pane have a more usable layout and updated look and feel.
- [Emails user interface](#): The Emails module now uses the Sidecar interface. It has been redesigned to store copies of email messages sent from Sugar or archived via one of the various email archiving options. Users can view and work with email records much like they view and work with records in other Sidecar modules.
- [Outgoing email accounts](#): Users can configure multiple SMTP accounts to send email from Sugar.
- [Unlink emails](#) : Explicitly related emails can be unlinked from the Emails subpanel of related records.

-
- [Drill-through report charts](#): Users can now click on a segment of a report's chart from the Reports module or Saved Reports Chart dashlet to view a filtered list view of records belonging to that segment and perform various actions (e.g. edit, mass update, refresh).
 - [Sharable dashboards](#) : Dashboards can now be shared with teams of users, pulled into their views by users, and pushed to teams of users by system admin users.
 - **Copying quote records** : Users can create quote records via duplication. The original quote's line items, groups, comments, and related account and opportunity data will carry over to the new quote.
 - **Product Catalog dashlet** : The Product Catalog dashlet is now available to access from the following views:
 - [Product Catalog dashlet for building quotes](#): Easily find and add products directly from the Product Catalog dashlet as you create and edit quote records.
 - **ProductCatalog dashlet in other views** : The Product Catalog dashlet can be added to other Sidecar modules, as well as the home page, and will be read-only.
 - [Saved Reports Chart dashlet enhancement](#) : The following improvements have been implemented for the Saved Reports Chart dashlet:
 - **Search and select reports** : The Select a Report field now allows users to search and select their desired report when adding/editing the Saved Reports Chart dashlet.
 - **View Report option** : The View Report option has been added to the Saved Reports Chart dashlet allowing users to access the originating report's detail view.
 - [Contracts user interface](#) : The Contracts module now uses the Sidecar user interface.
 - [Contract Types user interface](#) : The Contract Types module now uses the Sidecar user interface.
 - [Currencies user interface](#) : The Currencies module now uses the Sidecar user interface.
 - [Products- and Quotes-related modules user interface](#) : The Manufacturers, Shipping Providers, and Tax Rates modules now use the Sidecar user interface.
 - [Configure API Platforms user interface](#) : Administrators now have the ability to add or remove custom API platform identifiers via Admin > Configure API Platforms.
 - [SAML configuration enhancement](#) : Additional configuration options have been added for configuring SSO using SAML.
 - **Teams filter** : Multiple team selection is now supported when filtering based on the Teams field in Sidecar modules.
 - **Elasticsearch 5.4 support** : Elasticsearch version 5.4 is now supported.

Fixed Issues When Upgrading From Winter '18 (7.11)

The following issues are resolved in version 8.0.0 when upgrading from the Winter '18 release, 7.11. [Case Portal users](#) can use the following links for more details about each issue:

- [80579](#) : Logic hooks with class and function names matching in text, but not in case sensitivity, result in PHP fatal errors.
- [80489](#) : If Sugar's notification delay is set to exceed the limit of 35791 minutes, it may cause unexpected errors in the system.
- [80237](#) : When sending emails from a custom module's Emails subpanel, the related email record may not appear in the Emails subpanel as expected.
- [80160](#) : Users are prevented from sending transactional email messages via the Emails module to recipients who have opted out from marketing campaigns.
- [80113](#) : Filtering the Knowledge Base module using the Body field may cause a 500 error to occur.
- [79807](#) : When viewing knowledge base article attachments via the Notes module, the Related To field may display the incorrect label name (KBContentsAttachments) for the Knowledge Base module and not include the related knowledge base record.
- [79717](#) : The View Change Log option is not available in the Contracts record view actions menu.
- [79705](#) : Report charts do not respect user preferences for number formatting, currency symbol, or date and time formatting.
- [79486](#) : Attempting to edit the Shipping field for existing quote records may not work as expected if the record contains a calculated or dependent field.
- [79268](#) : Modules (e.g. Task) that have a custom relationship with the Activities module may not display email records as expected when emails are sent via the Emails subpanel.
- [79254](#) : An unexpected error may display while installing Sugar on a Windows server on PHP 7.1.
- [78808](#) : Downloading PDF templates containing ampersands may not work as expected and result in a 500 error.
- [78714](#) : Emailing a meeting record as a PDF file via the Email PDF option may improperly remove the meeting's relationship to the related account record.
- [78568](#) : Calculated fields containing the countConditional(), rollupConditionalSum(), or rollupSum() functions may improperly update and display incorrect values each time additional records are loaded in the related module subpanels.
- [78532](#) : In certain circumstances, the record's creation date may improperly default to 01/01/1970.
- [78496](#) : Report chart values may not render correctly for Integer fields

when using a period as the 1000s separator and a comma as the Decimal Symbol in the user's profile.

- [77764](#): Attempting to scroll in Legacy modules (e.g. Campaigns) may not work as expected when logged into Sugar on iPad.
- [73026](#): Workflow alert templates containing new and old value variables (e.g. `{::future::Contacts::do_not_call::}`, `{::past::Contacts::do_not_call::}`) for checkbox fields may not display the correct field values as expected in the email notification.

Fixed Issues When Upgrading From 7.9

The following issues are newly resolved in version 8.0.0 when you are upgrading from version 7.9. [Case Portal users](#) can use the following links for more details about each issue:

- [80579](#): Logic hooks with class and function names matching in text, but not in case sensitivity, result in PHP fatal errors.
- [80489](#): If Sugar's notification delay is set to exceed the limit of 35791 minutes, it may cause unexpected errors in the system.
- [80237](#): When sending emails from a custom module's Emails subpanel, the related email record may not appear in the Emails subpanel as expected.
- [80160](#): Users are prevented from sending transactional email messages via the Emails module to recipients who have opted out from marketing campaigns.
- [80124](#): Attempting to log into Sugar may fail with an error for instances using LDAP if the LDAP configuration (e.g. password) contains any HTML special characters (e.g. `<`, `>`, `&`).
- [80116](#): An error in the language file may cause MySQL errors.
- [80113](#): Filtering the Knowledge Base module using the Body field may cause a 500 error to occur.
- [80099](#): The RT GSync integration version 3.7 and higher may not pass the health check as expected.
- [80056](#): When creating a workflow action to create a record in a module associated with the target module (Leads), "Calls" may not be available to select from the modules dropdown list.
- [79972](#): Comments added to a group on the quote worksheet may improperly sort to the bottom of the worksheet after save.
- [79969](#): Changes made to a comment may not preserve as expected when adding another comment in edit mode.
- [79943](#): SAML authentication may not work as expected after upgrade for instances configured with an ADFS server.
- [79914](#): When logged into Sugar with the Spanish (Latin) language, the Accounts, Contacts, Notes, Reports, and Tasks module names display with lowercase characters (e.g. cuentas) on the navigation bar after upgrading

to 7.10.0.0.

- [79906](#), [79859](#) : Summation with Details-type reports containing an ampersand character (&) in the name may not render the report charts or results as expected.
- [79816](#) : Accessing quote records via the REST API may not work as expected.
- [79807](#) : When viewing knowledge base article attachments via the Notes module, the Related To field may display the incorrect label name (KBContentsAttachments) for the Knowledge Base module and not include the related knowledge base record.
- [79801](#) : The RT SalesMap integration version 4.3 and higher may not pass the health check as expected.
- [79734](#) : Drill through is not supported for report charts which are grouped by a relate field.
- [79717](#) : The View Change Log option is not available in the Contracts record view actions menu.
- [79707](#) : Clicking a URL link from an archived email's record view may cause the email body to improperly render blank. Navigating back to the email's record view will display the email body again.
- [79705](#) : Report charts do not respect user preferences for number formatting, currency symbol, or date and time formatting.
- [79486](#) : Attempting to edit the Shipping field for existing quote records may not work as expected if the record contains a calculated or dependent field.
- [79450](#) : Creating a call with a start date time (e.g. 9:10 am, 10:35 am) that is not available in the dropdown list may not block out the time period in the user's schedule as expected.
- [79329](#) : Outlook Plug-in version 2.4.0 may not connect to certain Sugar versions (7.9.1.0, 7.10.0.0).
- [79302](#) : Filtering on Decimal fields in list view search may not return search results as expected and result in a 500 error.
- [79268](#) : Modules (e.g. Task) that have a custom relationship with the Activities module may not display email records as expected when emails are sent via the Emails subpanel.
- [79254](#) : An unexpected error may display while installing Sugar on a Windows server on PHP 7.1.
- [79222](#) : If two modules contain calculated fields containing a related() function that point to each other, deleting one record may cause a new blank record to get created improperly.
- [79204](#) : Creating multiple quotes from a related module's Quotes subpanel (e.g. from an account record) incorrectly duplicates the original quote's line items as line items in each subsequent quote.
- [79076](#) : The quoted line item's Unit Price field in the Quotes module is uneditable.
- [78860](#) : Sugar SSO fails to connect to SAML login URL's containing an ampersand (&).

-
- [78808](#) : Downloading PDF templates containing ampersands may not work as expected and result in a 500 error.
 - [78714](#) : Emailing a meeting record as a PDF file via the Email PDF option may improperly remove the meeting's relationship to the related account record.
 - [78620](#) : The related record (e.g. Account Name field) links in a module's record view (e.g. Contacts) may incorrectly be disabled for users assigned a role with View permission set to "Owner".
 - [78568](#) : Calculated fields containing the countConditional(), rollupConditionalSum(), or rollupSum() functions may improperly update and display incorrect values each time additional records are loaded in the related module subpanels.
 - [78537](#) : Editing the run-time filter in reports may not auto-complete to pull up matches as expected.
 - [78532](#) : In certain circumstances, the record's creation date may improperly default to 01/01/1970.
 - [78496](#) : Report chart values may not render correctly for Integer fields when using a period as the 1000s separator and a comma as the Decimal Symbol in the user's profile.
 - [78364](#) : Sugar improperly attempts to establish an SSL/TLS connection when "Use SMTP Authentication" is unchecked.
 - [78306](#) : When importing calls or meetings, the current user performing the import is improperly added as a guest to the call/meeting.
 - [77764](#) : Attempting to scroll in Legacy modules (e.g. Campaigns) may not work as expected when logged into Sugar on iPad.
 - [77736](#) : User records may not be clickable from the Organization Chart dashlet to access the Employees detail view.
 - [77658](#) : Adding a custom time format in the config_override.php file may not work as expected and display the time values incorrectly.
 - [77481](#) : Manually clearing the cache directory may cause unexpected errors in the system if the Users module contains a custom field.
 - [77410](#) : After upgrading to Sugar version 7.7.1.0+, performing an import using the pre-set import settings may not map the Email Address field to the corresponding field (Email) as expected.
 - [77291](#) : Downloading PDF templates containing bold and/or italic text may not display the text in the correct font format.
 - [77255](#) : When resetting a password via the Reset Password email, including an ampersand character (&) in the new password may cause login issues for subsequent login attempts.
 - [77012](#) : When creating a custom module in module builder, adding a many-to-one relationship may incorrectly create a duplicate relationship field to the module after saving/deploying the package.
 - [76620](#) : Users assigned a role with "Access Type" set to "Admin & Developer" for "Users, Teams, and Roles" may not be able to reset passwords for other users as expected.
 - [76538](#) : When accessing Sugar using a device (e.g. laptop) with touch

screen enabled, the date picker may not work as expected.

- [76428](#): Sugar may improperly truncate the attachment name from inbound emails if it contains special characters in the name.
- [76183](#) : Sugar improperly attempts to send auto-reply emails using the Admin user's (id=1) email address instead of the system email address defined in Admin > Email Settings.
- [75571](#) : Enabling the "Delete related notes & attachments with deleted emails" option may not delete the related note record as expected when an email is deleted.
- [75471](#) : Sugar may not behave as expected and result in an error if the Sugar license is either expired or due to expire.
- [73026](#) : Workflow alert templates containing new and old value variables (e.g. {::future::Contacts::do_not_call::}, {::past::Contacts::do_not_call::}) for checkbox fields may not display the correct field values as expected in the email notification.
- [72921](#) : Setting the email client in the user's profile to "External Email Client" may not be respected properly as the Sugar email client continues to be used when replying to emails in Sugar.

Known Issues

The following are known issues in this release. [Case Portal users](#) can use the following links for more details about each issue:

- [80730](#) : Reports without charts are improperly available to select in the Saved Reports Chart dashlet.
- [80681](#): Making changes to a report's relationship-based filters may result in an error when running the report. As a workaround, re-create the report with the desired filter without making any changes to it.
- [80605](#) : Integrations and customizations using the REST v4.1 API may no longer work as expected for instances running on PHP 7.1.
- [80583](#) : Attempting to erase fields (e.g. Description) marked as "Personal Information" from the Opportunities module do not work as expected and result in a 500 error.
- [80376](#): Uninstalling custom modules from Sugar may not delete the associated workflows as expected. As a workaround, remove the affected workflow via the database.
- [80246,78121](#) : In certain circumstances, having calculated fields that are marked as required may cause unexpected behavior in the record.
- [80091](#) : Creating a dashboard may not work as expected and result in an error for users without private teams. Navigating to Admin > Repair and running "Repair Teams" will help resolve the issue.
- [80083, 79251](#) : In certain circumstances, performing a list view search may not return correct results as expected.

-
- [80002](#) : Generating PDFs using previously existing PDF templates may not display data as expected after upgrading to Sugar versions 7.9 or higher.
 - [79947](#) : Calculated fields may not populate when the record has multiple calculated fields using the related() function to the same module (e.g. related(\$accounts, field1) and related(\$accounts, field2)).
 - [79888](#) : Email accounts configured as a group inbound mailbox may not be used to send emails even though the "Allow users to send emails using the "From" Name and Address as the reply to address" option is enabled.
 - [79777](#) : The billing address and shipping address for quotes created via the "Quotes (Bill To)" or "Quotes (Ship To)" subpanel may appear incorrectly if the related record's (e.g. Accounts) billing and shipping address is different.
 - [79763](#) : The Account Name field does not get populated as expected for quoted line items related to a quote.
 - [79716](#) : The Contracts list view, record view, and subpanel do not have the option (star icon) to designate records as favorites.
 - [79715](#) : The Follow button does not appear in the Contracts record view as expected.
 - [79712](#) : The "Sign" and "Get latest" links do not appear as expected in the Documents subpanel of the Contracts module.
 - [79709](#) : Single sign-on authentication may fail when attempting to log into Sugar using Internet Explorer.
 - [79704](#) : When logged into Sugar with certain languages (e.g. Russian), the list view's Record Actions menu may not appear as expected for some modules (e.g. Dashboards).
 - [79686](#) : The List Order field in the Contract Types, Manufacturers, Tax Rates, and Shipping Providers modules does not control the order in which the options are listed in the corresponding fields (Type Name, Manufacturer Name, Shipping Provider, Tax Rate) for the Contracts, Quotes, and Product Catalog modules.
 - [79643](#) : Inbound emails containing emojis may not get imported into Sugar as expected.
 - [79492](#) : Setting the log level to "Error" may result in multiple relationship errors being written to the log file.
 - [79469](#) : When a web-to-lead form is created without including a redirect URL, the visitor may improperly get directed to an error page upon submitting the form.
 - [79453](#) : Emails that were previously archived to contact records, as well as newly archived emails to contacts, may not appear in the related account's Emails subpanel after upgrade to 7.9.x. As a workaround, select the related account when archiving an email to a contact using the Outlook plug-in.
 - [79344](#) : The quoted line item total may not respect the user's preferred currency as expected.
 - [79244](#) : Certain stock modules may be improperly hidden after converting from Sugar Professional 7.9.1.0 to Sugar Ultimate 7.9.1.0.
 - [79173](#) : When attempting to navigate away from the module or save the

record, the Unsaved changes warning message may unexpectedly appear for modules containing custom dependent fields.

- [79166](#) : Session locking may cause performance issues in Sugar.
- [79156](#) : When a Summation Report with Details is printed to PDF, there may be missing display columns (e.g. Name) on the PDF file.
- [79131](#) : When the "Listview items per page" setting in Admin > System Settings contains a large value (e.g. 50 or greater), it may cause an issue with rendering the "Download PDF" and "Email PDF" options in the record view's actions menu. Changing the "Listview items per page" setting to "20" may help resolve the issue.
- [79108](#) : When editing a record via the list view preview on the intelligence pane, the Resolve Conflict drawer may appear unexpectedly upon save.
- [78890](#) : Updating composer in instances with custom modules deployed from module builder may cause unexpected errors.
- [78850](#) : The Saved Reports Chart dashlet may not render the chart as expected for certain reports when the Bar chart value placement field is set to an option other than "None" or "Total". Selecting "None" or "Total" for the Bar chart value placement field will render the charts properly in the dashlet.
- [78845](#) : Attempting to add a code library to Sugar may cause the composer update process to fail and not update as expected.
- [78790](#) : Performing certain actions that rebuild the cache files (e.g. clearing the cache directory) may cause unexpected errors in the system.
- [78736](#) : Out-of-the-box dashboards may incorrectly be recreated after deleting it from the home page or intelligence pane.
- [78719](#) : Users may encounter an unexpected behavior when accessing Sugar if the Date Modified field in the user account contains the same value as another user.
- [78709](#) : Users assigned a role with Delete, Edit, or Export permission set to "Owner" may improperly be restricted from downloading and emailing PDFs.
- [78668](#) : In certain circumstances, the tooltip (e.g. Create) may continue to persist improperly while navigating through Sugar. Reloading the web browser will clear the tooltip from the screen.
- [78667](#) : Attempting to scroll in Sidecar modules (e.g. Meetings) may not work as expected when logged into Sugar on iPad.
- [78600](#) : Special characters are improperly allowed to be entered in dropdown lists' item names.
- [78580](#) : Saving a record without completing the Salutation field which is marked as required in Admin > Studio may result in unexpected behavior.
- [78570](#), [78509](#), [78507](#), [76151](#) : When accessing Sugar using Internet Explorer or Firefox, attempting to clear a saved filter from the list view or subpanel search may not work as expected. As a workaround, select the All "Module Name" option from the Filter options list to clear the applied filter.
- [78547](#) : When creating a meeting or call assigned to another user, removing the current user from the Guests panel may not remove the user

as expected upon saving the record.

- [78541](#) : Dropdown fields set with ReadOnly dependencies may improperly be allowed to be edited if the field contains a dependent visibility formula.
- [78527](#) : Inline editing a TextArea field via the subpanel may not work as expected. Reloading the web browser will resolve the issue and allow the user to inline edit the field properly.
- [78487](#) : When renaming modules via Admin > Rename Modules, only the most recent changes will remain and any previous updates to module names will be incorrectly removed after save.
- [78471](#) : Setting the default date format in the user's profile to "mm/dd/yyyy" may cause the time periods on the Forecasts Settings page to incorrectly display "Invalid date".
- [78334](#) : Performing certain actions in records containing calculated fields with rollup functions (e.g. rollupSum) and a large number of related records may cause performance issues in Sugar.
- [78272](#) : Populating an amount field during record creation via the REST version 10 API causes the record to fail to appear in listview as expected when filtering on the amount field's value. As a workaround, update the currency_id field to "-99" via the REST API, trigger a save event via the user interface, or update the field directly via the database.
- [78229](#) : Downloading the import file template may take longer than expected for module's containing a large number of records.
- [78224](#), [73891](#) : Attempting to reorder subpanels in the module's record view may not work as expected when logged into Sugar using Chrome on Windows touchscreen devices. As a workaround, disable the "Enable Touch Events" option for Chrome and users should be able to drag and drop subpanels accordingly.
- [78198](#) : Activity stream posts may incorrectly display system labels for instances where the system default language is set to a non-English language. As a workaround, navigate to Admin > Locale and change the language to "English (US)", perform a "Quick Repair and Rebuild", then switch the default language back to your desired language.
- [78153](#) : Changing the default value for the Popup Reminder Time field for calls or meetings may not work as expected and continue to display the old value.
- [78128](#) : For dropdown list values, a value's Display Label will improperly revert to a blank value if its Item Name is 0 (zero).
- [78093](#) : Reports filtered by date/time fields (e.g. End Date) may generate incorrect results if the user's timezone has changed.
- [78065](#) : Moving a stock field between the columns (e.g. Hidden, Available) for the List View and Subpanel layouts in Studio may result in unexpected behavior.
- [78031](#) : Setting a default value for a field during the import process may not preserve the value in the field as expected.
- [77979](#) : Using the v4 SOAP API to retrieve lists of records may experience performance issues when tags are heavily used in the instance.

-
- [77820](#) : The Visibility Editor window may not load as expected and display an error if there is a large number of dropdown dependencies defined in the visibility editor.
 - [77792](#) : Enabling Full Text Search for the Date Created field in a module (e.g. Leads) may cause unexpected errors when performing a global search filtered by the module.
 - [77780](#) : Instances using MS SQL may see unexpected behavior due to a lack of ORDER BY clause in the list view query.
 - [77738](#) : Attempting to merge two records (e.g. accounts) may fail with an error if the record that is being merged to the primary record contains a large number of related records (e.g. contacts).
 - [77648](#) : Removing the Name field from layouts in Studio may cause the hyperlink attribute ('link' => true,) to get removed from the field. To resolve the issue, add 'link' => true, to the Name field array in the relevant layout file within your custom directory, then perform a quick repair and rebuild.
 - [77609](#) : Generating reports with empty relate fields may not include the associated record in the report result as expected if the related record has been deleted.
 - [77596](#) : The Emails subpanel may not display more than five email records for Sugar instances running on MSSQL databases.
 - [77549](#) : Users may experience performance degradation when user records are modified.
 - [77539](#) : Changing the boost value for a field (e.g. Name) may not affect the relevancy of the field for full text search, resulting in unexpected behavior.
 - [77372,77612](#) : When logged into Sugar using the Chrome browser, highlighting/selecting text in certain fields (e.g. Description, Name) may cause issues loading the browser and display an error. As a workaround, edit the field without highlighting any text to make the necessary changes.
 - [77302](#) : Upgrades may fail due to queries posted by the upgrade exceeding the max_allowed_packet database setting.
 - [77287](#) : Performing certain actions (e.g. import, mass update) in Sugar may result in performance issues if there are numerous calculated fields to be updated in related records. As a workaround, add the following line to the config_override.php file to disable the related calculation field updates: `$sugar_config['disable_related_calc_fields'] = true;`. But keep in mind that the affected calculated values will not be updated and running Recalculate Values on related records.
 - [77249](#) : Guests may not get imported to call or meeting records as expected.
 - [77205](#) : Viewing a page (e.g. list view) in Sugar without performing any actions for a period of time (e.g. 3 minutes) may cause an error message (Request Timeout: The server is not responding. Check your internet connection.) to appear unexpectedly.
 - [77162](#) : When Sugar is deployed on a Linux OS, uploading packages in Module Loader may fail with an error when .PNG files are detected by the

package scanner.

- [77087](#) : When a record is assigned to the user's default private team, changing the Teams field from the private team to another team (e.g. Global) may incorrectly display the team name with the user's last name appended to the end (e.g. Global Smith).
- [77055](#) : Attempting to mass update the user's outbound email client via Admin > User Management may not work as expected.
- [76985](#) : When performing a global search, filtering by a module then hitting the Enter key after entering the search term may cause a 500 error to occur.
- [76783](#) : When a record view panel with a Sugar Logic dependency formula contains a required Date field, the field may improperly default to 01/01/1970 when the panel appears.
- [76730](#) : Field-level validations may not work as expected to trigger field validation messages upon saving the record.
- [76524](#) : Dependent fields may not appear in the layout (e.g. Record View) as expected if the parent field (e.g. parent dropdown) is not present.
- [76401](#) : The data in the report chart may be inconsistent between the report chart dashlet and the Reports module.
- [76126](#) : When entering a dependency formula for a DropDown-type field, clicking the Cancel button on Studio's formula builder may not close the window as expected and result in an error.
- [76114](#) : When importing products, the defined currency on the import file may not be respected properly and cause the currency values to be calculated incorrectly.
- [76000,75999](#) : Setting the Collation option to "utf8mb4_general_ci" via Admin > Locale may cause unexpected errors.
- [75398](#) : Workflow alert may not trigger as expected if the processing order for the workflow is set to "Actions then Alerts". As a workaround, changing the workflow's processing order to "Alerts then Actions" may help resolve the issue.
- [75258](#) : Mass updating or importing records (e.g. cases) that are related to records (e.g. accounts) with large datasets may result in PHP memory errors.
- [75254](#) : Printing reports (e.g. Summation With Details report) to PDF may not work as expected when logged into Sugar via a mobile browser.
- [74919](#) : Performing certain actions (e.g. Quick Repair and Rebuild) in Sugar that rebuild the cache files may cause unexpected issues in the system if there are multiple users logged in and utilizing Sugar. As a workaround, perform such actions during off-hours where users are not utilizing the system.
- [74893](#) : Opportunities with expected closed dates prior to a certain time period (e.g. 1/1/1970) may cause the Forecast Bar Chart dashlet to not display as expected.
- [74887](#) : When there are a large number of filters (greater than 20) created by the same user for a module (e.g. Accounts), the corresponding list view

dashlet may not behave as expected.

- [74713](#) : Designating an email address as the "Reply to" address in the user's profile may not save as expected to preserve the settings.
- [74628](#) : Certain workflows using a Relate-type field in the condition may fail to load as expected and result in errors after upgrading to 7.6.x.x. As a workaround, run the following query in the instance's expressions table:

```
UPDATE expressions
SET     exp_type = "id"
WHERE   exp_type = "relate"
        AND lhs_field = "assigned_user_id"
```

- [74539](#) : Custom fields may not display as expected when previewing call and meeting records via the intelligence pane.
- [74416](#): Creating a Summation-type report that is grouped and sorted by a certain field (e.g. Month: Expected Close Date, Quarter: Expected Close Date) in a related module (e.g. Opportunities) may cause the run-time filter to not work as expected.
- [74382](#) : The Case Summary dashlet may not work as expected and cause an internal server error if the account record has a large number of related cases.
- [74354](#) : Pie charts may not display properly in report results if there are labels with a large number of characters.
- [74350](#) : An unexpected error may occur when saving a record if there is an issue with the user's default team in the database. As a workaround, run the following query in the instance's team sets table. The affected users will then need to edit their profile to configure their default teams again.

```
UPDATE team_sets
SET     deleted = 1
WHERE   id NOT IN ("select team_set_id from team_sets_teams where
        deleted = 0")
        AND deleted = 0
```

- [74349](#) : Creating and saving a record assigned to a team that the current user is not associated to may display an error even though the record gets created successfully.
- [73912](#) : Certain reports may not generate as expected if the last group-by field is a date (e.g. Opportunities > Month: Expected Close Date) and the chart type is set (e.g. Horizontal Bar). As a workaround, do not use the date field as the last grouping in the Define Group By step or change the chart type to "None".
- [73729](#), [72956](#) : Populating a target list by selecting a report for a module (e.g. Accounts) may not add all records as expected if the report contains a

large amount of data.

- [73689](#): When users adjust the list view or subpanel column widths, the user's preferred column size may not be preserved if the browser window is resized.
- [73569](#): Attempting to add a row to an existing quote record may not work as expected when accessing Sugar via certain browsers (e.g. Chrome, Safari) on iPad.
- [73566](#): Calculated or dependent fields containing a related() function may not get calculated until after save for activity-type modules (e.g. Notes).
- [73468](#): Time-elapse workflow may not trigger as expected when a date field (e.g. Expected Close Date) in the condition is set to a date in the future.
- [73434](#): Custom entrypoints which require authentication may not work as expected when using session tokens.
- [73346](#): URLs generated for iFrame or URL fields may improperly include special character encoding when the record was created via an API call.
- [73301](#): Attempting to edit Sugar Logic formulas with too many lines in the formula may cause an error ("Formula Builder: Error Loading Content") in Studio's formula builder. As a workaround, edit the formula to remove any line breaks.
- [73187](#), [70389](#): Tabbing while inline editing the address block in the record view does not work as expected as it does not tab through all the fields (e.g. state, postal code) within the address block.
- [72962](#): Dependent fields containing a related() function may cause the data to not display as expected in the module's list view.
- [72810](#): Filtering the list view search using custom checkbox fields may not work as expected.
- [72625](#), [71848](#): When a large number (e.g. 60) of PDF templates are available in a module, users may not be able to scroll through the full list of templates via the "Download PDF" or "Email PDF" options in the record view. As a workaround, changing the screen resolution or reducing the number of templates may help resolve the issue.
- [72590](#): The Due Date field may not be available to select when mass updating in the Tasks module. To resolve the issue, navigate to Admin > Studio > Tasks > Fields and disable the Mass Update option for the Due Date field then enable it again.
- [72581](#): Attempting to merge records in modules containing required dependent fields may not work as expected.
- [72409](#): When generating a report, labels may improperly overlap in the chart (e.g. Vertical Bar chart) causing the labels to be unreadable.
- [72286](#): The Opportunity module's amount fields cannot accept negative values.
- [72264](#): Clicking on the opportunity's name in the Quotes detail view after downloading the quote to PDF improperly redirects to the home page instead of the opportunity's record view. As a workaround, right-click on the opportunity name after downloading the quote to PDF and the

opportunity's record view will open accordingly.

- [72022](#) : The Projects module is incorrectly available to select via Admin > Mobile.
- [71950](#) : Adding TinyMCE to a TextArea-type field (e.g. Description) may cause the field to not display properly in record view when accessing Sugar via certain browsers (e.g. Firefox).
- [71835](#) : The Reset Password option in Admin > User Management may not work as expected and fail to send the system-generated password message to users with an email address from a domain consisting of 5 or more characters.
- [71733](#) : Printing archived emails via the browser's print option may not display correctly.
- [71446](#) : For opportunities, changes made to labels via Studio > Opportunities > Labels do not affect list view labels. Make changes to labels via Studio > Opportunities > Layouts > List View to modify list view layouts.
- [70940](#) : Attempting to disable the SAML authentication via Admin > Password Management may not work as expected if the authenticationClass property in config.php has been set to SAMLAuthenticate.
- [70542](#) : The "Show Completed Meetings, Calls, and Tasks" option is not available in Calendar > Settings, which can cause performance issues in the Calendar module for users who have a large number of activity records. Setting Admin > System Settings > vCal Updates Time Period to "-1" will relieve the performance delay, however, this will prevent user availability information from appearing when scheduling a call or meeting.
- [70024](#) : Scheduled reports may have incorrect or missing charts in the emailed PDF. Select "Print as PDF" from the report's Actions menu to see the updated chart.
- [69985](#) : If a custom module has a relationship with an individual Activity-type module (e.g. Tasks), this related module may not be available to select when creating a workflow action to create a record in a module (Tasks) associated with the target module (e.g. custom module). As a workaround, create a one-to-many relationship between the custom module and the Activities module, and the appropriate Activity-type module (e.g. Tasks) will be available to select when creating the workflow action
- [69982](#) : Old Value variables (e.g. {::past::Opportunities::sales_stage::}) inserted in workflow alert templates may not display the correct value in the email as expected.
- [69957](#) : Large number of activity stream records may cause performance issues.
- [69918](#), [69801](#) : Changing the style attributes for table tags, image tags, etc. on email templates via HTML may result in parts of the code getting removed upon save.
- [68985](#) : Custom relationships created between a module and the Activities module via Admin > Studio cannot be deleted as expected.

-
- [68975](#) : Changing the order of subpanels via Admin > Display Modules and Subpanels does not preserve the order upon save.
 - [68461](#) : Searching by non-primary email addresses in the module's list view (e.g. Accounts) does not pull up results as expected.
 - [68112](#) : Matrix-type reports display incorrectly when exported to PDF.
 - [68024](#) : The Case Number, Date Created, and Created By field values for the Cases module may not get populated in workflow alerts that are triggered on first save.
 - [67875](#) : Attempting to insert old value variables for related modules in workflow alert templates may not work as expected.
 - [67294](#), [66900](#) : Resetting the forecast time period does not set the current time period correctly and causes various issues in the application.
 - [67186](#) : The View Change Log in module records (e.g. Quotes, Opportunities) may incorrectly reflect the original system currency (US Dollar) for currency fields even though the default system currency has been changed (e.g. Euro) in Admin > Locale.
 - [66995](#) : The Display server response times option is improperly available to select via Admin > System Settings.
 - [66910](#) : The notification for Cases module does not include the case number as expected when triggered via the workflow.
 - [66842](#) : Relate fields cannot be sorted as expected in the list view.
 - [66086](#) : Embedded images may not display properly in archived emails.
 - [64880](#) : Certain fields (e.g. email address, phone) in the Users module may not be available to add as field variables in PDF templates.
 - [61338](#) : When test entries are deleted from the campaign, the related campaign emails that are sent to targets may not be viewable and display an error (Error retrieving record. This record may be deleted or you may not be authorized to view it).

Developer

- [Developer Notes for Upgrading From Winter '18 \(7.11\)](#)
- [Developer Notes for Upgrading From 7.9](#)

Developer Notes for Upgrading From Winter '18 (7.11)

The following changes in this release may affect developers in version 8.0.0 when upgrading from the Winter '18 release, 7.11:

- After upgrading to PHP 7.1, you may encounter `ArgumentCountError` errors in your PHP error log related to customizations or integrations. Make sure to check out the recent blog post on [PHP 7.1 pitfalls](#) to help

address these issues.

- Two sugarcrm command line tool commands have been added. Both commands trigger a reindex of CRM data in Elasticsearch; `silent_reindex` runs synchronously while `reindex` schedules a job that allows indexing to run in the background.
 - `search:reindex`
 - `search:silent_reindex`
- The meetings relationship between the Emails and Meetings modules has been removed in this release. It was reserved for SNIP service but it was not actually used and was causing data inconsistencies in certain situations.
 - Meetings are related to other Sugar records using a flex relate (parent) field. It is therefore possible to retrieve Meetings created by an imported Email (from SNIP) using SugarQuery or Filter API to select Meetings where `parent_type=Emails` and `parent_id=email_id`.
- The `full_name` field found on Person-based Sugar modules (e.g. Contacts, Leads, etc.) is equivalent to the `name` field found on all Sugar modules. In this release, updates have been made to convert uses of `full_name` in Sugar view metadata to use the `name` field instead. The `full_name` field is being deprecated in this Sugar release for simplicity's sake. Developers should convert to using the `name` field in place of the `full_name` field in your Sugar customizations. The `full_name` field will continue to work for compatibility reasons until it is removed in a future release.
- Due to a [MySQL performance optimizer bug](#), we recommend updating the config for your MySQL server to disable the 'block_nested_loop' optimization.
 - SQL:

```
SET SESSION optimizer_switch='block_nested_loop=off';
```

- MySQL cnf:

```
[mysqld]  
optimizer_switch=block_nested_loop=off
```

This can cause an increase in the use of temporary files by MySQL. You may need to increase the open files limit on your system if you encounter an error like the following on Linux: "Too many open files in system". For more information, refer to [this easy EasyEngine tutorial](#).

- The Sidecar framework used by the base Sugar web client as well as the portal web client now uses the v11_1 REST API version by default.
- The following Sugar Config setting has been added:
 - `$sugar_config['new_email_addresses_opted_out']` : If set to true,

newly created Email Address records in Sugar are opted-out by default. This setting can also be controlled using the "Opt-out new email addresses by default" setting in the System Email Settings administration panel. For more information on Sugar configurations, please refer to the [Developer Guide](#).

- This Sugar release depends on the [Gnu Multiple Precision](#) (GMP) PHP extension. Make sure all your Sugar hosting environments include the GMP PHP extension prior to using Sugar 8.
- Charts within Sugar now fully support system and user locale settings. As part of this change, the Sucrose library version was upgraded from v0.7.5 to v0.8.0. To learn more about working with locales in Sucrose please check out the [locale guide](#).
- Security Subjects mark the first use of Sugar Resource Names (SRNs). SRNs are intended to identify a resource uniquely across all SugarCRM-deployed cloud services. Sugar Resource Name is inspired by the [Uniform Resource Name](#) (URN) concept and is similar to but not the same as [Amazon Resource Names](#) (ARNs).

```
{
  "_type": "user",
  "_srn": "srn:local:crm:::user:1fbbcf60-4aac-4503-b2eb-
d00c3eb391db",
  "name": "John Doe"
}
```

- For Sugar 8, SRNs usage is limited to local resources within the Sugar instance such as the currently logged in user. Local resources use the following scheme: srn:local:crm:::resource-type:resource-id
 - Do not invent SRNs if you develop your own custom Security Subjects. Resource naming conventions and associated SRNs are the responsibility of SugarCRM.
- Erased fields are tracked in a new database table called `erased_fields`.
- Several fields on Person module types (Contacts, Leads, etc.) have been marked as containing personal information by default. The Person template metadata has been updated accordingly. Each of these fields is now audited by default for additional change tracking and marked as personal information to make them eligible for erasure via a data privacy request.
 - salutation
 - first_name
 - last_name
 - title
 - facebook
 - twitter
 - googleplus
 - phone_home

-
- phone_mobile
 - phone_work
 - phone_other
 - phone_fax
 - primary_address_street
 - primary_address_street_2
 - primary_address_street_3
 - primary_address_city
 - primary_address_state
 - primary_address_postalcode
 - primary_address_country
 - alt_address_street
 - alt_address_street_2
 - alt_address_street_3
 - alt_address_city
 - alt_address_state
 - alt_address_postalcode
 - alt_address_country
 - assistant
 - assistant_phone

It is important to note that a field **MUST** be audited in order for it to be marked as personal information.

- The following example would be invalid:

```
'pii_stuff' => array(  
    'name' => 'pii_stuff',  
    'vname' => 'LBL_PII_STUFF',  
    'massupdate' => false,  
    'type' => 'varchar',  
    'len' => 100,  
    'comment' => 'PII Stuff',  
    'audited' => false, // default value is false  
    'pii' => true,  
),
```

- This example is valid:

```
'pii_stuff' => array(  
    'name' => 'pii_stuff',  
    'vname' => 'LBL_PII_STUFF',  
    'massupdate' => false,  
    'type' => 'varchar',  
    'len' => 100,  
    'comment' => 'PII Stuff',  
    'audited' => true,
```

```
        'pii' => true,  
    ),
```

- As part of the new data privacy features, a new method has been added to SugarBean class. This method ensures that data for these fields are erased from the audit log as well. This helps ensure that the data is unrecoverable.

```
SugarBean::erase(FieldList $fields, $check_notify)
```

- The audit log data model has also been updated to better support tracking of events. A new Audit\EventRepository class has been introduced for finer grain tracking of audited changes. In particular, the EventRepository is used to track when field changes are associated with an update or erase event as well as the source (Security\Subject) of the change. A new audit_events table in local database is associated with the new EventRepository implementation.
- The audit endpoint for the REST API has been updated to include the source of the change as well as the event_type (either update or erase) information.

For example, in the following two audit log entries, the first shows an erasure done by a Sugar user in response to a data privacy request while the second shows an update originating from Sugar installer creating demo data in the system.

```
GET /rest/v11_1/<module>/:record/audit  
{  
  "next_offset": -1,  
  "records": [{  
    "id": "b3361fa0-31ea-11e8-b38a-b243f839f789",  
    "parent_id": "602e5946-3139-11e8-aa9f-df5dba0db27d",  
    "event_id": "b34525c2-31ea-11e8-b7a8-8564fb05c062",  
    "date_created": "2018-03-27T11:14:33-07:00",  
    "created_by": "1",  
    "date_updated": "2018-03-27 18:14:33",  
    "field_name": "phone_home",  
    "data_type": "varchar",  
    "source": {  
      "subject": {  
        "_type": "user",  
        "id": "1",  
        "_module": "Users",  
        "client": {  
          "_type": "rest-api"  
        },  
        "first_name": null,  

```

```

    "last_name": "Administrator",
    "name": "Administrator"
  },
  "attributes": {
    "platform": "base",
    "dp_request_id": "8daab426-31ea-11e8-8011-34930ab25f6c"
  }
},
"event_type": "erasure",
"created_by_username": "admin",
"before": null,
"after": null
},
{
  "id": "603c6130-3139-11e8-b0e3-e077fbb971b3",
  "parent_id": "602e5946-3139-11e8-aa9f-df5dba0db27d",
  "event_id": "602ed3a8-3139-11e8-8d43-6a8a4f4452e9",
  "date_created": "2018-03-26T14:02:43-07:00",
  "created_by": "1",
  "date_updated": null,
  "field_name": "email",
  "data_type": "email",
  "source": {
    "subject": {
      "_type": "installer"
    }
  },
  "event_type": "update",
  "created_by_username": "admin",
  "before": null,
  "after": {
    "id": "60226c80-3139-11e8-bc41-845f0fd96813",
    "email_address": "sales.vegan@example.tv"
  }
}]
}

```

- The birthdate field on Contacts, Leads, Targets modules is marked as personal information by default.
- SugarACLDataPrivacy is a new SugarACLStrategy class that is used with Data Privacy module. It permits only users with admin-level role access to the Data Privacy module to edit certain fields (e.g. fields_to_erase).
- Sugar REST API endpoints that return record information now support an additional request parameter called erased_fields. The filter and module APIs are examples of REST APIs that now support this parameter.

-
- When `erased_fields=1` then record responses will include an `_erased_fields` array.

```
"_erased_fields": []
```

- If fields have been erased on this record by a Data Privacy administrator, then they will be listed in this array.

```
"_erased_fields": [  
    "last_name"  
]
```

This will allow an integration or client to know unambiguously when a field has been erased at a data subject's request. Integrations must take steps to erase any copies of this record data. For example, here is an example API call and response:

```
GET /rest/v11_1/Contacts/602e5946-3139-11e8-aa9f-df5dba0db27d?erased_fields=1&fields=first_name,last_name,phone_home,phone_mobile
```

```
{  
  "id": "602e5946-3139-11e8-aa9f-df5dba0db27d",  
  "date_modified": "2018-03-27T11:14:33-07:00",  
  "first_name": "",  
  "last_name": "Dillion",  
  "phone_home": "",  
  "phone_mobile": "(289) 902-4666",  
  "locked_fields": [],  
  "_acl": {  
    "fields": {}  
  },  
  "_erased_fields": [  
    "first_name",  
    "phone_home"  
  ],  
  "_module": "Contacts"  
}
```

- A new erasure feature has been added to Sidecar Fields. A new "erased" field template has been added that is used when a field has been permanently erased as part of a data privacy request. This field template will display a "Value Erased" pill: `./clients/base/fields/base/erased.hbs`.
- As part of the new personal information functionality for data privacy, several new Sidecar components have been added.

- Sidecar Layout:
 - PiiLayout
- Sidecar View:
 - PiiHeaderpaneView
 - PiiView
- Sidecar Field:
 - PiinameField
- Sidecar plugin:
 - Pii
- The mark-for-erasure flow is implemented as a new action field on the `DataPrivacy.RecordView` that launches a new `MarkForErasureLayout` layout. The following Sidecar components were added in addition to the standard record and list layouts for the Data Privacy module. These new layouts support the mark-for-erasure use case.

- Sidecar Fields

Class name	Directory location
<code>DataPrivacyEraseField</code>	<code>./clients/base/fields/dataprivacyerase/</code>

- Sidecar Layouts

Class name	Directory location
<code>DataPrivacy.MarkForErasureLayout</code>	<code>./modules/DataPrivacy/clients/base/layouts/mark-for-erasure</code>

- Sidecar Views

Class name	Directory location
<code>DataPrivacy.MarkForErasureHeaderpaneView</code>	<code>./modules/DataPrivacy/clients/base/views/mark-for-erasure-headerpane</code>
<code>DataPrivacy.MarkForErasureView</code>	<code>./modules/DataPrivacy/clients/base/views/mark-for-erasure</code>
<code>DataPrivacy.MarkedForErasureDashlet</code>	<code>./modules/DataPrivacy/clients/base/views/marked-for-erasure-dashlet/</code>

- A new Sugar Config setting has been added. If set to true, newly created Email Address records in Sugar are opted-out by default. This setting can also be controlled using the "Opt-out new email addresses by default" setting in the System Email Settings administration panel.

```
$sugar_config['new_email_addresses_opted_out']
```

- Email address `invalid_email` and `opt_out` fields added to `emails_email_addr_rel` relationship vardefs. This is used with `EmailParticipants` beans.
- The `modules/Emails/clients/base/fields/email-recipients/select2-selection.hbs` has been updated to make sure opted-out email addresses are decorated properly.
- The `invalid_email` and `opt_out` fields are now included with `from/to/cc/bcc` collection field data for `Emails` module.
- The `EmailField` Handlebars templates (`detail`, `list`) and JavaScript controller have been changed to allow opted-out email addresses to be selected for use in transactional email.

```
./sugarcrm/clients/base/fields/email/detail.hbs
./clients/base/fields/email/list.hbs
./sugarcrm/clients/base/fields/email/email.js
```

- With the new ability to disable activity streams, there are behavior changes to some Sugar APIs that developers should know about.
 - When using the existing Sugar PHP functions `Activity::enable()` or `Activity::disable()` to temporarily change status of activity streams, please use the `Activity::restoreToPreviousState()` function to restore the previous state. For example, use this sequence to temporarily disable activity streams in custom code:

```
Activity::disable();
...
Activity::restoreToPreviousState()
```

- When Activity Streams are disabled, then calls to the following Sugar REST API endpoints will return HTTP 401 NOT AUTHORIZED errors:

```
GET /<module>/:record/link/activities
GET /<module>/:record/link/activities/filter

GET /<module>/Activities
GET /<module>/Activities/filter

GET /Activities
GET /Activities/filter
```

- The Sugar REST API now supports Major and Minor versioning as discussed in a previous [Sugar Developer blog post](#). This release introduces the `v11_1` REST API. While maintaining full compatibility with `v11`, the `v11_1` REST API adds some additional endpoints and features.
- It is now possible for an administrator to add or remove custom API

platforms using the Configure API Platforms screen in the admin panel. It is no longer necessary for an integration developer to create and deploy a [Platform extension](#) into customer instances to allow use of [unregistered API platforms](#). This new admin functionality does not interfere with existing platform extensions that may be installed on Sugar instance.

- Dates on SugarBean objects should now be consistently formatted in the DB format instead of user-selected format. Developers should ensure their code relies on DB format when working with date information on SugarBean objects.
- The SugarAutoloader has been optimized to improve application performance. In particular, testing has revealed that the `./cache/file_map.php` cache introduces unnecessary overhead in terms of average memory usage and request response time. The Sugar Autoloader no longer maintains a map of each PHP file used by Sugar. As part of this change, the following SugarAutoloader methods are now deprecated and will be removed in an upcoming release.

Deprecated API	Alternative API
<code>SugarAutoLoader::fileExists()</code>	Use <code>file_exists()</code> instead.
<code>SugarAutoLoader::loadFileMap()</code>	Unnecessary. No alternative needed.
<code>SugarAutoLoader::addToMap()</code>	Unnecessary. No alternative needed.
<code>SugarAutoLoader::delFromMap()</code>	Unnecessary. No alternative needed.
<code>SugarAutoLoader::unlink()</code>	Use <code>unlink()</code> * instead.
<code>SugarAutoLoader::touch()</code>	Use <code>sugar_touch()</code> * instead.
<code>SugarAutoLoader::put()</code>	Use <code>file_put_contents()</code> * or <code>sugar_file_put_contents()</code> * instead.
<code>SugarAutoLoader::saveMap()</code>	Unnecessary. No alternative needed.

Please be reminded that these original APIs and these alternatives are all restricted in SugarCloud deployments. Avoid file system I/O in your Sugar code customizations.

- The following REST API endpoints were added in this release.

API Endpoint	Summary
<code>/rest/v11_1/<module>/:record/pii</code>	Retrieves current values for PII fields on the given record including the source of this data.

Developer Notes for Upgrading From 7.9

The following changes in this release may affect developers in version 8.0.0 when upgrading from version 7.9:

-
- After upgrading to PHP 7.1, you may encounter `ArgumentCountError` errors in your PHP error log related to customizations or integrations. Make sure to check out the recent blog post on [PHP 7.1 pitfalls](#) to help address these issues.
 - Two sugarcrm command line tool commands have been added; Both commands trigger a reindex of CRM data in Elasticsearch; `silent_reindex` runs synchronously while `reindex` schedules a job that allows indexing to run in the background.
 - `search:reindex`
 - `search:silent_reindex`
 - The meetings relationship between the Emails and Meetings modules has been removed in this release. It was reserved for SNIP service but it was not actually used and was causing data inconsistencies in certain situations.
 - Meetings are related to other Sugar records using a flex relate (parent) field. It is therefore possible to retrieve Meetings created by an imported Email (from SNIP) using SugarQuery or Filter API to select Meetings where `parent_type=Emails` and `parent_id=email_id`.
 - The `full_name` field found on Person-based Sugar modules (e.g. Contacts, Leads, etc.) is equivalent to the `name` field found on all Sugar modules. In this release, updates have been made to convert uses of `full_name` in Sugar view metadata to use the `name` field instead. The `full_name` field is being deprecated in this Sugar release for simplicity's sake. Developers should convert to using the `name` field in place of the `full_name` field in your Sugar customizations. The `full_name` field will continue to work for compatibility reasons until it is removed in a future release.
 - Sugar's Input Validation Framework is now enabled by default. The default value for `validation.soft_fail` is set to `false` starting with this Sugar release. Input validation violations are reported as FATAL errors in the `sugarcrm.log` and PHP exceptions are thrown which will result in HTTP 500 responses to these invalid requests.
 - Sugar `auto_increment` field values are now automatically populated on saved SugarBean objects. This will mean that updated auto-incremented values will be available without having to run additional queries.
 - `Administration::retrieveSettings()` could previously return JSON encoded objects instead of native PHP objects. This function has been updated to return only PHP variables such as array, int, string, boolean, etc. This change makes it consistent with behavior of other Administration class methods.
 - The deprecated PHP method `Account::remove_redundant_http()` has been removed. Please ensure your customizations are not using it.
 - Due to a [MySQL performance optimizer bug](#), we recommend updating the config for your MySQL server to disable the `'block_nested_loop'` optimization.
 - SQL:

```
SET SESSION optimizer_switch='block_nested_loop=off';
```

- MySQL cnf:

```
[mysqld]  
optimizer_switch=block_nested_loop=off
```

This can cause an increase in the use of temporary files by MySQL. You may need to increase the open files limit on your system if you encounter an error like the following on Linux: "Too many open files in system". For more information, refer to [this easy EasyEngine tutorial](#).

- The Sidecar framework used by the base Sugar web client as well as the portal web client now uses the v11_1 REST API version by default.
- The unminified Sidecar framework source code has been converted into EcmaScript 2015 modules in this Sugar release. Sidecar code is transpiled to EcmaScript 5 and minified when loaded into browsers, so these structural changes do not affect Sugar customizations.
- The following modules have been migrated to the Sidecar user interface. Any UI customizations to these modules will need to be rewritten in the Sidecar framework prior to upgrading to the new release.
 - Contracts
 - Manufacturers
 - Tax Rates
 - Shippers
 - Currencies
 - Contract Types
- In this Sugar release, we've made some changes to Sidecar framework which will prevent customizations from calling private Sidecar methods unless the JS config sidecarCompatMode is true. The default value will be false. The compatibility mode should be used as a temporary work around. All JavaScript customizations are expected to use public Sidecar APIs. If necessary, here is an example of how you can enable sidecarCompatMode using config_override.php.

```
<?php  
$sugar_config['additional_js_config'] = array('sidecarCompatMode'  
' => true);
```

- Many files have moved as part of this refactoring effort as well:

Old Location	New Location
sidecar/lib/backbone/**	sidecar/node_modules/backbone/
sidecar/lib/jasmine-sinon/jasmine-sinon.js	sidecar/node_modules/jasmine-sinon/lib/jasmine-sinon.js

sidecar/lib/jquery/jquery.min.js	sidecar/node_modules/jquery/dist/jquery.min.js
sidecar/lib/jquery/jquery-migrate-1.2.1.min.js	sidecar/node_modules/jquery-migrate/dist/jquery-migrate.min.js
sidecar/lib/jquery/jquery.iframe-transport.js	sidecar/node_modules/jquery.iframe-transport/jquery.iframe-transport.js
sidecar/lib/jquery-ui/js/**	include/javascript/jquery
sidecar/lib/sugarapi/sugarapi.js	sidecar/node_modules/@sugarcrm/ventana/src/client.js

- The following Sidecar methods have been deprecated and will be removed in a future Sugar release. Please update your customizations to use alternative APIs.

```
App.context.getContext()
App.acl.clearCache()
App.acl.action2permission
App.Acl._accessToAny
App.error.initialize()
App.error.remoteLogging
App.error.throwErrorWithCallStack()
App.routing.stop()
App.validation.validators.url()
```

- The following deprecated Sidecar methods were removed. Please verify that your customizations do not use this method prior to upgrading to Sugar 8.0.

```
beanCollection#resetOptions()
```

- The following Sidecar files were removed. Please verify that your customizations do not reference these files prior to upgrading to Sugar 8.0.

```
./include/api/help/module_filter_record_get_help.html
./include/javascript/jquery/jquery.effects.custombounce.js
./sidecar/gulp/assets/files.json
./sidecar/gulp/assets/base-files.json
./sidecar/gulp/assets/default-tests.json
./sidecar/gulp/assets/jquery.json
./sidecar/gulp/assets/zepto.json
./sidecar/gulp/util.js
```

```
./sidecar/lib/backbone
./sidecar/lib/backbone/backbone.js
./sidecar/lib/backbone/backbone.min.js
./sidecar/lib/jasmine
./sidecar/lib/jasmine/jasmine_favicon.png
./sidecar/lib/jasmine/jasmine-html.js
./sidecar/lib/jasmine/jasmine.js
./sidecar/lib/jasmine/MIT.LICENSE
./sidecar/lib/jasmine/jasmine.css
./sidecar/lib/jasmine-zepto
./sidecar/lib/jasmine-zepto/jasmine-zepto.js
./sidecar/lib/jasmine-ci
./sidecar/lib/jasmine-ci/utils
./sidecar/lib/jasmine-ci/utils/core.js
./sidecar/lib/jasmine-ci/utils/console-runner.js
./sidecar/lib/jasmine-ci/jasmine-reporters
./sidecar/lib/jasmine-ci/jasmine-reporters/jasmine.phantomjs-
reporter.js
./sidecar/lib/jasmine-jquery
./sidecar/lib/jasmine-jquery/jasmine-jquery.js
./sidecar/lib/jasmine-sinon
./sidecar/lib/jasmine-sinon/jasmine-sinon.js
./sidecar/lib/jquery-placeholder
./sidecar/lib/jquery-placeholder/jquery.placeholder.js
./sidecar/lib/jquery-placeholder/README.md
./sidecar/lib/jquery-placeholder/LICENSE-MIT.txt
./sidecar/lib/jquery-placeholder/tests
./sidecar/lib/jquery-placeholder/tests/tests.js
./sidecar/lib/jquery-placeholder/tests/index.html
./sidecar/lib/jquery-placeholder/demo.html
./sidecar/lib/jquery-placeholder/bower.json
./sidecar/lib/jquery-ui
./sidecar/lib/jquery-ui/css
./sidecar/lib/jquery-ui/css/smoothness
./sidecar/lib/jquery-ui/css/smoothness/jquery-
ui-1.11.4.custom.min.css
./sidecar/lib/jquery-ui/css/smoothness/images
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
bg_glass_65_ffffff_1x400.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
icons_454545_256x240.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
bg_glass_55_fbf9ee_1x400.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
icons_888888_256x240.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
```

```
bg_flat_75_ffffff_40x100.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
bg_glass_75_dadada_1x400.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
bg_glass_75_e6e6e6_1x400.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
bg_glass_95_feflec_1x400.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
bg_flat_0_aaaaaa_40x100.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
icons_2e83ff_256x240.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
icons_222222_256x240.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-bg_highlight-
soft_75_ccccc_1x100.png
./sidecar/lib/jquery-ui/css/smoothness/images/ui-
icons_cd0a0a_256x240.png
./sidecar/lib/jquery-ui/js
./sidecar/lib/jquery-ui/js/jquery-ui-1.11.4.custom.min.js
./sidecar/lib/jquery-ui/js/jquery-1.7.1.min.js
./sidecar/lib/jquery/wicked.js
./sidecar/lib/jquery/jquery-migrate-1.2.1.min.js
./sidecar/lib/jquery/jquery.iframe.transport.js
./sidecar/lib/jquery/jquery.min.js
./sidecar/lib/sugarapi/sugarapi.js
./sidecar/lib/sugarapi/demoServerData.js
./sidecar/lib/sugarapi/demoRestServer.js
./sidecar/lib/zepto
./sidecar/lib/zepto/ajax.js
./sidecar/lib/zepto/detect.js
./sidecar/lib/zepto/event.js
./sidecar/lib/zepto/fx.js
./sidecar/lib/zepto/fx_methods.js
./sidecar/lib/zepto/readme.txt
./sidecar/lib/zepto/selector.js
./sidecar/lib/zepto/touch.js
./sidecar/lib/zepto/zepto.js
./sidecar/src/core/cookies.js
```

- Deprecated Sidecar method signatures for the following functions have been removed:

```
Context#resetLoadFlag()
MetadataManager#getField()
Router#route()
```

```
ViewManager#createField()
```

- The following Sidecar functions have been deprecated and will be removed in a future Sugar release.

```
Utils.Date.parse()  
Utils.Date.guessFormat()  
Utils.Date.parseFormat()  
Utils.Date.format()  
Utils.Date.roundTime()  
Utils.Date.UTCtoLocalTime()  
Utils.Date.UTCtoTimezone()  
Utils.Date.toUTC()  
Utils.Date.getRelativeTimeLabel()  
Utils.Date.parseDisplayDefault()  
Utils.Date.stripIsoTimeDelimiterAndTZ()  
Utils.Date.isIso()
```

- The deprecated Sidecar method `AlertView#close()` has been removed. Use `AlertView#dispose()` instead.
- The deprecated Sidecar property `Layout#meta` has been removed. Use `Layout#xmeta` instead.
- The following controller methods have been added or updated:
 - Added `app.utils.getReadableFileSize()` for getting a file size in a human-readable format. The function produces descriptions that sacrifice precision for readability. For example, a file size like 1014K is more accurate, but a user would more likely expect to see 1M.
 - Added `app.utils.openEmailCreateDrawer()` for opening a layout for creating emails in a drawer. It is designed for opening the "create" layout for creating an archived email and the "compose-email" layout for creating a draft or sending an email.
 - The Sidecar `Field#setDisabled()` API now supports a boolean trigger option that will trigger a `field:disabled` event on the current Context when called.
 - Sidecar `BeanCollections` and `MixedBeanCollections` now take full advantage of Sugar Collection REST APIs. They now track changes of multiple adds or removals of related records, support for pagination, etc. Work with related collections of records within Sidecar can now be done more easily and efficiently.
 - `AlertView#cancelClicked()` now accepts the Event as a parameter and passes it to the `onCancel` callback.
 - `AlertView#confirmClicked()` now accepts the Event as a parameter and passes it to the `onConfirm` callback.
 - `AlertView#closeClicked()` now passes the Event parameter to the

onClose callback.

- EnumField#loadEnumOptions() now supports passing an error callback as an additional parameter.
- View/ViewManager.createField() takes a new parameter, nested. Set nested to true when the field is part of another parent field that controls the nested field's lifecycle. For example, the individual fields associated with a fieldset are nested. When nested is set to true, the field will not be added to the parent View's list of fields. nested is set to false by default.
- In order to address an issue with ambiguous queries causing 500 errors, we have changed the signature for the following method. You should specify the unambiguous database column name as the second parameter instead of the field name. Please update custom code that uses this PHP method:

```
SugarFieldBase#fixForFilter(&$value, $columnName, SugarBean $bean, SugarQuery $q, SugarQuery_Builder_Where $where, $op)
```

- The following methods have been deprecated and will be removed in a future Sugar release:
 - The Sidecar DataManager API
DataManager#getRelationshipFields() has been deprecated and will be removed in a future Sugar release. Use DataManager#getRelationFields instead.
 - The RecordView method RecordView#initButtons() has been deprecated. Ensure that you do not have custom code that calls this method. It is not necessary to call this method directly.
- The following deprecated method has been removed from Sugar:
 - The deprecated resetOptions method in sugar/sidecar/src/data/bean-collection.js has been removed.
- It is recommended to set `$sugar_config['SAML']['strict'] = true;` for production use of SAML in this Sugar release. This will ensure that the underlying PHP SAML implementation will reject unsigned and unencrypted messages.
- The following Sugar Config settings have been added:
 - `$sugar_config['new_email_addresses_opted_out']`
 - If set to true, newly created Email Address records in Sugar are opted-out by default. This setting can also be controlled using the "Opt-out new email addresses by default" setting in the System Email Settings administration panel.
 - `$sugar_config['logger']['channels']['input_validation']['level'] = 'warning';`
 - Admins can now control the log level for InputValidation regardless of the global log level by using this `config_override` directive.

-
- `$sugar_config['perfProfile']['TeamSecurity']['default']['use_denorm'] = true;`
 - Enable or disable the Team Security denormalization feature. This feature is disabled by default.
 - Team Security denormalization can significantly improve SQL query performance when there are a large number of teams in a Sugar instance. The cost is that this feature relies on a separate team security denormalization table that needs to be updated as records change to ensure Sugar's team security visibility rules are applied properly.
 - `$sugar_config['perfProfile']['TeamSecurity']['inline_update'] = true;`
 - Allows the denormalization table to be updated in real time as records are updated in the system.
 - When denormalization is in use and inline updates are disabled, the denormalization table will need to be scheduled by Sugar Administrator. A new "Rebuild Denormalized Team Security Data" job has been added to Admin > Scheduler.
 - `$sugar_config['logger']['channels']['authentication']`
 - Controls the logging level for authentication messages.
 - Admins and developers can set the log level for these messages as well as set a custom processor for log messages.
 - For example, "backtrace" and "request" will provide useful debug information:
`$sugar_config['logger']['channels']['authentication'] = array('level' => 'debug', 'processors' => ['backtrace']);`
 - `smtp_mailer_debug`
 - Makes it easier to debug SMTP issues that occur using PHPMailer.
 - 7.10.0.0 introduced a brand new, open source charting library called Sucrose. Please visit <http://sucrose.io> for more information. This library was created and is maintained by the SugarCRM development team. All of the out-of-the-box dashlets and reports use the new Sucrose library. Customizations to dashlets and reports should be tested and, where necessary, migrated to the new Sucrose APIs.
 - This Sugar release depends on the [Gnu Multiple Precision](#) (GMP) PHP extension. Make sure all your Sugar hosting environments include the GMP PHP extension prior to using Sugar 8.
 - We have added [ultra-lite/container](#) 6.1.0 as a dependency. This library is a [PSR-11](#) compliant dependency injection container.
 - This Sugar release introduces a new CanvasDataRenderer Sidecar plug-in based on [Phaser CE v2.9.1](#). This plug-in is used by the new Product Catalog dashlet also introduced in this release.
 - The DragdropSelect2 Sidecar plug-in will now allow developers to provide a custom dropDraggedItems(sourceCollection, targetCollection,

draggedItems, droppedItems) function.

- The droppedItems are clones of the draggedItems which makes them safe to manipulate.
- The default drop implementation, if no `dropDraggedItems()` exists, removes the dragged items from the sourceCollection and adds them to the targetCollection.
- Security Subjects mark the first use of Sugar Resource Names (SRNs). SRNs are intended to identify a resource uniquely across all SugarCRM-deployed cloud services. Sugar Resource Name is inspired by the [Uniform Resource Name](#) (URN) concept and is similar to but not the same as [Amazon Resource Names](#) (ARNs).

```
{
  "_type": "user",
  "_srn": "srn:local:crm:::user:1fbbcf60-4aac-4503-b2eb-
d00c3eb391db",
  "name": "John Doe"
}
```

- For Sugar 8, SRNs usage is limited to local resources within the Sugar instance such as the currently logged in user. Local resources use the following scheme: `srn:local:crm:::resource-type:resource-id`
- Do not invent SRNs if you develop your own custom Security Subjects. Resource naming conventions and associated SRNs are the responsibility of SugarCRM.
- Charts within Sugar now fully support system and user locale settings. As part of this change, the Sucrose library version was upgraded from v0.7.5 to v0.8.0. To learn more about working with locales in Sucrose please check out the [locale guide](#).
- The Backbone.js library has been upgraded from version 1.2.3 to 1.3.3. View the [Backbone.js change log](#) for more details on how your Sidecar-based customizations may be affected.
- We have upgraded the Doctrine Database Abstraction Layer (DBAL) library from version 2.5.4 to 2.5.12. For more information, refer to the [Doctrine DBAL website](#).
- As part of our plan to migrate to the Sucrose chart library, we are deprecating JIT and NVD3/D3 libraries and plan to remove them in a future Sugar release. Existing customizations that use JIT or NVD3/D3 should migrate to Sucrose at this time.
- The `iframe.transport` jQuery plug-in has been deprecated and will be removed in a future Sugar release. File uploads should be accomplished by sending [FormData](#) objects as part of AJAX requests.
- An unused `jquery-quicksearch` plug-in was removed in this Sugar release. Ensure that customizations do not use the `quicksearch` jQuery plug-in and that they do not reference `./sidecar/lib/jquery/wicked.js`.

-
- The [jquery-placeholder](#) plug-in has been removed. It is no longer needed in modern browsers that fully support HTML5. This change will not affect customizations unless direct references exist to `./sidecar/lib/jquery-placeholder`.
 - This Sugar release introduces significant changes to Sugar's CSS. This can affect UI customizations that inherit or override the default styling used by Sugar. Developers should test their customizations in the new UI prior to upgrade. See the [developer blog](#) for more details.
 - Sugar now includes and uses the [Open Sans](#) font across the user interface.
 - Sugar's legacy (BWC) modules did not receive any CSS or font changes as part of this Sugar release. For example, BWC modules continue to use Helvetica Neue as the default font.
 - Many duplicate, self-overriding, and unused CSS selectors were removed in this Sugar release.
 - We have updated the default web server access rules in this Sugar release. JavaScript and CSS files installed under `./custom/vendor/` directories are now web-accessible via `https://<instance>/vendor/`.
 - As of 7.11, we have removed references to the generic Twitter Bootstrap v2.2.2 CSS library. This library was heavily overridden by Sugar core code and altered over the years. It has been refactored to better consolidate Sugar CSS. Customizations that directly referenced the `styleguide/less/twitter-bootstrap` library will be affected by this change.
 - The following LESS files have been removed in this Sugar release. Customizations that reference these files directly will be affected by this change.

```
./styleguide/less/clients/portal/bootstrap.less
./styleguide/less/clients/base/bootstrap.less
./styleguide/less/twitter-bootstrap => ./styleguide/less/lib
```

- Erased fields are tracked in a new database table called `erased_fields`.
- Several fields on Person module types (Contacts, Leads, etc.) have been marked as containing personal information by default. The Person template metadata has been updated accordingly. Each of these fields is now audited by default for additional change tracking and marked as personal information to make them eligible for erasure via a data privacy request.
 - `salutation`
 - `first_name`
 - `last_name`
 - `title`
 - `facebook`
 - `twitter`
 - `googleplus`
 - `phone_home`
 - `phone_mobile`
 - `phone_work`

-
- phone_other
 - phone_fax
 - primary_address_street
 - primary_address_street_2
 - primary_address_street_3
 - primary_address_city
 - primary_address_state
 - primary_address_postalcode
 - primary_address_country
 - alt_address_street
 - alt_address_street_2
 - alt_address_street_3
 - alt_address_city
 - alt_address_state
 - alt_address_postalcode
 - alt_address_country
 - assistant
 - assistant_phone

It is important to note that a field **MUST** be audited in order for it to be marked as personal information.

- The following example would be invalid:

```
'pii_stuff' => array(
    'name' => 'pii_stuff',
    'vname' => 'LBL_PII_STUFF',
    'massupdate' => false,
    'type' => 'varchar',
    'len' => 100,
    'comment' => 'PII Stuff',
    'audited' => false, // default value is false
    'pii' => true,
),
```

- This example is valid:

```
'pii_stuff' => array(
    'name' => 'pii_stuff',
    'vname' => 'LBL_PII_STUFF',
    'massupdate' => false,
    'type' => 'varchar',
    'len' => 100,
    'comment' => 'PII Stuff',
    'audited' => true,
    'pii' => true,
),
```

-
- As part of the new data privacy features, a new method has been added to SugarBean class. This method ensures that data for these fields are erased from the audit log as well. This helps ensure that the data is unrecoverable.

```
SugarBean::erase(FieldList $fields, $check_notify)
```

- The audit log data model has also been updated to better support tracking of events. A new Audit\EventRepository class has been introduced for finer grain tracking of audited changes. In particular, the EventRepository is used to track when field changes are associated with an update or erase event as well as the source (Security\Subject) of the change. A new audit_events table in local database is associated with the new EventRepository implementation.
- The audit endpoint for the REST API has been updated to include the source of the change as well as the event_type (either update or erase) information.

For example, in the following two audit log entries, the first shows an erasure done by a Sugar user in response to a data privacy request while the second shows an update originating from Sugar installer creating demo data in the system.

```
GET /rest/v11_1/<module>/:record/audit
{
  "next_offset": -1,
  "records": [{
    "id": "b3361fa0-31ea-11e8-b38a-b243f839f789",
    "parent_id": "602e5946-3139-11e8-aa9f-df5dba0db27d",
    "event_id": "b34525c2-31ea-11e8-b7a8-8564fb05c062",
    "date_created": "2018-03-27T11:14:33-07:00",
    "created_by": "1",
    "date_updated": "2018-03-27 18:14:33",
    "field_name": "phone_home",
    "data_type": "varchar",
    "source": {
      "subject": {
        "_type": "user",
        "id": "1",
        "_module": "Users",
        "client": {
          "_type": "rest-api"
        }
      },
      "first_name": null,
      "last_name": "Administrator",
      "name": "Administrator"
    }
  ]
}
```

```

    "attributes": {
      "platform": "base",
      "dp_request_id": "8daab426-31ea-11e8-8011-34930ab25f6c"
    }
  },
  "event_type": "erasure",
  "created_by_username": "admin",
  "before": null,
  "after": null
},
{
  "id": "603c6130-3139-11e8-b0e3-e077fbb971b3",
  "parent_id": "602e5946-3139-11e8-aa9f-df5dba0db27d",
  "event_id": "602ed3a8-3139-11e8-8d43-6a8a4f4452e9",
  "date_created": "2018-03-26T14:02:43-07:00",
  "created_by": "1",
  "date_updated": null,
  "field_name": "email",
  "data_type": "email",
  "source": {
    "subject": {
      "_type": "installer"
    }
  },
  "event_type": "update",
  "created_by_username": "admin",
  "before": null,
  "after": {
    "id": "60226c80-3139-11e8-bc41-845f0fd96813",
    "email_address": "sales.vegan@example.tv"
  }
}]
}

```

- The birthdate field on Contacts, Leads, Targets modules is marked as personal information by default.
- SugarACLDDataPrivacy is a new SugarACLStrategy class that is used with Data Privacy module. It permits only users with admin-level role access to the Data Privacy module to edit certain fields (e.g. fields_to_erase).
- Sugar REST API endpoints that return record information now support an additional request parameter called `erased_fields`. The filter and module APIs are examples of REST APIs that now support this parameter.
 - When `erased_fields=1` then record responses will include an `_erased_fields` array.

```
"_erased_fields": []
```

- If fields have been erased on this record by a Data Privacy administrator, then they will be listed in this array.

```
"_erased_fields": [  
    "last_name"  
]
```

This will allow an integration or client to know unambiguously when a field has been erased at a data subject's request. Integrations must take steps to erase any copies of this record data. For example, here is an example API call and response:

```
GET /rest/v11_1/Contacts/602e5946-3139-11e8-aa9f-df5dba0db27d?erased_fields=1&fields=first_name,last_name,phone_home,phone_mobile
```

```
{  
    "id": "602e5946-3139-11e8-aa9f-df5dba0db27d",  
    "date_modified": "2018-03-27T11:14:33-07:00",  
    "first_name": "",  
    "last_name": "Dillion",  
    "phone_home": "",  
    "phone_mobile": "(289) 902-4666",  
    "locked_fields": [],  
    "_acl": {  
        "fields": {}  
    },  
    "_erased_fields": [  
        "first_name",  
        "phone_home"  
    ],  
    "_module": "Contacts"  
}
```

- A new erasure feature has been added to Sidecar Fields. A new "erased" field template has been added that is used when a field has been permanently erased as part of a data privacy request. This field template will display a "Value Erased" pill: `./clients/base/fields/base/erased.hbs`.
- As part of the new personal information functionality for data privacy, several new Sidecar components have been added.
 - Sidecar Layout:
 - PiiLayout
 - Sidecar View:

- PiiHeaderpaneView
 - PiiView
- Sidecar Field:
 - PiinameField
- Sidecar plugin:
 - Pii
- The mark-for-erasure flow is implemented as a new action field on the `DataPrivacy.RecordView` that launches a new `MarkForErasureLayout` layout. The following Sidecar components were added in addition to the standard record and list layouts for the Data Privacy module. These new layouts support the mark-for-erasure use case.

- Sidecar Fields

Class name	Directory location
<code>DataPrivacyEraseField</code>	<code>./clients/base/fields/dataprivacyerase/</code>

- Sidecar Layouts

Class name	Directory location
<code>DataPrivacy.MarkForErasureLayout</code>	<code>./modules/DataPrivacy/clients/base/layouts/mark-for-erasure</code>

- Sidecar Views

Class name	Directory location
<code>DataPrivacy.MarkForErasureHeaderpaneView</code>	<code>./modules/DataPrivacy/clients/base/views/mark-for-erasure-headerpane</code>
<code>DataPrivacy.MarkForErasureView</code>	<code>./modules/DataPrivacy/clients/base/views/mark-for-erasure</code>
<code>DataPrivacy.MarkedForErasureDashlet</code>	<code>./modules/DataPrivacy/clients/base/views/marked-for-erasure-dashlet/</code>

- A new Sugar Config setting has been added. If set to true, newly created Email Address records in Sugar are opted-out by default. This setting can also be controlled using the "Opt-out new email addresses by default" setting in the System Email Settings administration panel.

```
$sugar_config['new_email_addresses_opted_out']
```

- Email address `invalid_email` and `opt_out` fields added to `emails_email_addr_rel` relationship vardefs. This is used with

EmailParticipants beans.

- The modules/Emails/clients/base/fields/email-recipients/select2-selection.hbs has been updated to make sure opted-out email addresses are decorated properly.
- The invalid_email and opt_out fields are now included with from/to/cc/bcc collection field data for Emails module.
- The EmailField Handlebars templates (detail, list) and JavaScript controller have been changed to allow opted-out email addresses to be selected for use in transactional email.

```
./sugarcrm/clients/base/fields/email/detail.hbs
./clients/base/fields/email/list.hbs
./sugarcrm/clients/base/fields/email/email.js
```

- The following Sugar Authentication classes have been deprecated and will be removed in a future Sugar release:

- ./modules/Users/authentication/LDAPAuthenticate/
./modules/Users/authentication/SAMLAuthenticate/
./modules/Users/authentication/SugarAuthenticate/

- Developers should use the following Identity Manager classes instead:

```
./modules/Users/authentication/IdmLDAPAuthenticate/  
./modules/Users/authentication/IdmSAMLAuthenticate/  
./modules/Users/authentication/IdmSugarAuthenticate/
```

- In the long term, we plan to refactor Identity Management into a service that is separate from the core Sugar application.
- With the new ability to disable activity streams, there are behavior changes to some Sugar APIs that developers should know about.
 - When using the existing Sugar PHP functions Activity::enable() or Activity::disable() to temporarily change status of activity streams, please use the Activity::restoreToPreviousState() function to restore the previous state. For example, use this sequence to temporarily disable activity streams in custom code:

```
Activity::disable();  
...  
Activity::restoreToPreviousState()
```

- When Activity Streams are disabled, then calls to the following Sugar REST API endpoints will return HTTP 401 NOT

AUTHORIZED errors:

```
GET /<module>/:record/link/activities
GET /<module>/:record/link/activities/filter

GET /<module>/Activities
GET /<module>/Activities/filter

GET /Activities
GET /Activities/filter
```

- Advanced Workflow REST API documentation has been improved. Refer to the built-in REST documentation or the [Sugar Developer Guide](#) for more details.
- The PMSEEmailHandler class will now lazy-load member objects such as the Mailer object via getters and setters. The Mailer object used with PMSEEmailHandler will also now use system default system Mail object. These changes have been made to improve performance, testability, and future platform enhancement work.
- It is recommended that Sugar Developers use getters and setters to access member objects on PMSE classes.
- The following methods have been removed from PMSEEmailHandler.

```
PMSEEmailHandler#retrieveSugarPHPMailer()
PMSEEmailHandler#setupMailObject()
```

- The following methods have been deprecated and will be removed in a future Sugar release.

```
PMSEEmailHandler#__construct()
PMSEEmailHandler#getLocale()
PMSEEmailHandler#setLocale()
```

- More lazy-loading implementations will follow, so when extending Advanced Workflow classes, it is a best practice to use getter/setter methods where needed instead of using direct object access.
- As part of the new Sharable Dashboards, previous versions' out-of-the-box dashboards will be converted to the new style of dashboard. If you have customized your dashboards to names other than list-dashboard.php or record-dashboard.php, they will NOT be upgraded to the new dashboard type, but the metadata will still exist. If you want to create a shareable dashboard record similar to your old customized dashboard, you can generate it manually through the UI.
- The view_name field has changed from type varchar to enum. The view_name field will be converted during upgrade process. If you happen to

have custom layouts that use Dashboards besides the default "Record," "List," or "Activity Stream," then you will need to update the "dashboard_view_name_list" dropdown list to add those additional layouts.

- The dashboard_module field has been changed from type varchar to enum.
- A new ACL class called SugarACLAdminOnlyFields has been created. The ACL allows for a list of non_writeable_fields that can only be editable by a Sugar admin user. This is used on new Dashboards module to ensure that only admin users are able to designate default dashboards which are pushed to all users within the set team.
- As part of the new shareable dashboards, the old default dashboards that were stored in metadata are converted into dashboard records in the Sugar database. To configure different default dashboards for Sugar users, please use shareable dashboards instead of modifying dashboard metadata. The following default dashboard metadata locations are no longer recognized in this Sugar release.

```
./modules/<module>/clients/base/layouts/list-dashboard  
./modules/<module>/clients/base/layouts/record-dashboard
```

- As part of the new sharable dashboards, the previous out-of-the-box dashboards will be converted to the new style of dashboard. If you have customized your dashboards to names other than "list-dashboard.php" or "record-dashboard.php," they will NOT be upgraded to the new dashboard type, but the metadata will still exist. If you want to create a shareable dashboard record similar to your old customized dashboard, you can generate it manually through the UI. Otherwise, everything should continue to work as it has.
- The portal home page has been renamed. As part of this renaming, the following files have been renamed:

- sugarcrm/clients/portal/layouts/dashboard/dashboard.php is now sugarcrm/clients/portal/layouts/home/home.php
 - sugarcrm/clients/portal/layouts/dashboard/dashboard.js is now sugarcrm/clients/portal/layouts/home/home.js

Any customizations related to dashboard.php, dashboard.js, or home.php will need to be updated.

- dashboard-list has been renamed to portal-list. As part of this renaming, the following files have been renamed:
 - sugarcrm/clients/portal/layouts/dashboard-list/dashboard-list.php is now sugarcrm/clients/portal/layouts/portal-list/portal-list.php
 - sugarcrm/clients/portal/views/dashboard-list-top/dashboard-list-top.hbs is now sugarcrm/clients/portal/views/portal-list-top/portal-list-top.hbs
 - sugarcrm/clients/portal/views/dashboard-list-top/dashboard-list-top.js is now sugarcrm/clients/portal/views/portal-list-top/portal-list-top.js
- Any customizations related to dashboard-list.php, dashboard-list-top.hbs, or dashboard-list-top.js will need to be updated.
- sugarcrm/modules/KBCContents/clients/portal/layouts/list-dashboard/list-

dashboard.php has been replaced with

sugarcrm/modules/KBContents/clients/portal/layouts/rhs-pane/rhs-pane.php. Any customizations related to list-dashboard.php will need to be updated.

- The hidden sidebar on the right hand side of the Home dashboard is being removed. As part of this removal, sugarcrm/modules/Home/clients/base/layouts/record/record.js will be deleted.
- Generating a dashboard from metadata has been deprecated. All dashboards should be created using sharable dashboards.
- As part of the new sharable dashboards, all custom Dashboard module record views will be renamed to end in .bak. You have the option of retrieving your custom Dashboard record view by renaming it so it does not end in .bak. Note that if you reinstate your old custom Dashboard record view, you will override the Sugar dashboard record view. A future release will remove all custom Dashboard record views ending in .bak, so you should review them to see if you still need them.
- The APIs used by legacy Email Compose are now deprecated including the following:
 - Many of the routes in sugarcrm/modules/Emails/clients/base/api/MailApi.php have been deprecated and are registered with maxVersion as v11.
 - The OutboundEmailConfiguration/listendpoint in sugarcrm/modules/OutboundEmailConfiguration/clients/base/api/OutboundEmailConfigurationApi.php has been deprecated. Emails/enum/outbound_email_id can be used instead.
 - All routes in sugarcrm/modules/Emails/MailRecord.php have been deprecated.
- As part of new Emails module work, many PHP methods, properties, and some Sidecar components have been deprecated and will be removed in a future Sugar release.
 - PHP methods

```
Email::email2init()  
Email::email2saveAttachment()  
Email::safeAttachmentName()  
Email::email2ParseAddresses()  
Email::email2ParseAddressesForAddressesOnly()  
Email::email2GetMime()  
Email::sendEmailTest()  
Email::isDraftEmail()  
Email::email2Send()  
Email::getNamePlusEmailAddressesForCompose()  
Email::_arrayToDelimitedString()  
Email::saveTempNoteAttachments()  
Email::saveEmailAddresses()  
Email::linkEmailToAddress()
```

```
Email::cleanEmails()  
Email::loadAdditionalEmailData()  
Email::getForwardHeader()  
Email::getNotes()  
Email::getReplyHeader()  
Email::quotePlainTextEmail()  
Email::quoteHtmlEmail()  
Email::quoteHtmlEmailForNewEmailUI()  
Email::check_email_settings()  
Email::js_set_archived()  
Email::u_get_clear_form_js()  
Email::pickOneButton()  
Email::getUserEditorPreference()  
Email::parse_addrs()  
Email::remove_empty_fields()  
Email::findEmailsLink()  
Email::handleAttachments()  
Email::hasSignatureInBody()  
Email::removeAllNewlines()  
Email::getStartPage()  
Email::send()  
Email::getSystemDefaultEmail()  
Email::create_new_list_query()  
Email::fill_in_additional_list_fields()  
Email::fill_in_additional_detail_fields()  
Email::quickCreateForm()  
Email::searchImportedEmails()  
Email::doesImportedEmailHaveAttachment()  
Email::_generateSearchImportedEmailsQuery()  
Email::_generateSearchImportWhereClause()  
Email::trimLongTo()  
Email::distributionForm()  
Email::userSelectTable()  
Email::checkInbox()  
Email::fillPrimaryParentFields()  
Email::cid2Link()  
Email::cids2Links()  
Email::setFieldNullable()  
Email::revertFieldNullable()  
Email::toDatabaseSearchDateTime()  
Email::_setMailerFactoryClassName()  
Email::retrieveEmailAddresses()
```

- PHP properties

```
Email::$from_addr
```

```
Email::$from_addr_name
Email::$from_name
Email::$to_addrs
Email::$to_addrs_names
Email::$to_addrs_arr
Email::$to_addrs_emails
Email::$to_addrs_ids
Email::$cc_addrs
Email::$cc_addrs_names
Email::$cc_addrs_arr
Email::$cc_addrs_emails
Email::$cc_addrs_ids
Email::$bcc_addrs
Email::$bcc_addrs_names
Email::$bcc_addrs_arr
Email::$bcc_addrs_emails
Email::$bcc_addrs_ids
Email::$attachment_image
Email::$imagePrefix
Email::$type_name
Email::$status_name
Email::$type
Email::$status
Email::$flagged
Email::$cachePath
Email::$cacheFile
Email::$replyDelimiter
Email::$emailDescription
Email::$emailDescriptionHTML
Email::$emailRawSource
Email::$link_action
Email::$relationshipMap
Email::$modifiedFieldDefs
Email::$addedFieldDefs
Email::$emailAddress
Email::$saved_attachments
Email::$set
Email::$reply_to_addr
Email::$reply_to_name
Email::$reply_to_email
```

- Sidecar components

- EmailsAttachmentButtonField is deprecated.
./modules/Emails/clients/base/fields/attachment-button
- EmailsRecipientsField is deprecated.
./modules/Emails/clients/base/fields/recipients

-
- EmailsSenderField is deprecated. Use OutboundEmailField instead.
./modules/Emails/clients/base/fields/sender
 - EmailsComposeDocumentsLayout is deprecated
Emails/clients/base/layouts/compose-documents
 - EmailsComposeActionBarField is deprecated. Use
HtmlEditable_tinymceField instead.
Emails/clients/base/fields/compose-actionbar
 - EmailsArchiveEmailLayout is deprecated. Use CreateLayout instead.
Emails/clients/base/layouts/archive-email
 - EmailsArchiveEmailView is deprecated. Use CreateView instead.
Emails/clients/base/views/archive-email
 - EmailsComposeLayout is deprecated. Use
ComposeEmailLayout instead.
Emails/clients/base/layouts/compose
 - EmailsComposeView is deprecated. Use ComposeEmailView instead.
Emails/clients/base/views/compose
 - EmailsComposeSenderoptionsView is deprecated.
Emails/clients/base/views/compose-senderoptions
 - BaseAttachmentsField is deprecated. Use
EmailAttachmentsField instead.
clients/base/fields/attachments
- As part of our new Email functionality, many of the APIs related to email have changed:
 - Recipients returned by Sugar Email REST APIs will now include standard ACL data (_acl) for each object.
 - The following endpoints are new and available beginning in version 10:
 - POST /EmailAddresses
 - PUT /EmailAddresses/:record
 - GET /Emails/:record
 - GET /Emails/filter
 - GET /Emails
 - GET /Emails/count
 - POST /Emails/filter
 - POST /Emails/filter/count
 - GET /Emails/filter/count
 - POST /Emails/:record/link/:link_name/:remote_id
 - POST /Emails/:record/link
 - DELETE /Emails/:record/link/:link_name/:remote_id
 - POST
/Emails/:record/link/:link_name/add_record_list/:remote_id

-
- GET /Mail/recipients/find
 - POST /OutboundEmail
 - PUT /OutboundEmail/:record
 - The following endpoints are new and available beginning in version 11:
 - POST /Emails
 - PUT /Emails/:record
 - The following endpoints are now deprecated:
 - POST /Mail
 - POST /Mail/archive
 - POST /Mail/recipients/lookup
 - POST /Mail/address/validate
 - POST /Mail/attachment
 - DELETE /Mail/attachment/:file_guid
 - DELETE /Mail/attachment/cache
 - GET /OutboundEmailConfiguration/list
 - New Emails subpanel metadata is used throughout Sugar modules to allow unlinking of Email records except where emails are implicitly related via a matching email address or automated import email process. Custom Emails subpanels will need to be tested and updated.
 - We have created version 11 of the Sugar REST API. The previous API version (10) will continue to behave as before. Refer to the [Developer Guide](#) for more information.
 - The Sugar REST API now supports Major and Minor versioning as discussed in a previous [Sugar Developer blog post](#). This release introduces the v11_1 REST API. While maintaining full compatibility with v11, the v11_1 REST API adds some additional endpoints and features.
 - The deprecated REST endpoint /rest/<version>/<module>/filter/<filter_id> has been removed. Use /rest/<version>/<module>/filter?filter_id=<id> instead.
 - As of Sugar 7.11, we are restricting API access using unregistered custom platforms by default. This means that API integrations that use custom platforms will need to make sure Sugar instances are configured to allow your integration. Please review this [Developer Blog](#) post for more details. Developers can easily create a module-loadable package that customer administrators can install to allow their integration. Please feel free to use this [example package](#) as a basis.
 - It is now possible for an administrator to add or remove custom API platforms using the Configure API Platforms screen in the admin panel. It is no longer necessary for an integration developer to create and deploy a [Platform extension](#) into customer instances to allow use of [unregistered API platforms](#). This new admin functionality does not interfere with existing platform extensions that may be installed on Sugar instance.
 - Dates on SugarBean objects should now be consistently formatted in the DB format instead of user-selected format. Developers should ensure their code relies on DB format when working with date information on

SugarBean objects.

- The REST API version can now be specified via an HTTP Accept header. The REST API version should be specified in either the URL or in the HTTP Accept header but not in both locations at once. Both of the examples below represent valid requests to v10 API:
 - GET <sugar>/rest/v10/me
 - GET <sugar>/rest/me
 - ...
 - Accept: application/vnd.sugarcrm.core+json; version=10
- The following new REST API endpoints have been added. Refer to the [Developer Guide](#) for more information.
 - GET rest/v11/Dashboards
 - POST rest/v11/Dashboards
 - GET /rest/ProductTemplates/tree
- The SugarAutoloader has been optimized to improve application performance. In particular, testing has revealed that the ./cache/file_map.php cache introduces unnecessary overhead in terms of average memory usage and request response time. The Sugar Autoloader no longer maintains a map of each PHP file used by Sugar. As part of this change, the following SugarAutoloader methods are now deprecated and will be removed in an upcoming release.

Deprecated API	Alternative API
SugarAutoLoader::fileExists()	Use file_exists() instead.
SugarAutoLoader::loadFileMap()	Unnecessary. No alternative needed.
SugarAutoLoader::addToMap()	Unnecessary. No alternative needed.
SugarAutoLoader::delFromMap()	Unnecessary. No alternative needed.
SugarAutoLoader::unlink()	Use unlink()* instead.
SugarAutoLoader::touch()	Use sugar_touch()* instead.
SugarAutoLoader::put()	Use file_put_contents()* or sugar_file_put_contents()* instead.
SugarAutoLoader::saveMap()	Unnecessary. No alternative needed.

Please be reminded that these original APIs and these alternatives are all restricted in SugarCloud deployments. Avoid file system I/O in your Sugar code customizations.

- The following REST API endpoints were added in this release.

API Endpoint	Summary
/rest/v11_1/<module>/:record/pii	Retrieves current values for PII fields on the given record including the source of this data.

- New Sugar REST APIs for administering Elasticsearch have been added. More details are available in the REST API documentation.

```
rest/<version>/Administration/elasticsearch/refresh/status
rest/<version>/Administration/elasticsearch/refresh/trigger
rest/<version>/Administration/elasticsearch/refresh/enable
rest/<version>/Administration/elasticsearch/replicas/status
rest/<version>/Administration/elasticsearch/replicas/enable
```

- PHP Type hints were added to many REST API service and Elasticsearch related classes and interfaces to support improvements in support for type declarations in PHP 7. This can cause runtime errors when functions are called using incorrectly typed parameters. These errors can affect custom classes that extend core REST API classes or interfaces.

```
./src/Elasticsearch/Query/QueryBuilder.php
    public function addFilter(\Elastica\Query\AbstractQuery $filter
    )
    public function addPostFilter(\Elastica\Query\AbstractQuery $postFilter)
```

```
./src/Elasticsearch/Provider/Visibility/StrategyInterface.php
    public function elasticAddFilters(User $user, \Elastica\Query\BooleanQuery $filter, Visibility $provider);
```

```
./clients/base/api/BulkApi.php
    public function bulkCall(ServiceBase $api, array $args)
```

```
./clients/base/api/CalendarEventsApi.php
    public function updateCalendarEvent(ServiceBase $api, array $args)
    public function deleteCalendarEvent(ServiceBase $api, array $args)
    public function updateRecurringCalendarEvent(SugarBean $bean
    , ServiceBase $api, array &$args)
    public function deleteRecordAndRecurrences(ServiceBase $api,
    array $args)
    public function deleteRecurrences(SugarBean $bean)
    protected function initializeArgs(array $args, SugarBean $bean = null)
    protected function adjustStartDate(array &$args)
    protected function filterOutRecurrenceFields(array $args)
    protected function shouldAutoInviteParent(SugarBean $bean, array $args)
    protected function getRecurringSequence(array $args)
```

```
./clients/base/api/CollectionApi.php
```

```

        protected function mapSourceArguments(CollectionDefinitionIn
terface $definition, $source, array $args)
        abstract protected function getSourceData(ServiceBase $api,
$source, array $args);
        abstract protected function getSourceCount(ServiceBase $api,
$source, array $args);

./clients/base/api/ConfigModuleApi.php
    public function config(ServiceBase $api, array $args)

./clients/base/api/ConnectorApi.php
    public function getConnectors(ServiceBase $api, array $args)
    public function validateHash(array $args)

./clients/base/api/CurrentUserApi.php
    public function retrieveCurrentUser(ServiceBase $api, array
$args)
    protected function getUserHash(User $user)
    public function updateCurrentUser(ServiceBase $api, array $a
rgs)
    public function updatePassword(ServiceBase $api, array $args
)
    public function verifyPassword(ServiceBase $api, array $args
)
    protected function getUserPref(User $user, $pref, $metaName,
$category = 'global')
    protected function getUserPrefTimezone(User $user, $category
= 'global')
    protected function getUserPrefCurrency(User $user, $category
= 'global')
    protected function getUserPrefSignature_default(User $user)
    protected function getUserPrefSignature_prepend(User $user,
$category = 'global')
    protected function getUserPrefEmail_link_type(User $user)
    protected function getUserPrefLanguage(User $user)
    protected function changePassword(SugarBean $bean, $old, $ne
w)
    public function userPreferences(ServiceBase $api, array $arg
s)
    public function userPreferencesSave(ServiceBase $api, array
$args)
    public function userPreference(ServiceBase $api, array $args
)
    public function userPreferenceSave(ServiceBase $api, array $
args)
    public function userPreferenceDelete(ServiceBase $api, array

```

```

    $args)
        public function getMyFollowedRecords(ServiceBase $api, array
    $args)

./clients/base/api/DuplicateCheckApi.php
    protected function trimArgs(array $args)
    protected function populateFromApi(ServiceBase $api, SugarBe
an $bean, array $args, array $options = array())

./clients/base/api/FileApi.php
    public function saveFilePut(ServiceBase $api, array $args, $
stream = 'php://input')
    protected function checkFileAccess(SugarBean $bean, $field,
array $args)
    public function saveFilePost(ServiceBase $api, array $args,
$temporary = false)
    public function getFileList(ServiceBase $api, array $args)
    public function getFile(ServiceBase $api, array $args)
    public function removeFile(ServiceBase $api, array $args)
    public function getArchive(ServiceBase $api, array $args)
    protected function deleteIfFails(SugarBean $bean, array $arg
s)
    protected function getFileInfo(SugarBean $bean, $field, Serv
iceBase $api)
    protected function isFileEncoded(ServiceBase $api, array $ar
gs)

./clients/base/api/FileTempApi.php
    public function saveTempImagePost(ServiceBase $api, array $a
rgs)
    public function getTempImage(ServiceBase $api, array $args)

./clients/base/api/FilterApi.php
    protected function populateRelatedFields(SugarBean $bean, $d
ata)
    protected function removeRelateCollectionsFromSelect(array $
options)

./clients/base/api/GlobalSearchApi.php
    public function globalSearch(ServiceBase $api, array $args)
    protected function formatResults(ServiceBase $api, ResultSet
Interface $results)
    protected function formatBeanFromResult(ServiceBase $api, Re
sult $result)

./clients/base/api/HelpApi.php

```

```

        public function getHelp(ServiceBase $api, array $args)
        public function getExceptionsHelp(ServiceBase $api, array $a
args)

./clients/base/api/ListApi.php
    public function parseArguments(ServiceBase $api, array $args
, SugarBean $seed = null)
        public function listModule(ServiceBase $api, array $args)
        protected function performQuery(ServiceBase $api, array $arg
s, SugarBean $seed, $queryParts, $limit, $offset)

./clients/base/api/LocaleApi.php
    public function localeOptions(ServiceBase $api, array $args)

./clients/base/api/LoggerApi.php
    public function logMessage(ServiceBase $api, array $args)

./clients/base/api/MassUpdateApi.php
    public function massDelete(ServiceBase $api, array $args)
    public function massUpdate(ServiceBase $api, array $args)

./clients/base/api/MetadataApi.php
    protected function getTypeFilter(array $args, $default)
    protected function getModuleFilter(array $args, $default)
    protected function isOnlyHash(array $args)
    public function getPublicMetadata(ServiceBase $api, array $a
args)
    protected function filterResults(array $args, $data, $typeFil
ter, $onlyHash = false, $baseChunks = array(), $perModuleChunks
= array(), $moduleFilter = array())

./clients/base/api/ModuleApi.php
    public function getEnumValues(ServiceBase $api, array $args)
    public function updateRecord(ServiceBase $api, array $args)
    public function updateRelatedRecords(ServiceBase $api, Sugar
Bean $bean, array $args)
    public function retrieveRecord(ServiceBase $api, array $args
)
        public function deleteRecord(ServiceBase $api, array $args)
        public function setFavorite(ServiceBase $api, array $args)
        public function unsetFavorite(ServiceBase $api, array $args)
        protected function getRelatedFields(array $args, SugarBean $
bean)
            protected function moveTemporaryFiles(array $args, SugarBean
$bean)
            protected function getLoadedAndFormattedBean(ServiceBase $ap

```

```
i, array $args)
    protected function processAfterCreateOperations(array $args,
        SugarBean $bean)

./clients/base/api/ModuleCollectionApi.php
    protected function getSourceData(ServiceBase $api, $source,
array $args)
    protected function getSourceCount(ServiceBase $api, $source,
array $args)

./clients/base/api/OAuth2Api.php
    protected function getOAuth2Server(array $args)
    public function token(ServiceBase $api, array $args)
    public function logout(ServiceBase $api, array $args)
    public function bwcLogin(ServiceBase $api, array $args)

./clients/base/api/PasswordApi.php
    public function requestPassword(ServiceBase $api, array $arg
s)

./clients/base/api/PingApi.php
    public function ping(ServiceBase $api, array $args)

./clients/base/api/PipelineChartApi.php
    public function pipeline(ServiceBase $api, array $args)
    protected function buildQuery(ServiceBase $api, SugarBean $s
eed, $tp, $amount_field, $type = 'user')

./clients/base/api/RSSFeedApi.php
    public function getFeed(ServiceBase $api, array $args)
    public function getFeedLimit(array $args)

./clients/base/api/RecentApi.php
    protected function parseArguments(array $args)
    public function getRecentlyViewed(ServiceBase $api, array $a
rgs, $acl = 'list')
    protected function getRecentlyViewedQueryObject(SugarBean $s
eed, array $options)

./clients/base/api/RecordListApi.php
    public function recordListCreate(ServiceBase $api, array $ar
gs)
    public function recordListDelete(ServiceBase $api, array $ar
gs)
    public function recordListGet(ServiceBase $api, array $args)
```

```
./clients/base/api/RegisterLeadApi.php
    protected function updateBean(SugarBean $bean, ServiceBase $
api, array $args)
    public function createLeadRecord(ServiceBase $api, array $ar
gs)

./clients/base/api/RelateCollectionApi.php
    protected function getSourceData(ServiceBase $api, $source,
array $args)
    protected function getSourceCount(ServiceBase $api, $source,
array $args)

./clients/base/api/RelateRecordApi.php
    protected function checkRelatedSecurity(ServiceBase $api, ar
ray $args, SugarBean $primaryBean, $securityTypeLocal = 'view',
    $securityTypeRemote = 'view')
    protected function getRelatedFields(ServiceBase $api, array
$args, SugarBean $primaryBean, $linkName, SugarBean $seed = null
)
    public function getRelatedRecord(ServiceBase $api, array $ar
gs)
    public function createRelatedRecord(ServiceBase $api, array
$args)
    public function createRelatedLink(ServiceBase $api, array $a
rgs)
    public function createRelatedLinks(ServiceBase $api, array $
args, $securityTypeLocal = 'view', $securityTypeRemote = 'view')
    public function updateRelatedLink(ServiceBase $api, array $a
rgs)
    public function deleteRelatedLink(ServiceBase $api, array $a
rgs)
    public function createRelatedLinksFromRecordList(ServiceBase
$api, array $args)

./clients/base/api/TwitterApi.php
    public function getTweets(ServiceBase $api, array $args)
    public function getCurrentUser(ServiceBase $api, array $args
)

./clients/base/api/vCardApi.php
    public function vCardSave(ServiceBase $api, array $args)
    public function vCardDownload(ServiceBase $api, array $args)
    public function vCardImport(ServiceBase $api, array $args)
    protected function getVcardForRecord(ServiceBase $api, array
$args)
```

```

./clients/mobile/api/CurrentUserMobileApi.php
    protected function getUserHash(User $user)

./clients/portal/api/CurrentUserPortalApi.php
    public function retrieveCurrentUser(ServiceBase $api, array
$args)
    public function updateCurrentUser(ServiceBase $api, array $a
rgs)
    protected function changePassword(SugarBean $bean, $old, $ne
w)

./data/SugarBean.php
    public function populateFromRow(array $row, $convert = false
)

./include/Expressions/Dependency.php
    public function fire(SugarBean $target)
    private function fireActions(SugarBean $target, $useFalse =
false)

./include/Expressions/Expression/Numeric/NumericExpression.php
    protected function isCurrencyField(SugarBean $bean, $field)
    protected function getFieldPrecision(SugarBean $bean, $field
)

./include/SugarObjects/templates/file/File.php
    public function populateFromRow(array $row, $convert = false
)

./include/SugarObjects/templates/person/Person.php
    protected function getVCalData(array $options)

./include/api/SugarApi.php
    public function requireArgs(array $args, $requiredFields = a
rray())
    protected function formatBean(ServiceBase $api, array $args,
SugarBean $bean, array $options = array())
    protected function formatBeans(ServiceBase $api, array $args
, $beans, array $options = array())
    protected function loadBean(ServiceBase $api, array $args, $
aclToCheck = 'view', array $options = array())
    protected function updateBean(SugarBean $bean, ServiceBase $
api, array $args)
    protected function toggleFavorites(SugarBean $bean, $favorit
e)

```

```
./include/api/SugarListApi.php
    public function parseArguments(ServiceBase $api, array $args
, SugarBean $seed = null)

./modules/Accounts/clients/base/api/AccountsApi.php
    public function opportunityStats(ServiceBase $api, array $ar
gs)

./modules/ActivityStream/clients/base/api/ActivitiesApi.php
    protected function checkParentPreviewEnabled(User $user, $mo
dule, $id)

./modules/Administration/clients/base/api/AdministrationApi.php
    public function searchReindex(ServiceBase $api, array $args)
    public function searchStatus(ServiceBase $api, array $args)
    public function searchFields(ServiceBase $api, array $args)
    public function elasticSearchQueue(ServiceBase $api, array $
args)
    public function elasticSearchRouting(ServiceBase $api, array
$args)
    public function elasticSearchIndices(ServiceBase $api, array
$args)
    public function elasticSearchMapping(ServiceBase $api, array
$args)

./modules/Audit/clients/base/api/AuditApi.php
    public function viewChangeLog(ServiceBase $api, array $args)

./modules/Calendar/clients/base/api/CalendarApi.php
    public function inviteeSearch(ServiceBase $api, array $args)
    protected function buildSearchParams(array $args)
    protected function transformInvitees(ServiceBase $api, array
$args, $searchResults)
    protected function getMatchedFields(array $args, $record, $m
axFields = 0)

./modules/Categories/clients/base/api/TreeApi.php
    protected function createNewBean(ServiceBase $api, array $ar
gs)
    public function prepend(ServiceBase $api, array $args)
    public function append(ServiceBase $api, array $args)
    public function insertBefore(ServiceBase $api, array $args)
    public function insertAfter(ServiceBase $api, array $args)
    public function moveBefore(ServiceBase $api, array $args)
    public function moveAfter(ServiceBase $api, array $args)
    public function moveFirst(ServiceBase $api, array $args)
```

```
public function moveLast(ServiceBase $api, array $args)
public function tree(ServiceBase $api, array $args)
public function children(ServiceBase $api, array $args)
public function next(ServiceBase $api, array $args)
public function prev(ServiceBase $api, array $args)
public function getParent(ServiceBase $api, array $args)
public function path(ServiceBase $api, array $args)
public function filterSubTree(ServiceBase $api, array $args)
public function filterTree(ServiceBase $api, array $args)

./modules/Contacts/clients/base/api/ContactsApi.php
    public function influencers(ServiceBase $api, array $args)
    public function opportunityStats(ServiceBase $api, array $args)
    public function getFreeBusySchedule(ServiceBase $api, array $args)
    protected function getBean(ServiceBase $api, array $args)
    protected function getAccountBean(ServiceBase $api, array $args)
    protected function getAccountRelationship(ServiceBase $api, array $args, $account, $relationship, $limit = 5, $query = array())

./modules/Dashboards/clients/base/api/DashboardApi.php
    public function createDashboard(ServiceBase $api, array $args)

./modules/Dashboards/clients/base/api/DashboardListApi.php
    public function getDashboards(ServiceBase $api, array $args)

./modules/Documents/Document.php
    public function populateFromRow(array $row, $convert = false)

./modules/Documents/clients/base/api/DocumentsFileApi.php
    protected function checkFileAccess(SugarBean $bean, $field, array $args)
    protected function deleteIfFails(SugarBean $bean, array $args)

./modules/DynamicFields/templates/Fields/TemplateField.php
    public function populateFromRow(array $row)

./modules/DynamicFields/templates/Fields/TemplateRelatedTextField.php
    public function populateFromRow(array $row)
```

```

./modules/Emails/Email.php
    public function populateFromRow(array $row, $convert = false
)

./modules/Emails/clients/base/api/MailApi.php
    public function createMail(ServiceBase $api, array $args)
    public function updateMail(ServiceBase $api, array $args)
    public function archiveMail(ServiceBase $api, array $args)
    protected function handleMail(ServiceBase $api, array $args)
    public function recipientLookup(ServiceBase $api, array $arg
s)
    public function findRecipients(ServiceBase $api, array $args
)
    public function validateArguments(array &$args)
    protected function validateRecipients(array $args, $argName,
$isRequired = false)
    protected function initMailRecord(array $args)
    public function validateEmailAddresses(ServiceBase $api, arr
ay $args)
    public function saveAttachment(ServiceBase $api, array $args
)
    public function removeAttachment(ServiceBase $api, array $ar
gs)
    public function clearUserCache(ServiceBase $api, array $args
)

./modules/EmbeddedFiles/clients/base/api/EmbeddedFileApi.php
    public function saveFilePost(ServiceBase $api, array $args,
$temporary = false)
    public function getFile(ServiceBase $api, array $args)

./modules/ExpressionEngine/clients/base/api/RelatedValueApi.php
    public function getRelatedValues(ServiceBase $api, array $ar
gs)
    protected function isFieldCurrency(SugarBean $bean, $field)

./modules/ExpressionEngine/clients/base/api/SugarLogicFunctionsA
pi.php
    public function getSugarLogicFunctions(ServiceBase $api, arr
ay $args)

./modules/Filters/clients/base/api/PreviouslyUsedFiltersApi.php
    public function setUsed(ServiceBase $api, array $args)
    public function getUsed(ServiceBase $api, array $args)
    public function deleteUsed(ServiceBase $api, array $args)

```

```
./modules/ForecastManagerWorksheets/clients/base/api/ForecastManagerWorksheetsApi.php
    public function assignQuota(ServiceBase $api, array $args = array())

./modules/ForecastWorksheets/clients/base/api/ForecastWorksheetsApi.php
    public function forecastWorksheetSave(ServiceBase $api, array $args)
    protected function getClass(array $args)

./modules/Forecasts/clients/base/api/ForecastsApi.php
    public function returnEmptySet(ServiceBase $api, array $args)
    public function forecastsInitialization(ServiceBase $api, array $args)
    public function retrieveSelectedUser(ServiceBase $api, array $args)
    public function timeperiod(ServiceBase $api, array $args)
    protected function getTimeperiodFilterClass(array $args)
    public function getReportees(ServiceBase $api, array $args)
    public function getOrgTree(ServiceBase $api, array $args)
    protected function compareSettingsToDefaults(Administration $admin, $forecastsSettings, ServiceBase $api)
    public function getQuota(ServiceBase $api, array $args)

./modules/Forecasts/clients/base/api/ForecastsChartApi.php
    public function chart(ServiceBase $api, array $args)
    protected function getClass($file, $klass, array $args)

./modules/Forecasts/clients/base/api/ForecastsFilterApi.php
    public function forecastsCommitted(ServiceBase $api, array $args)

./modules/Forecasts/clients/base/api/ForecastsModuleApi.php
    protected function getClass(array $args)

./modules/Forecasts/clients/base/api/ForecastsProgressApi.php
    public function progressRep(ServiceBase $api, array $args)
    public function progressManager(ServiceBase $api, array $args)

./modules/History/clients/base/api/HistoryApi.php
    protected function scrubFields(array $args)
```

```

./modules/Home/clients/base/api/MostActiveUsersApi.php
    public function getMostActiveUsers(ServiceBase $api, array $
args)

. /modules/KBContents/clients/base/api/KBContentsApi.php
    public function relatedDocuments(ServiceBase $api, array $ar
gs)
    public function disableApi(ServiceBase $api, array $args)

./modules/KBContents/clients/base/api/KBContentsFilterApi.php
    protected function filterByContainingExcludingWords(ServiceB
ase $api, array $args, $filterArgs)

./modules/KBContents/clients/base/api/KBContentsRelateRecordApi.
php
    public function createRelatedLinks(ServiceBase $api, array $
args, $securityTypeLocal = 'view', $securityTypeRemote = 'view')

./modules/KBContents/clients/base/api/KBContentsUsefulnessApi.ph
p
    protected function vote(ServiceBase $api, array $args, $isUs
eful)
    public function voteUseful(ServiceBase $api, array $args)
    public function voteNotUseful(ServiceBase $api, array $args)

./modules/Leads/clients/base/api/LeadConvertApi.php
    public function convertLead(ServiceBase $api, array $args)
    protected function loadModule(ServiceBase $api, $module, $da
ta)
    protected function loadModules(ServiceBase $api, $modulesToC
onvert, $data)

./modules/Leads/clients/base/api/LeadsApi.php
    public function getFreeBusySchedule(ServiceBase $api, array
$args)
    protected function getAccountBean(ServiceBase $api, array $a
rgs, $record)
    protected function getAccountRelationship(ServiceBase $api,
array $args, $account, $relationship, $limit = 5, $query = array
())

./modules/Meetings/clients/base/api/MeetingsApi.php
    public function getAgenda(ServiceBase $api, array $args)
    public function getExternalInfo(ServiceBase $api, array $arg
s)

```

```

./modules/ModuleBuilder/parsers/views/AbstractMetaDataParser.php
    protected function _standardizeFieldLabels(array &$fielddefs
    )
        public static function _trimFieldDefs(array $def)
        public static function getClientStudioValidation(array $Studio, $view, $client)

./modules/ModuleBuilder/parsers/views/GridLayoutMetaDataParser.php
    public static function _trimFieldDefs(array $def)

./modules/ModuleBuilder/parsers/views/ListLayoutMetaDataParser.php
    public static function _trimFieldDefs(array $def)

./modules/ModuleBuilder/parsers/views/SubpanelMetaDataParser.php
    public static function _trimFieldDefs(array $def)

./modules/Opportunities/clients/base/api/OpportunitiesEnumApi.php
    public function getEnumValues(ServiceBase $api, array $args)

./modules/OutboundEmailConfiguration/clients/base/api/OutboundEmailConfigurationApi.php
    public function listConfigurations(ServiceBase $api, array $args)
    public function convertQuote(ServiceBase $api, array $args)
    public function createRecordList(ServiceBase $api, array $args)

./modules/Reports/clients/base/api/ReportsDashletsApi.php
    public function getSavedReports(ServiceBase $api, array $args)
    public function getSavedReportChartById(ServiceBase $api, array $args)

./modules/Reports/clients/base/api/ReportsExportApi.php
    public function exportRecord(ServiceBase $api, array $args)

./modules/RevenueLineItems/clients/base/api/RevenueLineItemsGlobalChartApi.php
    public function salesByCountry(ServiceBase $api, array $args)

./modules/SchedulersJobs/SchedulersJob.php
    protected function sudo(User $user)

```

```
./modules/Tags/clients/base/api/TagsApi.php
    public function updateBean(SugarBean $bean, ServiceBase $api
, array $args)

./modules/Teams/clients/base/api/TeamsRelateRecordApi.php
    protected function checkRelatedSecurity(ServiceBase $api, ar
ray $args, SugarBean $primaryBean, $securityTypeLocal = 'view',
$securityTypeRemote = 'view')

./modules/TimePeriods/clients/base/api/TimePeriodsCurrentApi.php
    public function getCurrentTimePeriod(ServiceBase $api, array
$args)
    public function getTimePeriodByDate(ServiceBase $api, array
$args)

./modules/Users/User.php
    public function populateFromRow(array $row, $convert = false
)

./modules/Users/clients/base/api/UsersApi.php
    public function deleteUser(ServiceBase $api, array $args)
    public function getFreeBusySchedule(ServiceBase $api, array
$args)

./modules/Users/clients/base/api/UsersRelateRecordApi.php
    protected function checkRelatedSecurity(ServiceBase $api, ar
ray $args, SugarBean $primaryBean, $securityTypeLocal = 'view',
$securityTypeRemote = 'view')

./modules/pmse_Inbox/clients/base/api/PMSECasesListApi.php
    public function selectCasesList(ServiceBase $api, array $arg
s)
    public function selectLogLoad(ServiceBase $api, array $args)
    public function clearLog(ServiceBase $api, array $args)
    public function configLogLoad(ServiceBase $api, array $args)
    public function configLogPut(ServiceBase $api, array $args)
    public function returnProcessUsersChart(ServiceBase $api, ar
ray $args)
    public function returnProcessStatusChart(ServiceBase $api, a
rray $args)

./modules/pmse_Inbox/clients/base/api/PMSEEngineApi.php
    public function getNotes(ServiceBase $api, array $args)
    public function saveNotes(ServiceBase $api, array $args)
    public function deleteNotes(ServiceBase $api, array $args)
```

```

        public function retrieveHistoryLog(ServiceBase $api, array $
args)
        public function engineRoute(ServiceBase $api, array $args)
        public function engineClaim(ServiceBase $api, array $args)
        public function reassignRecord(ServiceBase $api, array $args
)
        public function adhocReassign(ServiceBase $api, array $args)
        public function getReassign(ServiceBase $api, array $args)
        public function getFormDataReassign(array $args)
        public function getAdhoc(ServiceBase $api, array $args)
        public function getFormDataAdhoc(array $args)
        public function changeCaseUser(ServiceBase $api, array $args
)
        public function userListByTeam(ServiceBase $api, array $args
)
        public function updateChangeCaseFlow(ServiceBase $api, array
$args)
        public function reactivateFlows(ServiceBase $api, array $arg
s)
        public function cancelCase(ServiceBase $api, array $args)
        public function reassignFlows(ServiceBase $api, array $args)
        public function getReassignFlows(ServiceBase $api, array $ar
gs)
        public function getUnattendedCases(ServiceBase $api, array $
args)
        public function selectCase(ServiceBase $api, array $args)
        public function getCaseRecord(ServiceBase $api, array $args)
        public function getSettingsEngine(ServiceBase $api, array $a
rgs)
        public function putSettingsEngine(ServiceBase $api, array $a
rgs)

./modules/pmse_Inbox/clients/base/api/PMSEEngineFilterApi.php
        protected function formatBeans(ServiceBase $api, array $args
, $beans, array $options = array())

./modules/pmse_Inbox/clients/base/api/PMSEImageGeneratorApi.php
        public function getFile(ServiceBase $api, array $args)
        public function getTempImage(ServiceBase $api, array $args)
        public function getProjectFile(ServiceBase $api, array $args
)
        public function getProcessImage(ServiceBase $api, array $arg
s)

./modules/pmse_Project/clients/base/api/PMSEProjectApi.php
        public function retrieveCustomProject(ServiceBase $api, arra

```

```

y $args)
    public function updateCustomProject(ServiceBase $api, array
$args)
        protected function getLoadedAndFormattedBean(ServiceBase $ap
i, array $args)
            public function getBRFields(ServiceBase $api, array $args)
            public function getCrmData(ServiceBase $api, array $args)
            public function putCrmData(ServiceBase $api, array $args)
            public function getActivityDefinition(ServiceBase $api, arra
y $args)
            public function putActivityDefinition(ServiceBase $api, arra
y $args)
            public function getEventDefinition(ServiceBase $api, array $
args)
            public function putEventDefinition(ServiceBase $api, array $
args)
            public function getGatewayDefinition(ServiceBase $api, array
$args)
            public function putGatewayDefinition(ServiceBase $api, array
$args)
            public function verifyRunningProcess(ServiceBase $api, array
$args)

./modules/pmse_Project/clients/base/api/PMSEProjectCRUDApi.php
    public function deleteRecord(ServiceBase $api, array $args)

./modules/pmse_Project/clients/base/api/PMSEProjectImportExportA
pi.php
    public function projectDownload(ServiceBase $api, array $arg
s)
    public function projectImport(ServiceBase $api, array $args)

./modules/pmse_Project/pmse_BpmFlow/pmse_BpmFlow.php
    public function populateFromRow(array $row, $convert = false
)

```

Supported Platforms

For information on supported platform components, see [Sugar 8.0.x Supported Platforms](#).

Note: After upgrading to PHP 7.1, you may encounter `ArgumentCountError` errors

in your PHP error log related to customizations or integrations. Make sure to check out the recent blog post on [PHP 7.1 pitfalls](#) to help address these issues.

Upgrade Paths

Package	From Version(s)	MySQL	SQLServer
New Installs		<input type="checkbox"/>	<input type="checkbox"/>
7.9.4.0-to-8.0.0	7.9.4.0	<input type="checkbox"/>	<input type="checkbox"/>
7.11.x-to-8.0.0	7.11.0.0 - 7.11.1.0	<input type="checkbox"/>	<input type="checkbox"/>

Last Modified: 2020-08-11 21:08:21