Sugar Developer Webinar - Q2 2020 (10.0.0)

Michael Shaheen
Developer Advocate

Matt Marum
Director, ISV Alliances

March 17 & 18, 2020

Speaker Bios

Michael Shaheen - Developer Advocate

- 20+ year full stack developer
- 6 years as an instructor
- Certified Scrum Master
- With Sugar since April 1, 2019
- 2 Teenage Daughters
- 2 - 4 cats
- Love baking

Matt Marum - Director, ISV Alliances

- Developer, Solution Architect, PM
- With Sugar since August, 2012
- #TWINNING
- At home in Raleigh, NC

© 2019 SugarCRM, Inc. All rights reserved.

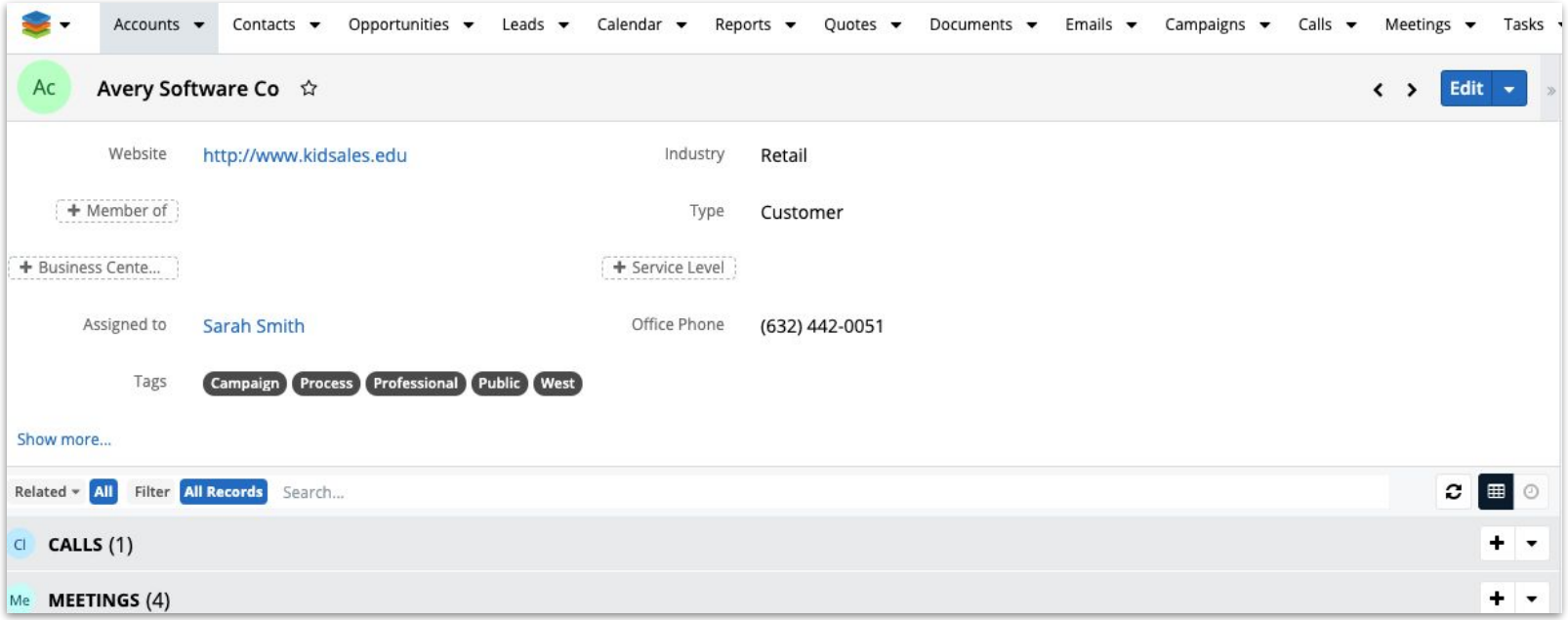
Agenda

- Field Label Placement
- Product Catalog Module
- Relate Fields Denormalization
- API Updates
- New Product Teaser
- Other Updates
- Q&A

© 2019 SugarCRM, Inc. All rights reserved.

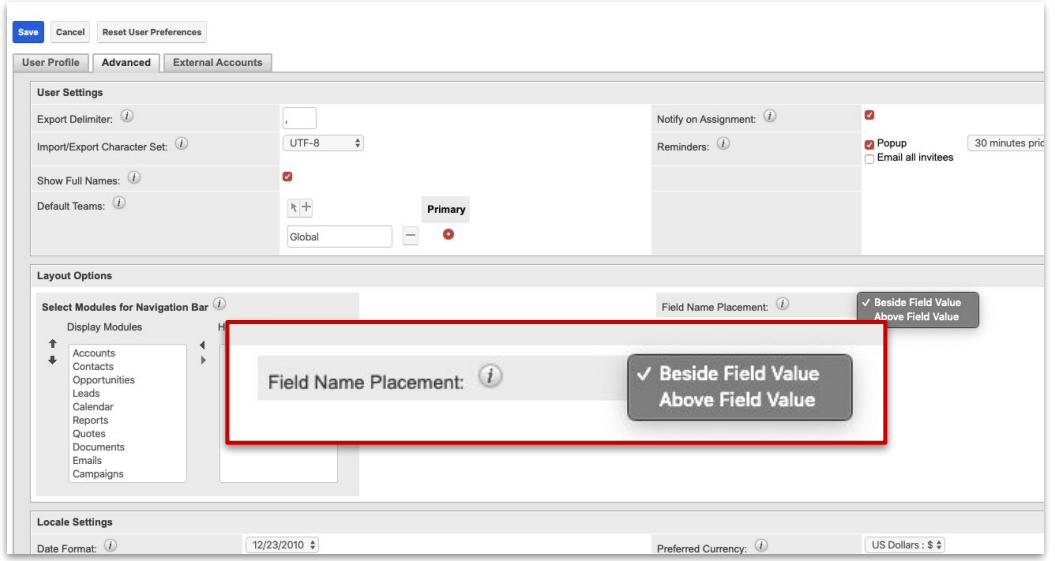
Field label placement

- By default, record view field label positioning will be to the left of the field
- Fixing the “white space issue” on record views



© 2019 SugarCRM, Inc. All rights reserved.

Field label placement



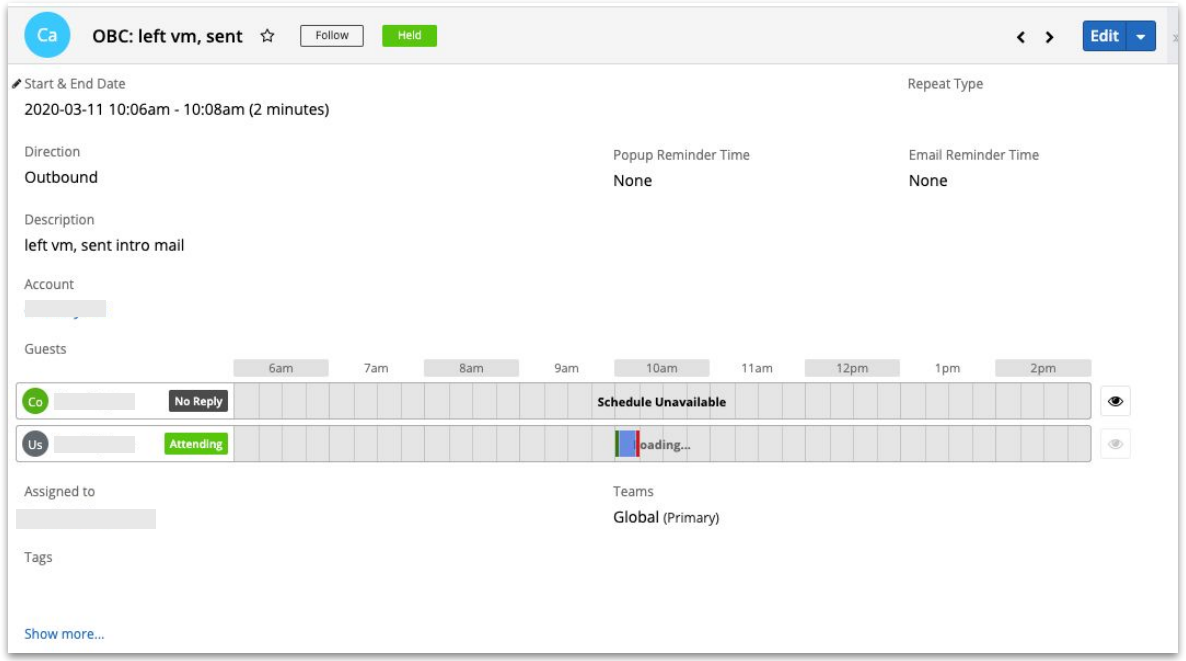
- user profile setting called "Field Name Placement"
- this is per-user not global
- potential values:
  - Beside Field Value
    - default
    - labels in record views for this user will appear to the left of the fields
  - Above Field Value
    - labels in record views for this user will appear top of the fields
- this setting is replacing the labelsOnTop viewDef which is deprecated as of this release.
- there is no code option to set this value - only the user profile setting.
- By default all users will be set to "Beside Field Value" regardless of any previous settings

© 2019 SugarCRM, Inc. All rights reserved.

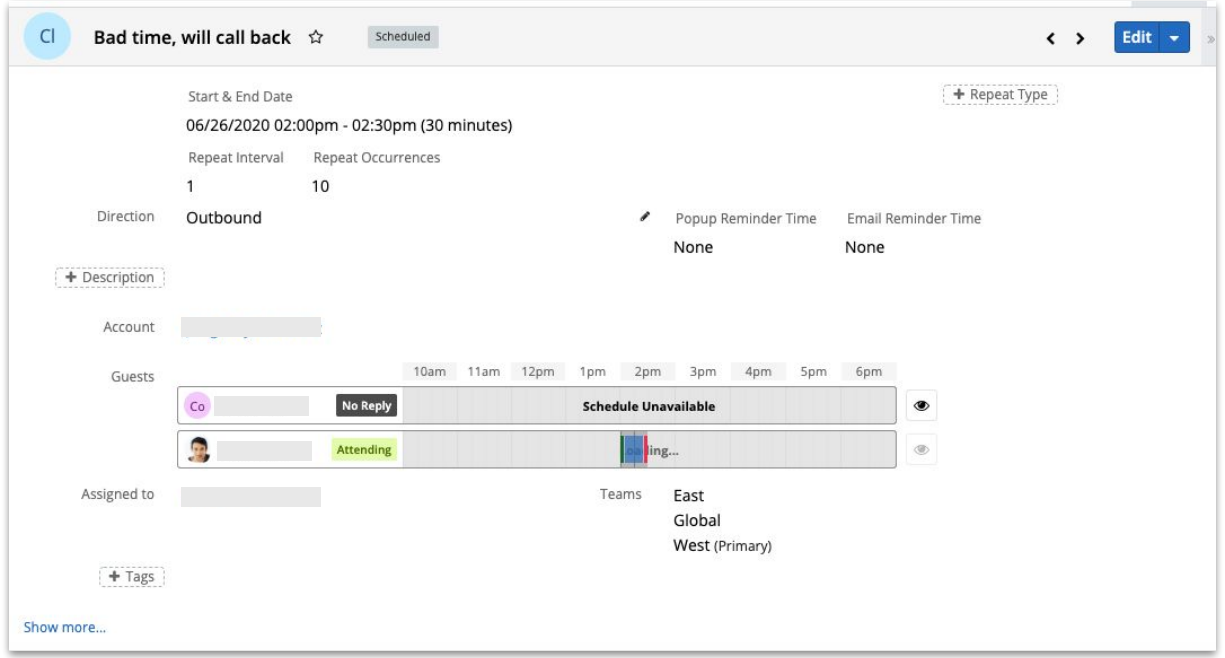
Field label placement

- Calls and Meetings don’t play nice with side labels.
  - They don’t exactly break, but the layouts become a bit less aligned.
  - Since side-labels are enabled by default, these views may look a bit “off” in 10.0 (unless you do a custom CSS fix)

Above Field Value (pre 10.0)



Beside Field Value (new)



© 2019 SugarCRM, Inc. All rights reserved.

Field Label positioning CSS fix example

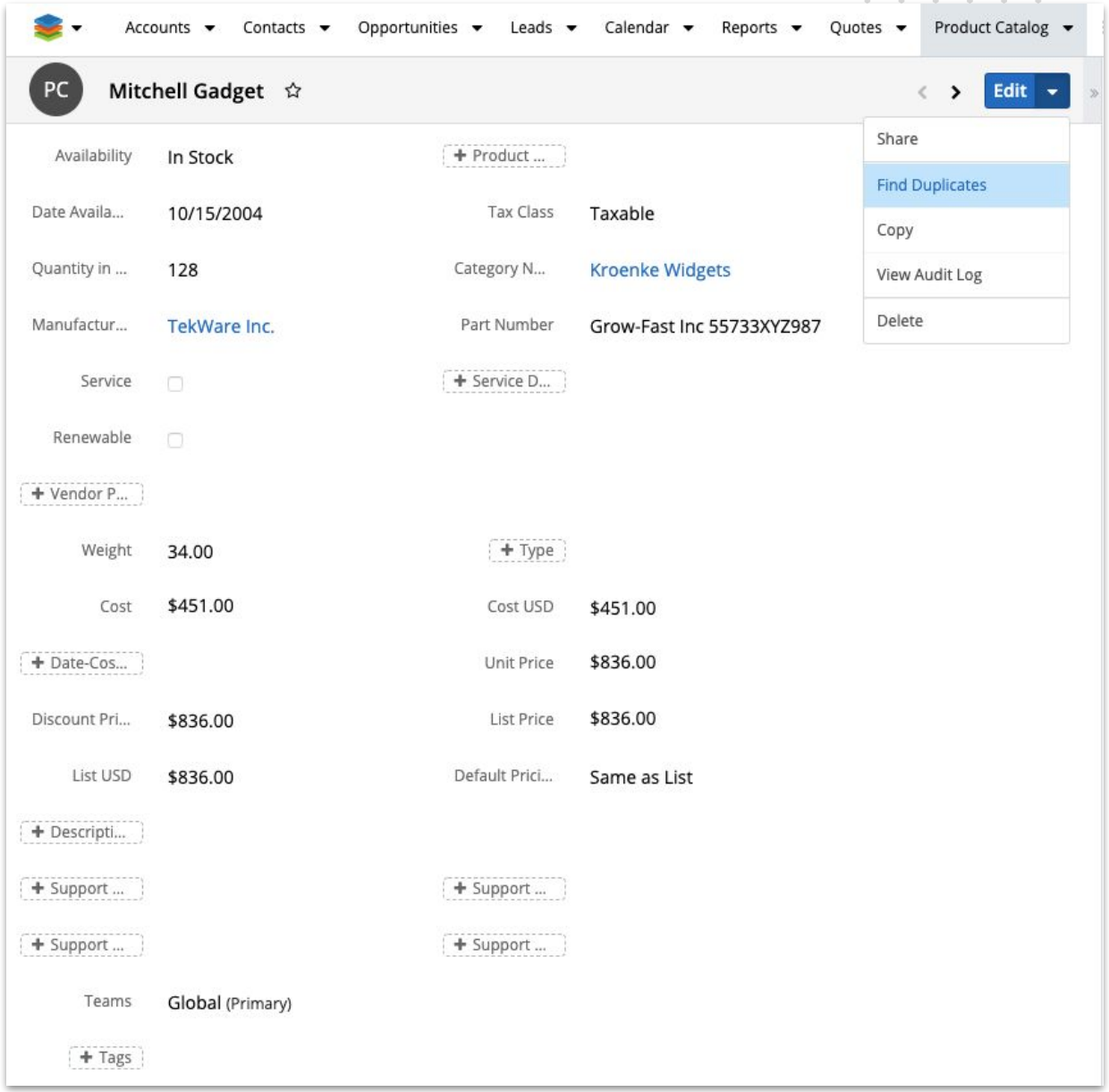
- If this new setting conflicts with your customizations, try a CSS fix
- Created an MLP with custom.less file
- This is just the basics.
  - could use A LOT more adjustments depending on your customizations

```
.row-fluid {
    .labels-on-side {
        [class*="span"] {
            min-height: initial;
            float: none;
        }
        .record-label-wrapper {
            margin-bottom: 0px;
            .record-label {
                float: none;
            }
        }
        .record-link-wrapper {
            margin-left: 1px;
        }
    }
}
```

© 2019 SugarCRM, Inc. All rights reserved.

Product Catalog

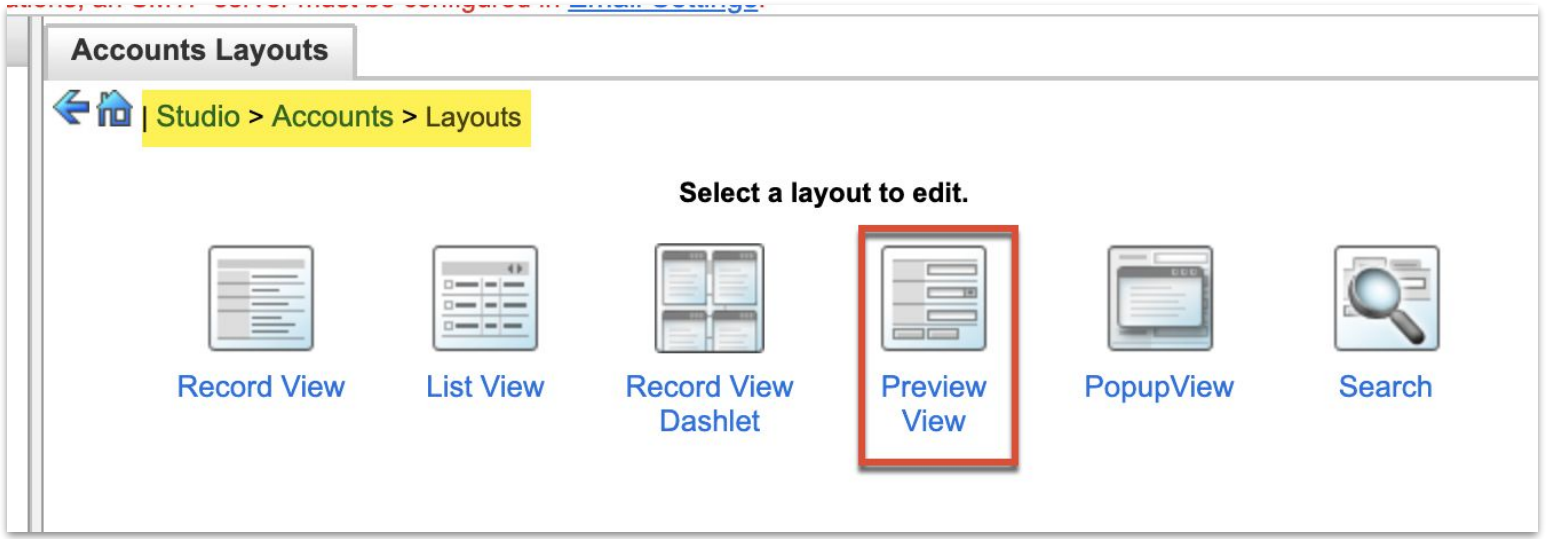
- The product catalog module is now available to all users OOTB
- all records will be associated with the global team automatically upon upgrade
- Administrators can restrict access to the Product Catalog module through roles and ACLs just like other standard modules
- “View audit log” button is now available on Product Catalog’s record view (like other standard Sugar modules)



© 2019 SugarCRM, Inc. All rights reserved.

Edit Preview View Layout

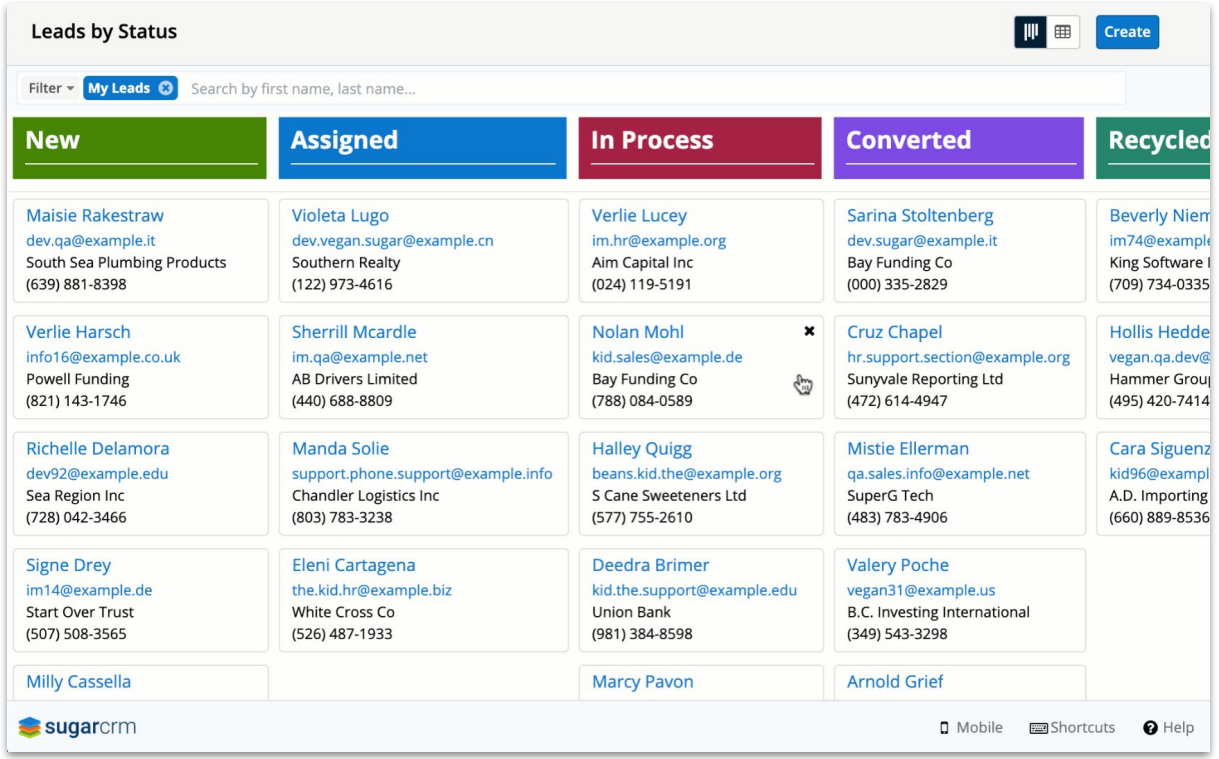
- The ability to customize the Preview View has been added
- Previously, Preview would use same basic layout as record view
- Could only be changed via customization
- Now it can be configured in Studio to be unique



© 2019 SugarCRM, Inc. All rights reserved.

Leads Tile View

- Tile view has been added for Leads module
- Available OOTB for new customers
- For existing customers, it must be enabled in Tile View Configuration screen
- Lead conversion process can be initiated from Tile view by dragging a Lead tile to the “Converted” column
- Note: you cannot move a tile back out of the “Converted” column (you can’t un-convert a lead)



© 2019 SugarCRM, Inc. All rights reserved.

Enterprise updates

- Ent now has the following Dashlets available:
  - Record Dashlet
  - Interactions Dashlet
  - Comments Log Dashlet
- In 10.0, Ent on-prem will get all Ent features that were released in 9.x
  - 9.0 Release Notes
  - 9.1 Release Notes
  - 9.2 Release Notes
  - 9.3 Release Notes

© 2019 SugarCRM, Inc. All rights reserved.

Relate Fields Denormalization

- To reduce redundancy and to ensure data consistency, we normalize data by utilizing relate-to (lookup) fields
- This means, however, that a query for a list view will have joins - and those relate fields cannot be indexed
- So, sorting a list view can take a very long time (depending on your data)
- To combat this long query, we can now DENORMALIZE individual relate-to fields
  - add copies of the data to the parent table
  - use logic hooks to handle future updates
- This will likely mean increased time when updating a parent record
- So you decide: what is more important in your situation? faster sorting on list views or faster update on (for example) an Opportunity record

© 2019 SugarCRM, Inc. All rights reserved.

Relate Fields Denormalization

- Sugar administration tool
- optimize the DB structure for faster sorting and data load of relate-type field values on list view pages
- Copies the values of stock Sugar relate fields from their own tables into a field in the table of the related module.
  - field will be prefixed with denorm_
  - only used for sorting
    - original field still used for display
    - denorm_ field is NOT returned in API calls
  - depending on size of table, this could take a long time (10 min, 20 min, 1 hour!)
  - runs via cron job
  - runs in the background and shows (a pretty accurate) estimate of time remaining
  - if, for any reasons, you cannot utilize cron jobs, the SQL is supplied in the admin tool so you can run it yourself

© 2019 SugarCRM, Inc. All rights reserved.

Relate Fields Denormalization

- logic hooks will continue to monitor any value changes made to the relate field, the source field which the relate field points to, and to the linked record ID. The logic hook will also react to changes made to the relationship.
- process can be reversed by moving the denormalized field back to the original field list in the admin tool
  - this will remove the logic hooks and the system will go back to using the original fields
  - it WILL NOT remove the denorm_ field from the DB (may be added to 10.1)
  - for now, do not cancel during the process. wait until it completes and then reverse it

© 2019 SugarCRM, Inc. All rights reserved.

Home Dashboard list limit

- The maximum number of Dashboards that will display in the Home Dashboard List has been increased from 20 to 50

© 2019 SugarCRM, Inc. All rights reserved.

REST API changes

© 2019 SugarCRM, Inc. All rights reserved.

REST API Updates

Deprecated endpoint

GET: /ExpressionEngine/:record/related

Replaced with

POST: /ExpressionEngine/:record/related

GET couldn’t handle long URIs that resulted from SugarLogic related value formulas.

```
{
  "id": "a915cff8-1dec-11ea-b203-7200015efab0",
  "action": "related",
  "module": "Opportunities",
  "fields": [
    {
      "type": "rollupConditionalSum",
      "link": "revenuelineitems",
      "relate": "likely_case",
      "condition_field": "sales_stage",
      "condition_expr": "forecastSalesStages(\"true\",\"false\")"
    },
    {
      "type": "maxRelatedDate",
      "link": "revenuelineitems",
      "relate": "date_closed"
    },
    {
      "type": "rollupMax",
      "link": "revenuelineitems",
      "relate": "date_closed_timestamp"
    },
    {
      "type": "rollupConditionalSum",
      "link": "revenuelineitems",
      "relate": "best_case",
      "condition_field": "sales_stage",
      "condition_expr": "forecastSalesStages(\"true\",\"false\")"
    },
    {
      "type": "rollupConditionalSum",
      "link": "revenuelineitems",
      "relate": "worst_case",
      "condition_field": "sales_stage",
      "condition_expr": "forecastSalesStages(\"true\",\"false\")"
    },
    {
      "type": "count",
      "link": "revenuelineitems"
    },
    {
      "target": "closed_revenue_line_items",
      "type": "countConditional",
      "link": "revenuelineitems",
      "condition_field": "sales_stage",
      "condition_expr": "createList(\"Closed Won\",\"Closed Lost\")"
    },
    {
      "target": "included_revenue_line_items",
      "type": "countConditional",
      "link": "revenuelineitems",
      "condition_field": "commit_stage",
      "condition_expr": "forecastIncludedCommitStages()"
    }
  ]
}
```

© 2019 SugarCRM, Inc. All rights reserved.

Module Loader API

- Module loader is the utility used to get your customizations into your instance of Sugar
- Has a package scanner to ensure the safety etc of what you are uploading
- Can be cumbersome if you are making a lot of tweaks etc during development
  - Go into admin
  - Uninstall previous version
  - Delete previous version
  - Upload new version
  - Install new version
  - repair & Rebuild
- New endpoints allow for automation

© 2019 SugarCRM, Inc. All rights reserved.

Module Loader API

- /Administration/packages/:unFile
  - DELETE endpoint.
  - Delete the specified package.
  - Package 'unFile' hash must be provided.
  - Package 'unFile' hash can be found in staging package list.
- /Administration/packages/:id/disable/
  - GET endpoint.
  - Disable the specified package.
  - Package id hash must be provided.
  - Package id can be found in installed package list.
- /Administration/packages/:id/enable/
  - GET endpoint.
  - Enable the specified package.
  - Package id hash must be provided.
  - Package id can be found in installed package list.
- /Administration/packages/:file_install/install/
  - GET endpoint.
  - Installs the specified package.
  - Package 'file_install' hash must be provided.
  - file_install hash could be found in staging package list.
- /Administration/packages/
  - GET endpoint.
  - Lists all packages in the system.
- /Administration/packages/installed/
  - GET endpoint.
  - Lists all installed packages.
- /Administration/packages/staged/
  - GET endpoint.
  - Lists staged packages in the system.
- /Administration/packages/:id/uninstall/
  - GET endpoint.
  - Uninstalls the specified package.
  - Package id hash must be provided.
  - Package id can be found in installed package list.
- /Administration/packages
  - POST endpoint.
  - Uploads a module loadable package via a multi-part form request.
  - The only parameter is 'upgrade_zip' which must contain a valid module loadable zip file.

© 2019 SugarCRM, Inc. All rights reserved.

MLP API example

Pay attention to the help docs (/rest/v11_8/help) for which values are needed for each call

1. is the MLP already in the system?
   GET: /Administration/packages


```
{
    "packages": [
        {
            "name": "No Side Labels",
            "version": "1.2.1",
            "type": "module",
            "published_date": "2020-03-11 10:31:09",
            "description": "Updates the UI to override side field label
                placement",
            "uninstallable": true,
            "file_install": "8d177e59ae0c4de2242156c63fef6185",
            "file": "8d177e59ae0c4de2242156c63fef6185",
            "enabled": "ENABLED",
            "id": "6ed080ea-6383-11ea-902b-0242ac120006",
            "installed": true
        }
    ]
}
```


2. grab the id to uninstall
   GET: /Administration/packages/:id/uninstall

3. upload MLP - be sure it is a proper .zip package
   POST: /Administration/packages


```
{
    "file_install": "1b4ec7ff30109a02a07d42524f48d14e",
    "unFile": "1b4ec7ff30109a02a07d42524f48d14e"
}
```


4. was it uploaded?
   GET: /Administration/packages/staged


```
{
    "packages": [
        {
            "name": "No Side Labels",
            "version": "1.2.1",
            "published_date": "2020-03-10 14:22:24",
            "description": "Updates the UI to override side field label
                placement",
            "uninstallable": "Yes",
            "type": "Module",
            "file": "ec559c87ddec5b45c3186e747d800c32",
            "file_install": "1b4ec7ff30109a02a07d42524f48d14e",
            "unFile": "1b4ec7ff30109a02a07d42524f48d14e"
        }
    ]
}
```


5. grab the file_install value to install MLP
   GET: /Administration/packages/:hash/install

© 2019 SugarCRM, Inc. All rights reserved.

MLP API error

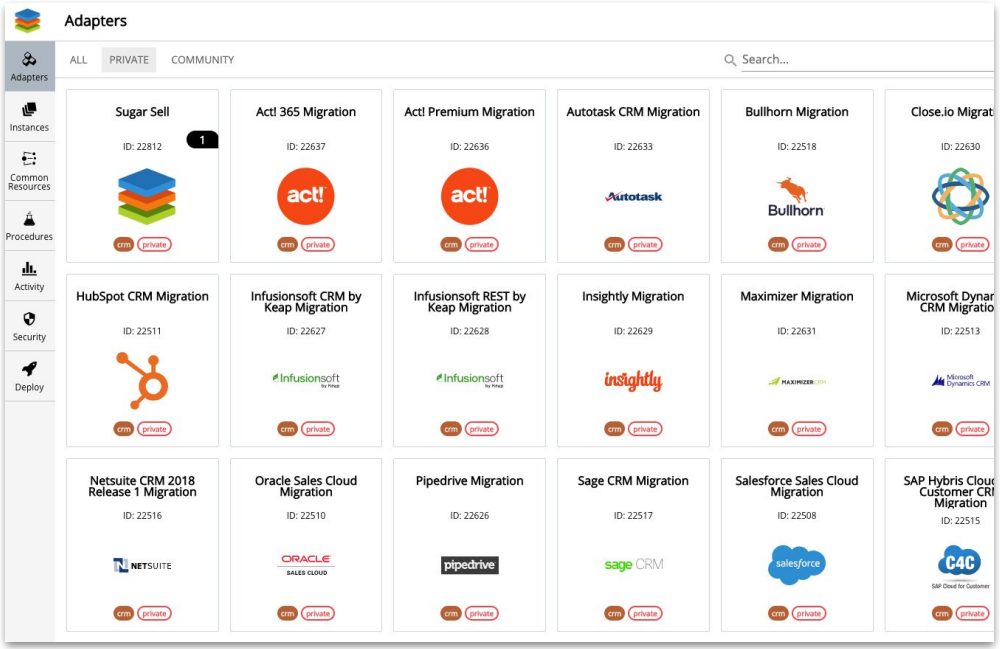
- Package Scanner will run when you call the installation endpoint (/Administration/packages/:hash/install)
- If a violation is found
  - you will still receive a 200 status
  - the response body will contain HTML listing out the errors
- Note: uninstall is possible even if the uninstallable flag is set to false

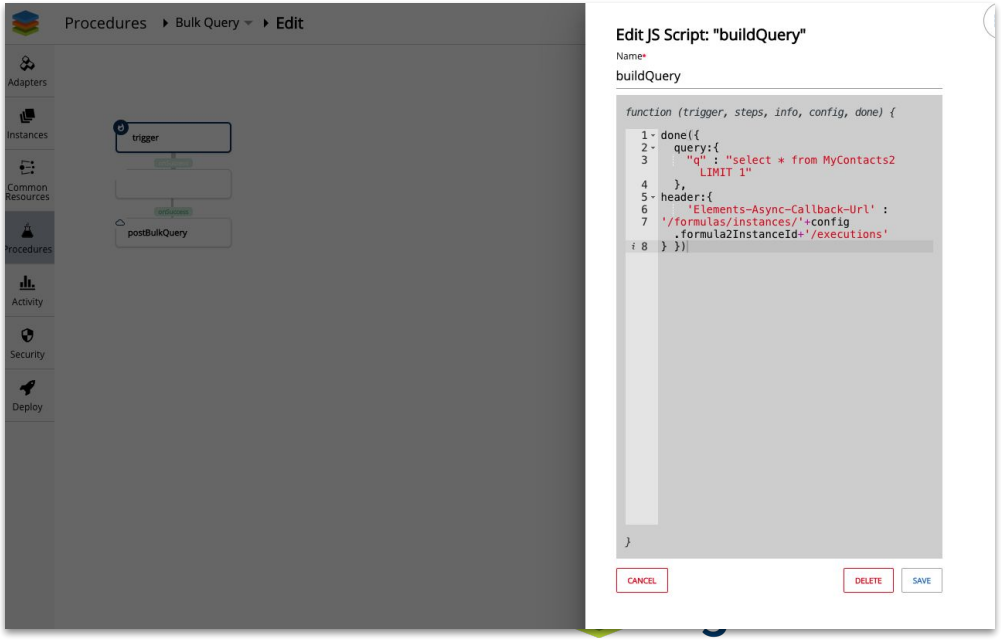
```
<h2>Scanning Package</h2><BR>
<h2 class="error">Installation failed!</h2><br>
<p>The package you are attempting to install does not conform to the policies
    established within the Sugar Open Cloud or
    by your system administrator.</p>
<ul>
    <li>Sugar Open Cloud customers must obtain a new package from the package
        provider that addresses the issues
        described below.
    <li>If you are running Sugar locally, you can relax your Module Loader
        restrictions to allow the package to be
        installed.
</ul>
</p><br> <a href="http://www.sugarcrm.com/crm/product_doc.php?
    module=FailPackageScan&version=10.0.0&edtion=ENT"
    target="_blank">Package Scan Policy Guide</a><br><br>
<div class="error">
    <h2>File Issues</h2>
</div>
<div id="detailsfile">
    <div style="position:relative;left:10px"><b>badfile.zip</b></div>
    <div style="position:relative;left:20px">Invalid file extension <br></div>
    </div><br><input class='button' onclick='document.location.href="index.php?
        module=Administration&action=UpgradeWizard&view=module"' type='button'
        value="Back to Module Loader" />
```

© 2019 SugarCRM, Inc. All rights reserved.

Coming soon

- We will be offering a new product for integrations and migrations.
- iPaaS that facilitates connecting disparate systems with SugarCRM products
- Can create procedures to help sync data between 2 systems via pre-made adapters
- Great for migrating from another CRM to Sugar
- Can set up events so that when a value changes in QuickBooks (for example), a corresponding record in Sugar will be updated
- Available for both cloud and on-premise (Sell, Serve, Pro, Enterprise)
- There will be pre-made adapters but users can also create custom adapters from scratch





© 2019 SugarCRM, Inc. All rights reserved.

New Integrations Product Plan

- Plan to offer access to a staging environment
- Will be starting to on-board partners soon
- Preparing resources in Sugar university
- Planning developer workshops.
- Creating Community blog posts
- Watch the Community and your e-mail for more information in the coming weeks

© 2019 SugarCRM, Inc. All rights reserved.

© 2019 SugarCRM, Inc. All rights reserved.

Sugar Upgrade Reminders

SugarCloud upgrades

- 7 day notice prior to upgrade
- Can be extended to 30 days after release
- Customers can request a scheduled upgrade

Sugar 8.0.x End-of-Life

- Security only support began **October 30, 2019**
- End of life is **April 30, 2020**
- *Support consists of 18 months of full maintenance and security support followed by 6 months of security-only support for a total of 24 months of support*

https://support.sugarcrm.com/Resources/Sugar_Cloud_Policy_Guide/#Upgrade_Policy

https://support.sugarcrm.com/Resources/Supported_Versions/#End-of-Life_Plan

© 2019 SugarCRM, Inc. All rights reserved.

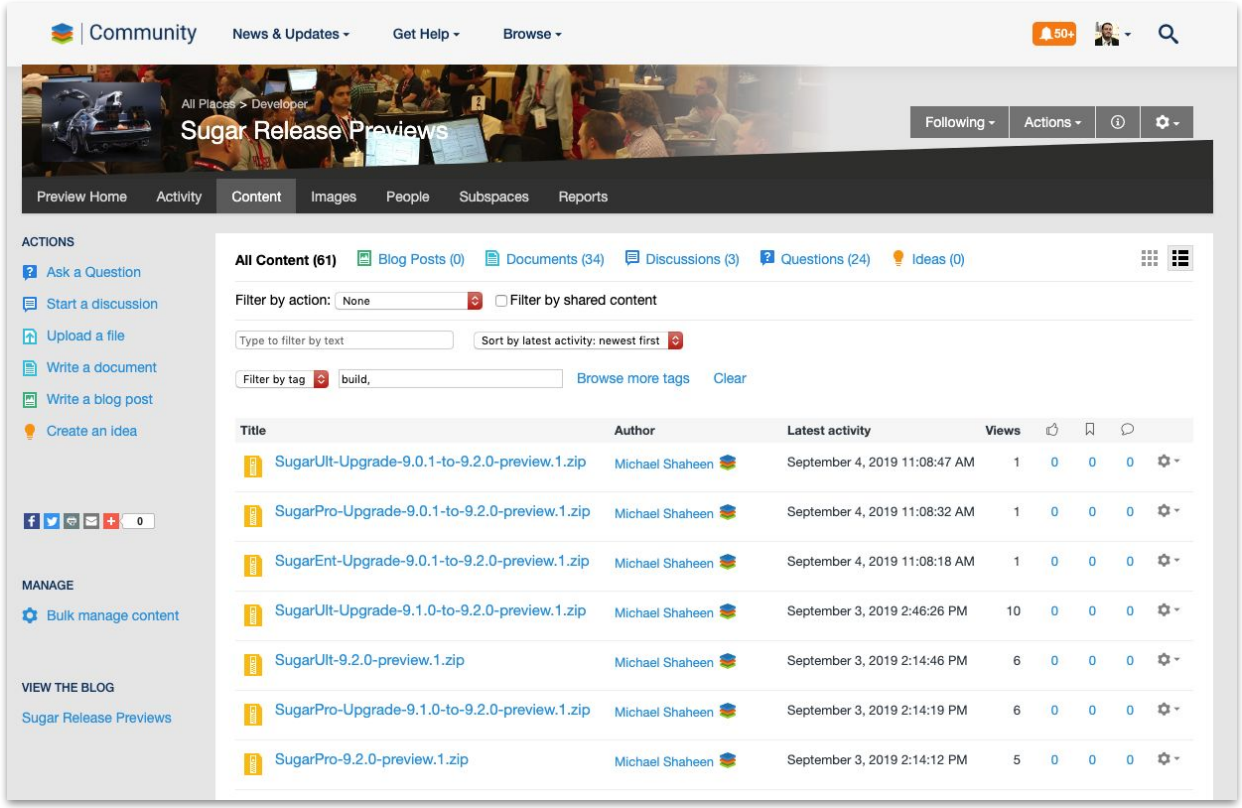
sugarcrm

Sugar Release Previews

© 2019 SugarCRM, Inc. All rights reserved.

Sugar Release Previews

- We offer preview builds prior to each release
- Usually 2 drops
- Great for testing out how your customizations will work with the upgrade
- Helps us with testing
- We'd like to offer it more widely
  - So let us know if you'd like to participate
  - developers@sugarcrm.com



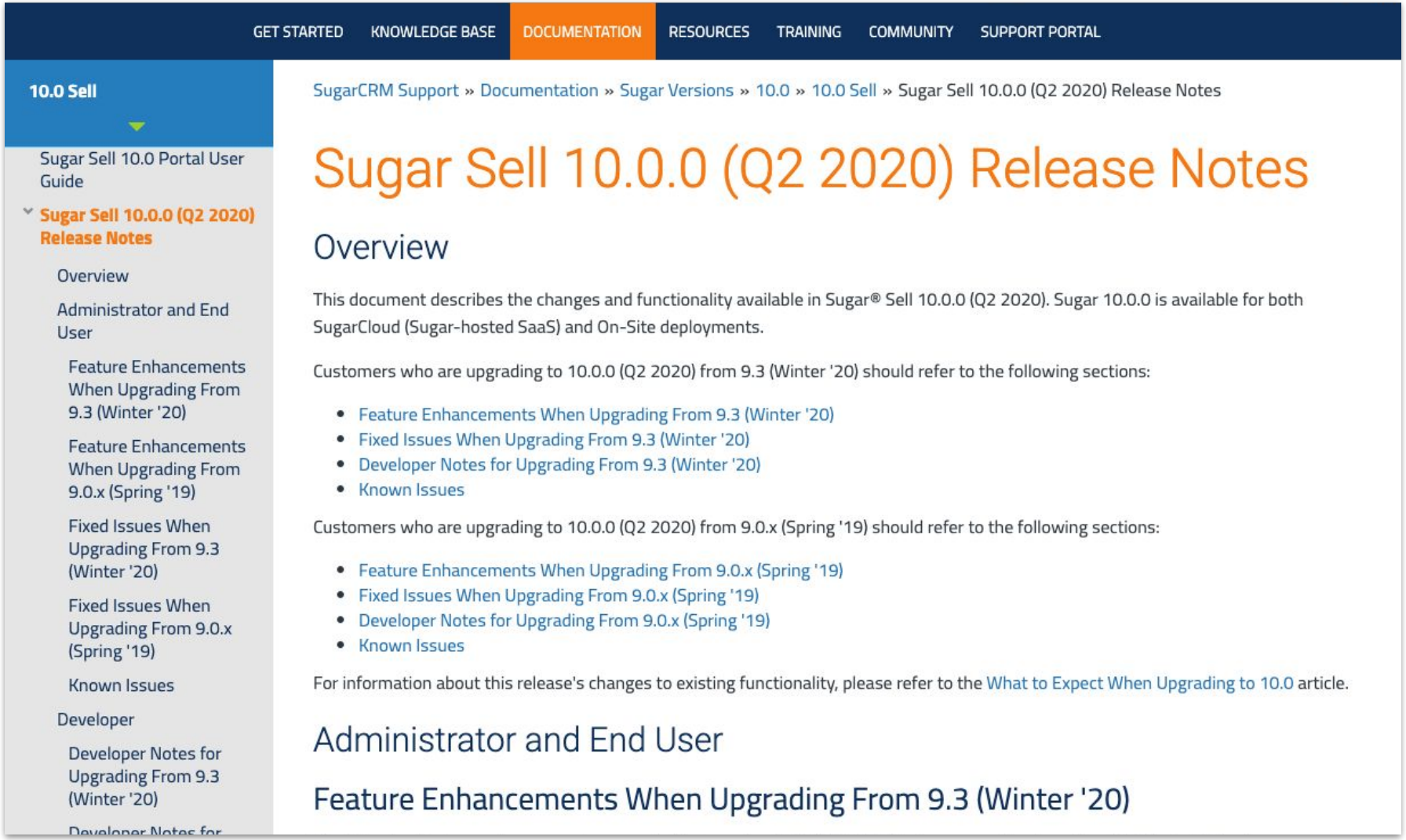
© 2019 SugarCRM, Inc. All rights reserved.

sugarcrm

Additional Development Resources

© 2019 SugarCRM, Inc. All rights reserved.

Sugar 10.0 (Q2 2020) Developer Release Notes



© 2019 SugarCRM, Inc. All rights reserved.

sugarcrm

Sugar 10.0 (Q2 2020) Migration Guide

© 2019 SugarCRM, Inc. All rights reserved.

sugarcrm

Sugar 10.0 (Q2 2020) Unit Tests

Sugar Unit Test Repository (unit-tests)

Please read the Test Tools License Terms before using. These terms are located in LICENSE within this repository

These unit tests contain both server side (PHP) and client side (JavaScript) test suites. PHP unit tests are implemented in PHPUnit. JavaScript tests are implemented using Jasmine.

Check out the branch on this repository that matches the version of Sugar that you are using. The specific steps to run the tests may vary from Sugar version to Sugar version as we adopt new test technology and expand on the framework.

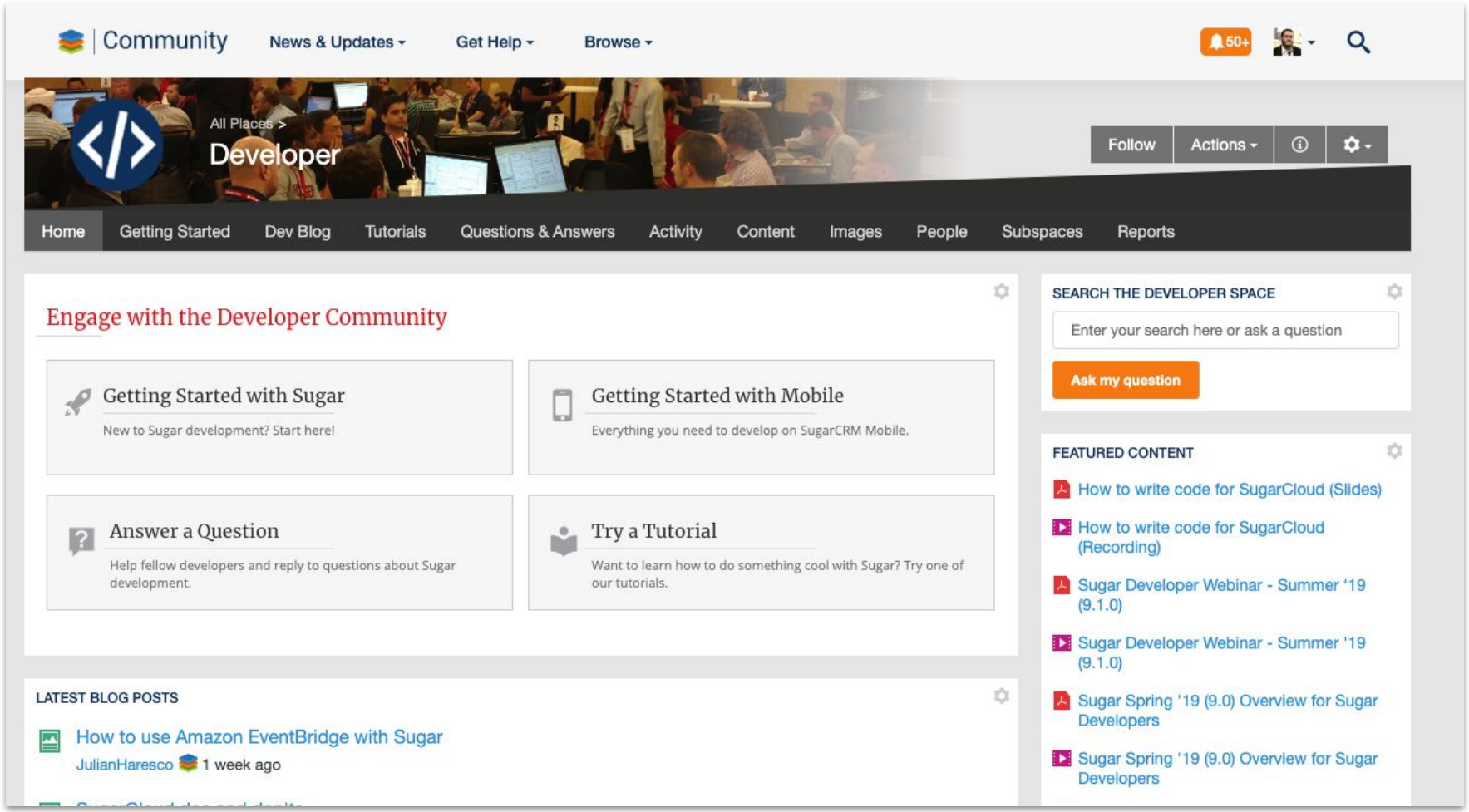
Repository Structure

The test suites are separated into different branches based upon Sugar version. Check out the branch for the version of Sugar you are using.

© 2019 SugarCRM, Inc. All rights reserved.

sugarcrm

Sugar Developer Community

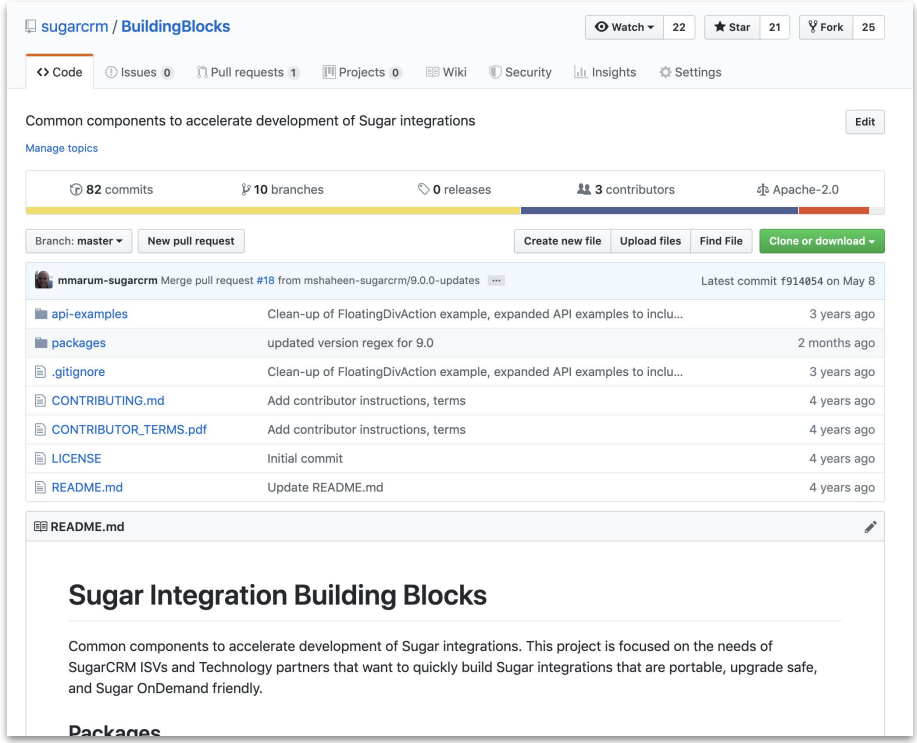


https://community.sugarcrm.com/docs/DOC-4842-on-boarding-new-sugar-developers

© 2019 SugarCRM, Inc. All rights reserved.

sugarcrm

Open-Source Repos



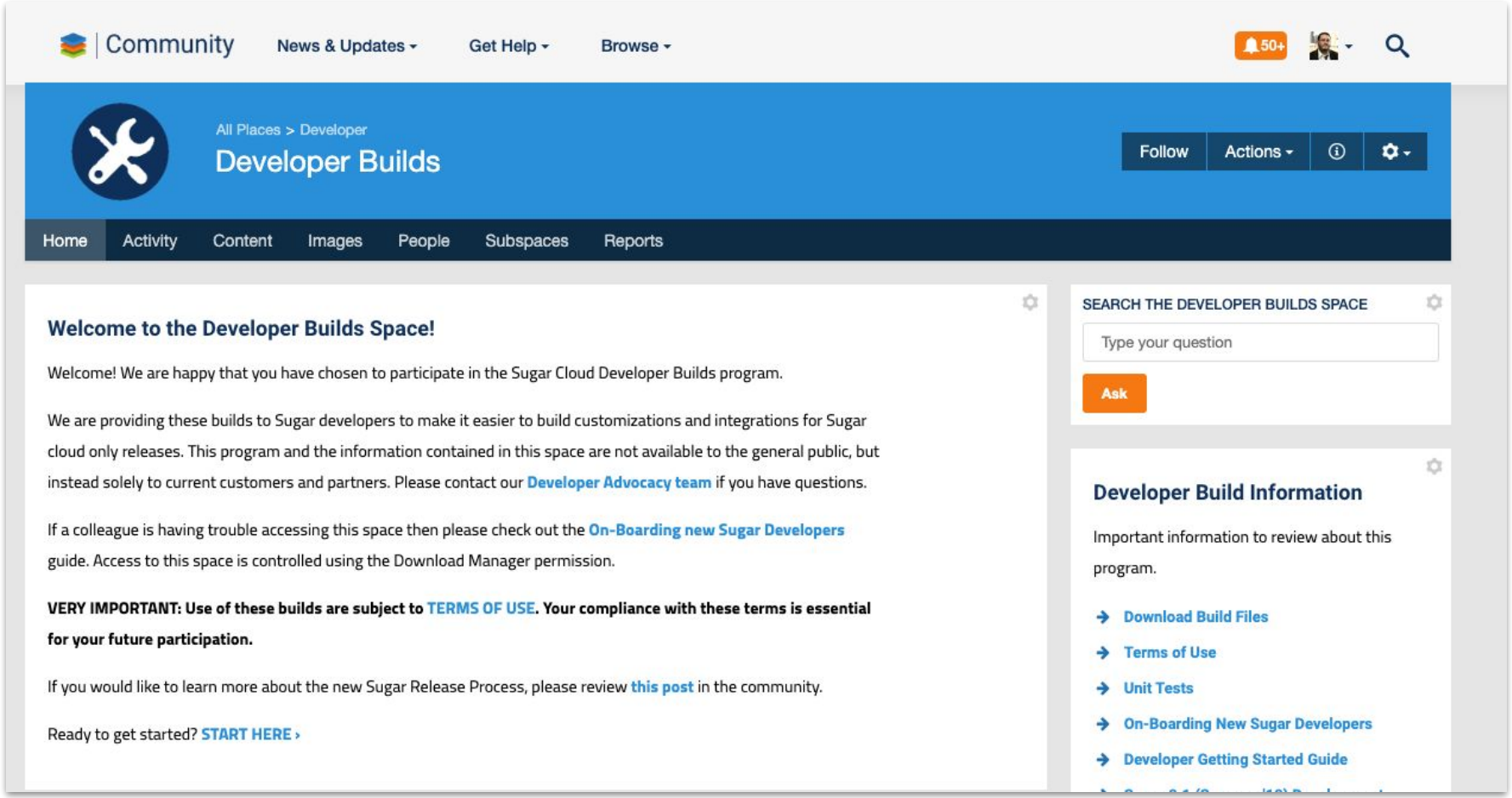
Building Blocks - https://github.com/sugarcrm/BuildingBlocks

Professor M - https://github.com/sugarcrm/school

© 2019 SugarCRM Inc. All rights reserved.

Developer Builds

- https://community.sugarcrm.com/community/developer/developer-builds
- Sugar 10.0/Q2 2020 build coming soon!



© 2019 SugarCRM Inc. All rights reserved.

Questions?

© 2019 SugarCRM, Inc. All rights reserved.