Customer Feedback Surveys for Social Customer Service and LiveMessage

Salesforce, Summer ’20

@salesforcedocs

Last updated: September 21, 2020

CONTENTS

CUSTOMER FEEDBACK SURVEYS APP

EDITIONS

Available in:

- Salesforce Classic
- Enterprise
- Performance
- Unlimited

Send Customer Satisfaction (CSAT) or Net Promoter Surveys (NPS) to Twitter or Facebook users after you've resolved their cases. Surveys are delivered as direct messages for Twitter and as private messages for Facebook, and responses come straight back to Salesforce. This app provides 10 NPS questions and 38 CSAT questions to choose from.

After you've closed a Twitter or Facebook case, you can send a customer only one survey question at a time. You can ask either an (NPS) question like "How likely are you to recommend <company name> to a friend?" or a (CSAT) question like "What is your overall satisfaction with <company name>?". NPS questions get a 10-point numerical answer scale ranging from 0 which means "Not likely" to 10 which means "Very likely." CSAT questions get 5-point answer scale ranging from "Very dissatisfied" (frowning face) to "Very satisfied" (grinning face).

We provide you with 10 NPS question variations and 38 CSAT question variations to choose from.

In addition to answering the survey question, users have the option to include a comment with their response. If they answer 7 or higher for a NPS question or "Satisfied" or "Very Satisfied" for a CSAT question, they'll also see an option to Tweet publicly about their positive experience.

For each Twitter handle or Facebook Page that is enabled for surveys, Twitter and Facebook will only allow one survey card to be sent to an end user every 24 hours. We don't want to overwhelm your customers with surveys.

The survey cards are sent as direct messages or private messages (or SMS with link for LiveMessage if channel is SMS). Therefore it is a better experience if the case interaction has ended with the exchange of direct messages or private messages rather than public messages. That makes the survey card a seamless follow-up to the case. However, if you know that you can send a direct message or private message to the user you can still send them the survey card -- just be aware that it will appear in their messages inbox screen not in the public conversation they've been having with you.

As of this publication, Twitter localizes the survey cards into the following 39 languages on their native Web client and iOS and Android mobile apps. This package is localized for eight languages:

- BR Portuguese (Brazil)
- DE German
- EN English
- ES Spanish
- FR French
- IT Italian
- JP Japanese
- NL Dutch

Note: ***We have added LiveMessage channels (Facebook and SMS) for sending surveys through related Case object. If you have LiveMessage managed package installed in the org and have a Facebook page or SMS number enabled you'll be able to send Feedback Surveys for those channels too!***

Important: Twitter Customer feedback cards are only available in the iOS or Android Twitter apps. They don't display on twitter.com, mobile.twitter.com, TweekDeck, or third party Twitter apps, even if the feedback request was sent from a website or third-party app.

INSTALL AND CONFIGURE THE CUSTOMER FEEDBACK SURVEYS APP FOR SOCIAL CUSTOMER SERVICE AND LIVEMESSAGE

Prerequisites and instructions to install and configure the Customer Feedback App.

Prerequisites for using with SCS:

Prior to installing the Customer Feedback Surveys app to your Salesforce org, Social Customer Service must be enabled.

Here is how to check if your organization has Social Customer Service enabled:

1. From Setup, type *soc* in the `Quick Find` box in the setup sidebar menu.
2. Choose **Social Customer Service**, which is located under the Social Media section.
3. If the **Enable Customer Service** checkbox is not already enabled, enable it.
4. You need at least one Twitter or Facebook social account that is already active with the Social Customer Service feature, and have access to the username and password of those accounts.

My domain must be enabled and deployed to your users.

1. From Setup type *Domain Management* in the `Quick Find` box, click **My Domain** under Domain Management.
2. Enter a URL.
3. Click **Register Domain**.

Note: After you click Register Domain, Salesforce takes a few minutes to update its naming registries. You receive an email when it's done.

EDITIONS

Available in:

- Salesforce Classic
- Enterprise
- Performance
- Unlimited

USER PERMISSIONS

To install apps and administer Social Customer Service:

- "Manage Users"
- "Customize Application"

To send requests and view company information:

- "View Setup and Configuration"

To create case feed items:

- "Feed Tracking for All Related Objects on the Case object"

You need an active Force.com Site for a unique URL to identify your Salesforce organization so we know where exactly to send survey data.

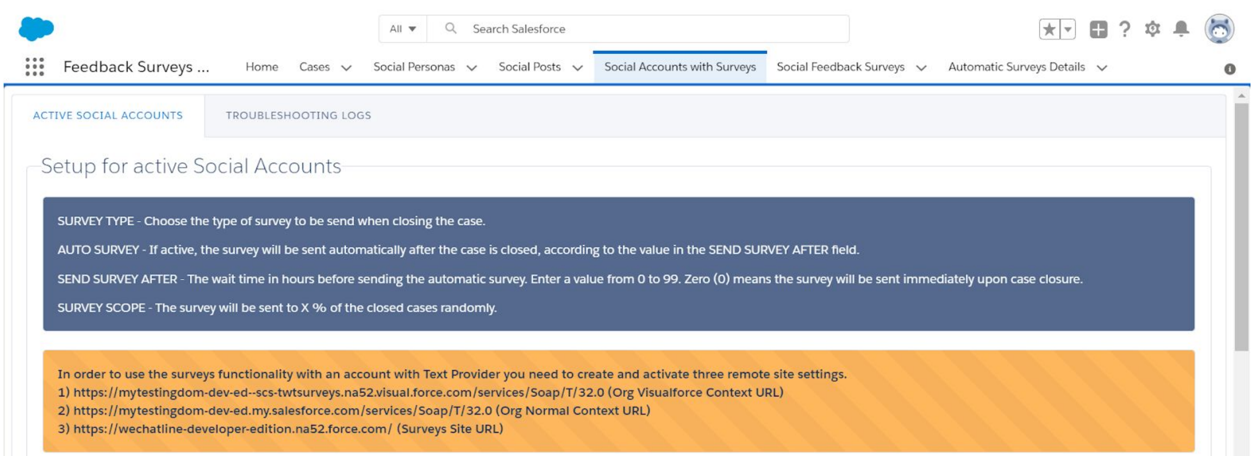
1. Type *sites* in the `Quick Find` box in the setup sidebar menu.

2. Navigate to the Sites page (it's under the Develop section).
3. If you don't already have a Force.com Domain, register one.
4. If you don't already have a Site, make one with a name like "Surveys" and a default Web address like `https://<yourdomain>.force.com/surveys`. Make sure to activate it by selecting the **Active** checkbox. If you need to select a site homepage, you can choose the default UnderConstruction page.
5. Ensure Require Secure Connections (HTTPS) checkbox is marked for the Site.
6. Once you have an active Site, you need to note its secure URL. You can find this by going to the main Sites page that lists all of your organization's Sites. From the list on that page, find the correct link in the Site URL column. Click on the link in the Site URL column, to open that URL in a new browser window. From that new browser window, copy the URL from the address bar. Save this URL address somewhere convenient: you'll need it soon.

Note: If you made a new Force.com Site just for surveys as described in the above step, the URL will be like this: `https://<your domain>.secure.force.com/surveys`.

Prerequisites for using with LIVEMESSAGE :

1. You need to have the LiveMessage managed package installed in the org (and properly setup).
2. You need to have a Facebook page and (or) an SMS number enabled for use with LiveMessage.
3. You need to add three **Remote Site settings** for using SMS (Text) channel with no issue, as soon as you install (or upgrade) latest version of the package you'll be asked to add mentioned Remote Site Settings when first accessing Social Accounts with Surveys tab like this:



Message will be shown no more as soon as you add three mentioned Remote Site Settings.

Install and setup the managed package

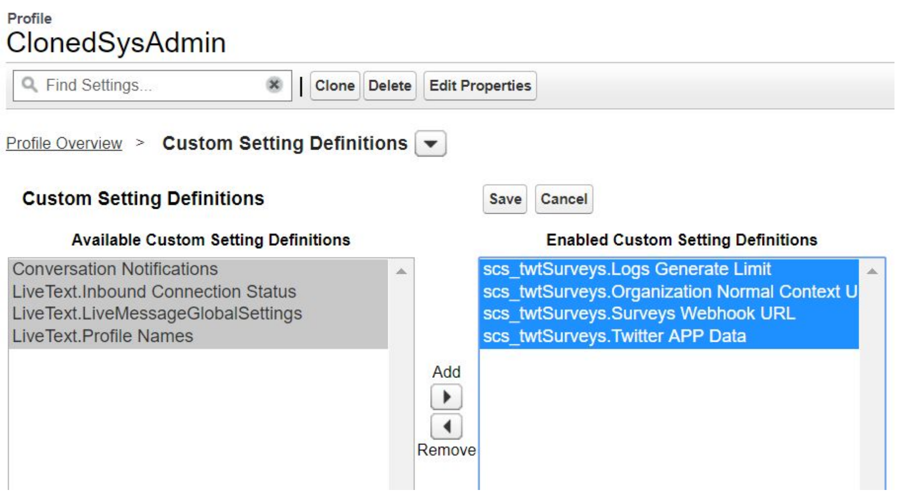
1. Open the AppExchange installation URL:
https://login.salesforce.com/packaging/installPackage.apexp?p0=04t1C000000gouq

   Note: If you are installing the app on a Sandbox, change the first segment of the URL from `login` to `test`.

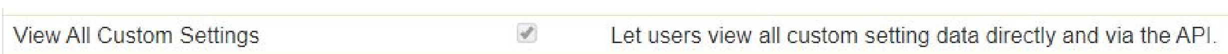
2. During installation, if a dialog box asks you to trust the Twitter API, the Facebook API server and website, or the Heroku app check **Yes, grant access...** and then click **Continue**.
3. After installation has been completed, you need to tell the app what your organization's unique Force.com Site URL is so Twitter or Facebook and Salesforce know exactly where to send survey data.
    a. From Setup, type *custom settings* into the `Quick Find` box in the Setup Sidebar Menu.
    b. Navigate to the Custom Settings page (it's under the Develop section in *Salesforce Classic* or under Custom Code in *Salesforce Lightning*).
    c. Next to the `Twitter Surveys Webhook URL` item, click **Manage** link even if you are setting up surveys for a Facebook page.
    d. Click **New** (you can do it at org level **Location** or specifically for a **Profile** or **User**).
    e. Enter the URL of your Force.com Site. It may look like `https://<your domain name>.secure.force.com/surveys.`
    f. Click **Save**.
4. Now you need to give the app permission to execute the code that communicates with Twitter or Facebook.
    a. On the Sites page, which lists all of your organization's Force.com Sites, click the Site Label to open its detail page.
    b. Click **Public Access Settings** at the top of the page to edit the profile.
    c. On the Profile page, click **Apex Class Access**.
    d. Click **Edit**.
    e. From the list of available Apex classes, add ***`scs_twtSurveys.TwitterSurveyController`*** to enabled, even if you are setting up surveys for a Facebook page.
    f. Click **Save**.
5. Give Site Guest User permission to the Document object.
    a. On the Sites page, which lists your organization's Force.com Sites, click the site label to open its detail page.
    b. To edit the profile, click **Public Access Settings** at the top of the page.
    c. Select Read and Create for the Document object.
    d. Click **Save**.
6. Identify which of your managed social accounts from the Social Customer Service feature you want to send surveys from.
    a. From Setup, go to **Manage Users** > **Permission Sets**.
    b. Create a new permission set and name it something like *`Social Handles for Surveys`*.
    c. On the permission set overview page, click **Assigned Social Accounts**.
    d. Click **Edit**.
    e. Use the arrow button to add the correct social accounts to the `Enabled Social Accounts` box.
    f. Click **Save**.
    g. Click **Manage Assignments** to select the users who should be allowed to send surveys.
    h. Click **Assign**.
    i. Back on the permission set overview page, click **App Permissions**.
    j. Click **Edit**.
    k. Enable the **Public Twitter Response** permission, even if you are setting up surveys for a Facebook page.

l. Click **Save**.

7. As an extra Step if you have **Restrict access to custom settings Enabled** at **Schema Settings** level you need to add all package custom settings to the profiles you want to use for sending out surveys following next:
    a. Go to Profile
    b. **Custom Setting Definitions**
    c. Select all the ones with **scs_twtsurveys** prefix and pass therm to **Enabled Custom Setting Definitions**



    d. Go to System Permissions for that profile.
    e. Edit and **MARK View All Custom Settings** option.



Note: A packaged permission set is included with the application: **Customer Feedback Surveys App.** We keep this permission set updated with latest changes (new fields, etc). ***Assign it to all the users you consider necessary.***

Clone mentioned permission and add Read, Create and Edit permissions for Case Object. Assign this cloned permission to all users you consider necessary.

8. Since the surveys are related to cases, you need to add the survey fields to the case page layout. We recommend using the page layout editor to add the following:
    a. On the page layout you already use for Twitter and Facebook cases, add a section called Social Feedback Survey.
    Tip: We have included a sample page layout in the AppExchange package.
    b. Add these fields in this order to the first column of this section.
        - scs_twtSurveys__Feedback_Status__c
        - scs_twtSurveys__Survey_Status_Message__c
        - scs_twtSurveys__Completed__c
        - scs_twtSurveys__Feedback_Type__c
        - scs_twtSurveys__Feedback_Score_Value__c
        - scs_twtSurveys__Feedback_Score__c
    c. Add these fields in this order to the second column of this section.
        - scs_twtSurveys__Feedback_Submitted_Date__c
        - scs_twtSurveys__Feedback_Completed__c
        - scs_twtSurveys__Auto_Survey_Job_Id__c
        - scs_twtSurveys__Feedback_Score_Text__c
9. The AppExchange package also includes the quick action **Request Feedback** so you can send surveys one at a time manually. To allow your support agents to send surveys manually from the case feed:
    a. Edit the case page layout you currently use for Twitter or Facebook cases.
    b. In the page layout editor, drag the **Request Feedback** quick action to the area for Quick Actions in the Salesforce Classic Publisher.
    You probably want to put it right after the Social quick action, where agents compose and send Twitter and Facebook responses. We'll discuss the **Request Feedback** quick action later in the *Manual Surveys* section.
    c. Click **Save**.

10. Now we're done setting up things in the main Setup area of Salesforce. Let's proceed with the necessary settings found in the Customer Feedback app itself.
    Important: You must be logged in as a user with an Admin Role in Facebook to validate the Facebook page.

    a. Open the Social Accounts with Surveys tab, which should show the Twitter or Facebook handles you enabled in the permission set you just made.

    Note: If you have LiveMessage managed package installed and you already have a Facebook page and (or) SMS number already enabled they will also appear here.

b. Click **Validate** to connect that Twitter account or Facebook page with survey functionality.
c. Click **Accept** to authorize the connection. You may be asked to enter the username and password for the Twitter or Facebook Account.

If the validation succeeds, congratulations! You now are ready to start sending customer feedback surveys to your Twitter and Facebook users (Facebook and SMS users for LiveMessage).

11. To match the Outbound DM sent with each survey request to the case of original social post and to thread the "Thanks message" to the case of the original social post (for Facebook part) you'll need to modify the method:

```
global Social.InboundSocialPostResult handleInboundSocialPost(SocialPost post,
SocialPersona persona, Map<String, Object> rawData) {...}
```

to the Apex Class you are using as your inbound social post handler.

Here is the default one with the mentioned code modifications:

```
global Social.InboundSocialPostResult handleInboundSocialPost(SocialPost post,
SocialPersona persona, Map<String, Object> rawData) {
	Social.InboundSocialPostResult result = new
Social.InboundSocialPostResult();
	result.setSuccess(true);
	matchPost(post);
	matchPersona(persona);

	if ((post.Content != null) && (post.Content.length() >
CONTENT_MAX_LENGTH)) {
		post.Content = post.Content.abbreviate(CONTENT_MAX_LENGTH);
	}

	if (post.Id != null) {
		handleExistingPost(post, persona);
		return result;
	}

	setReplyTo(post, persona);
	buildPersona(persona);

	// Custom code for TwitterSurvey Inbound DMs and Facebook Inbound
messages
	Case parentCase = null;
	if ((post.Provider == 'Twitter' && post.MessageType == 'Direct') ||
(post.Provider == 'Facebook' && post.MessageType == 'Private' &&
post.IsOutbound == false)) {
		if (post.Provider == 'Twitter') {
		Map<String, String> mapExternalIdsParentsIds = new Map<String,
```

```
String>();
              for(scs_twtSurveys__Social_Feedback_Survey__c sfS : [SELECT Id,
scs_twtSurveys__Case__c, scs_twtSurveys__Direct_Message_ID__c FROM
scs_twtSurveys__Social_Feedback_Survey__c WHERE
scs_twtSurveys__Direct_Message_ID__c = :post.ExternalPostId LIMIT 1]) {
                     post.ParentId = sfS.scs_twtSurveys__Case__c;
                     post.IsOutbound = true;
                     parentCase = [SELECT Id FROM Case WHERE Id =
:post.ParentId LIMIT 1].get(0);
              }
              } else {
              //If the post is from Facebook I need to check if is a Thanks!
message..
              //so I can link it to the corresponding case of the replyTo Post
              ExternalSocialAccount[] esas = [SELECT Id FROM
ExternalSocialAccount WHERE ProviderUserId = :post.Recipient AND IsActive =
True AND IsAuthenticated = True AND Provider = 'Facebook' LIMIT 1];
              if (!esas.isEmpty()) {
                     ExternalSocialAccount esa = esas[0];
                     scs_twtSurveys.FcbkCredentialManager fcM =
scs_twtSurveys.FcbkCredentialManager.getInstance();

                     if (fcM.existsCredentials(esa.Id)) {
                            // Then this is a message from a managed account
                            // Search for the parentId in the replyTo of this
post
                            for (SocialPost sp : [SELECT Id, Language, ParentId
FROM SocialPost WHERE Id = :post.ReplyToId LIMIT 1]) {
                            // Check if the message is indeed a Thanks type
                            List<scs_twtSurveys__Auto_Survey_Data__c>
autoSurveysData =
                            [SELECT Id,
scs_twtSurveys__Facebook_Survey_Thanks_Message__c,

scs_twtSurveys__Facebook_Survey_Thanks_Translations__c,
                            scs_twtSurveys__Default_Language__c
                            FROM scs_twtSurveys__Auto_Survey_Data__c
                            WHERE scs_twtSurveys__Object_Id__c = :esa.Id LIMIT
1];

                            if (!autoSurveysData.isEmpty()) {
                                   scs_twtSurveys__Auto_Survey_Data__c
autoSurveyData = autoSurveysData[0];

                                   // Here you can hardcode the thanks message
- Using default now
                                   String surveyThanksMsg = 'Thanks for rating
our service!';
```

```
                                // Need to find the language used for the
survey created
                                String language = null;
                                Set<String> objectFields =
Schema.SObjectType.Case.fields.getMap().keySet();
                                // First get namespace prefix
                                String np = [SELECT NamespacePrefix FROM
Organization].NamespacePrefix;
                                String languageField = !String.IsEmpty(np) ?
np + '__Language__c' : 'Language__c';
                                languageField = languageField.toLowerCase();

                                if (!objectFields.contains(languageField) &&
!String.IsEmpty(np)) {
                                // In this situation I need to check
directly the language field from the //package, no prefix
                                languageField =
'scs_twtSurveys__language__c';
                                }
                                if (objectFields.contains(languageField) &&
sp.ParentId != null) {
                                String soqlQuery = 'SELECT ' + languageField
+ ' FROM Case WHERE Id = \'' + sp.ParentId + '\' LIMIT 1';
                                List<Case> cases = (List<Case>)
Database.query(soqlQuery);
                                language = (String)
cases[0].get(languageField);
                                }
                                if (String.isEmpty(language) || language ==
'None' || language == 'Unsupported') {
                                if
(String.IsEmpty(autoSurveyData.scs_twtSurveys__Default_Language__c) ||

autoSurveyData.scs_twtSurveys__Default_Language__c == 'None') {
                                      // Just use the lastSocialPost
language
                                      language = sp.Language;
                                } else {
                                      language =
autoSurveyData.scs_twtSurveys__Default_Language__c;
                                }
                                }
                                // If we cannot find a language from the
case or social post, then the default is English
                                if
(!scs_twtSurveys.TwitterUtils.mapLanguagesToLocale.containsKey(language)) {
                                language = 'English';
                                }
```

```
                                surveyThanksMsg =
autoSurveyData.scs_twtSurveys__Facebook_Survey_Thanks_Message__c;
                                if
(autoSurveyData.scs_twtSurveys__Facebook_Survey_Thanks_Translations__c !=
null) {
                                String customTrans =
autoSurveyData.scs_twtSurveys__Facebook_Survey_Thanks_Translations__c.replaceA
ll('\r\n','\n');
                                for (String langValue :
customTrans.split('\n')) {
                                        String lang =
langValue.split(':').get(0);
                                        if (lang == language) {
                                                surveyThanksMsg =
langValue.split(':').get(1);
                                                break;
                                        }
                                }
                                }

                                if (String.isEmpty(surveyThanksMsg)) {
                                // Then I need to check the default
thanksMsg translation
                                surveyThanksMsg =
scs_twtSurveys.TwitterUtils.mapTranslationsThanksMsg.get(scs_twtSurveys.Twitte
rUtils.mapLanguagesToLocale.get(language));
                                }

                                system.debug('Thanks msg: ' +
surveyThanksMsg);
                                String postContent = post.Content;
                                system.debug('Post content: ' +
postContent);

                                if (language == 'Arabic') {
                                // If the language is Arabic, I need to
ignore the exclamation, question signs
                                // The post content populated by SCS changes
the order has problems detecting Arabic properly
                                surveyThanksMsg =
surveyThanksMsg.replaceAll('(!|\\?|؟)+', '').trim();
                                postContent =
postContent.replaceAll('(!|\\?|؟)+', '').trim();
                                }

                                if (postContent == surveyThanksMsg) {
                                // This message is a Thanks Msg generated
for the case of the ReplyTo socialPost
                                post.ParentId = sp.ParentId;
```

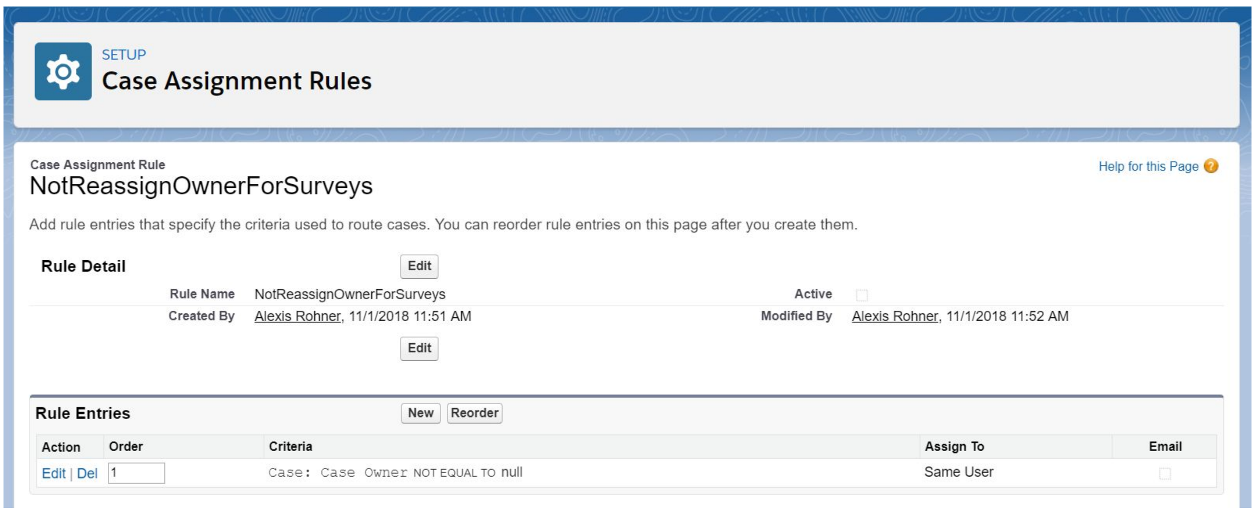
```
                                  post.IsOutbound = true;
                                  parentCase = [SELECT Id FROM Case WHERE Id =
:post.ParentId LIMIT 1].get(0);
                                  }
                           }
                           }
                    }
             }
             }
      }
      if (parentCase == null) {
             parentCase = buildParentCase(post, persona, rawData);
      }
      // ---------------------------------------------
      //Case parentCase = buildParentCase(post, persona, rawData);
      setRelationshipsOnPost(post, persona, parentCase);
      setModeration(post, rawData);

      upsert post;

      if(isNewCaseCreated) {
             updateCaseSource(post, parentCase);
      }

      return result;
      }
```

Important - When using Case Assignment Rules

If you have Case Assignment Rules in place in your environment you MUST follow next steps in order to avoid case owner reassignments.

1- Create a new Case Assignment Rule with this name **NotReassignOwnerForSurveys** (it MUST be exactly this name).

2- Add an entry like in the screenshot and make sure to mark the checkbox 

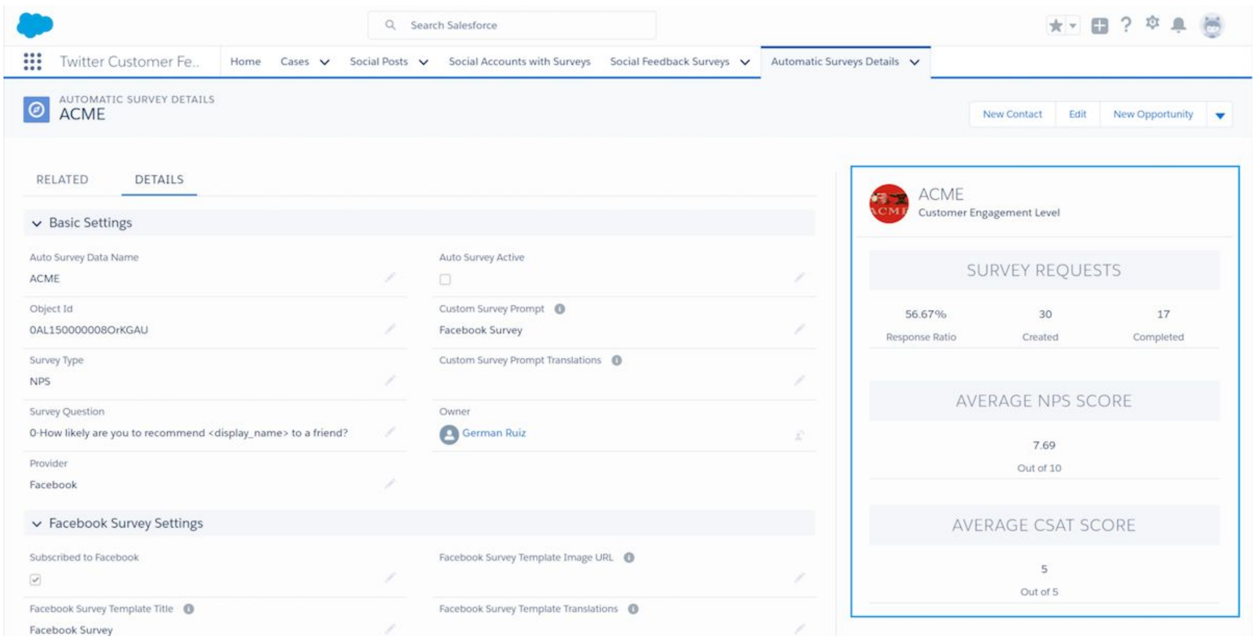


3- And Save it. **Notice you don't need to have this new assignment rule ACTIVE** (so you can continue using your assignment rule normally).

Lightning Components

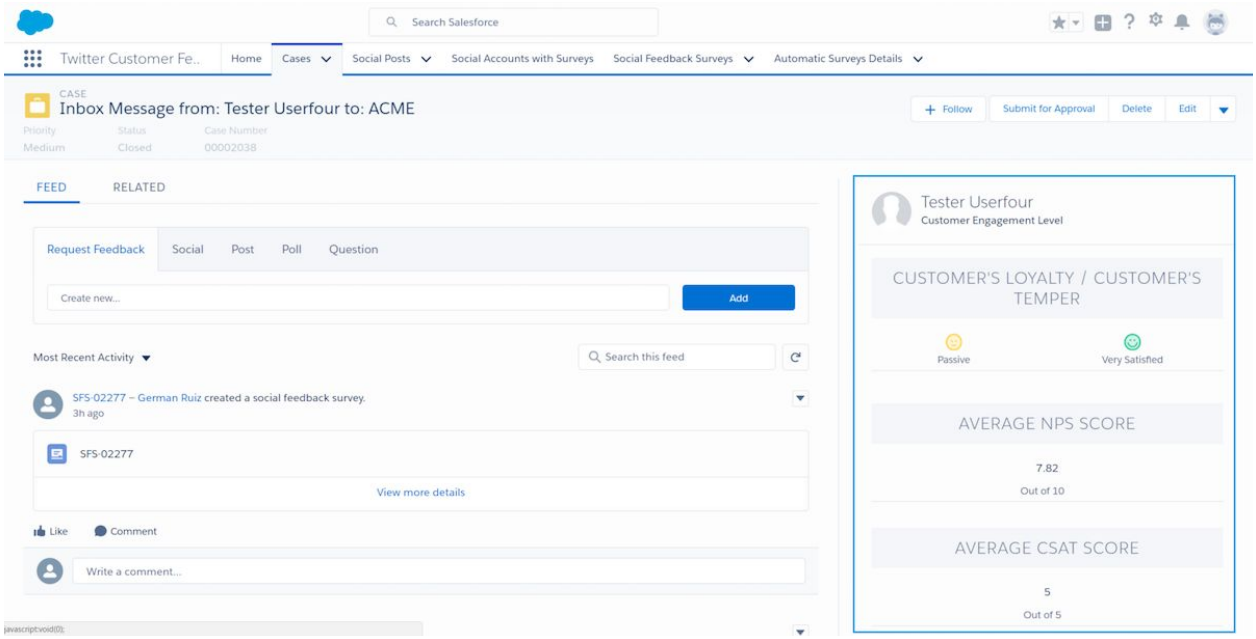
Two Lightning components are included in the package.

1. One called **`customerengagement`**, which can be added to the Automatic Surveys Detail page layout and shows information related with Social handle metrics (Surveys sent, completed, response ratio, etc).



2. Another called **`engagementlevel`**, which can be added to the Case page layout and shows information related to end user metrics for Loyalty (NPS survey type responses) and Temper (CSAT

survey type responses).



WORK WITH THE CUSTOMER FEEDBACK SURVEYS APP

Learn the concepts and procedures needed to work with the Customer Feedback App.

Automatic Surveys

EDITIONS

Available in:

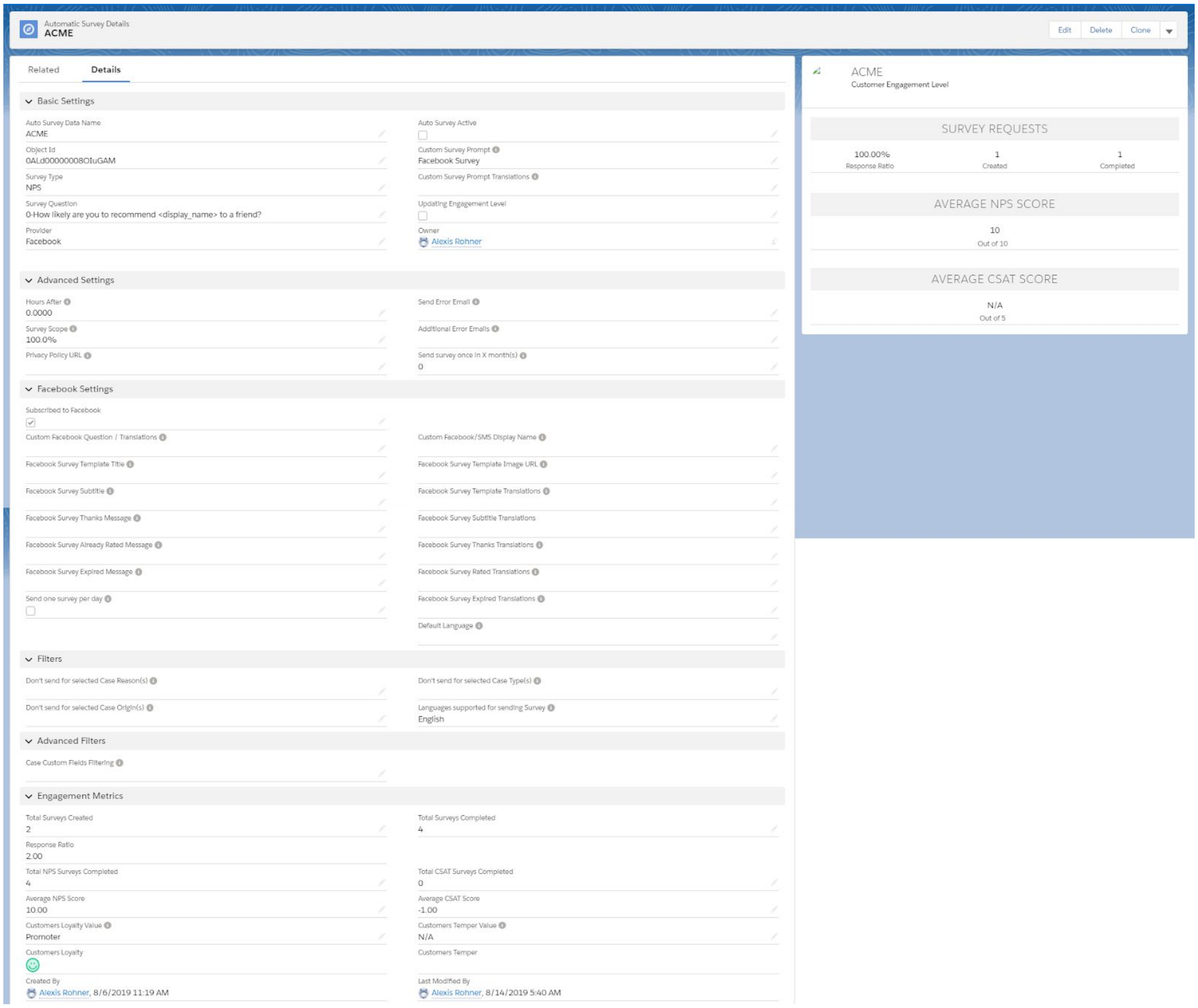
- Salesforce Classic
- Enterprise
- Performance
- Unlimited

The Customer Feedback app allows you to send survey cards automatically after a case's status has been set to closed. The Auto Survey Job ID and Status fields in the Social Feedback Survey section of the case record detail page reflect when an automatic job has been kicked off to send a survey card for a given case. For an automatic job to exist, it has to be activated on the Social Accounts with Surveys tab that is included in the app.

1. To configure the details of an automatic survey card, open the Automatic Surveys Details tab that is included in the Customer Feedback Surveys app.
2. In the `Survey Type` picklist, choose either **CSAT** or **NPS**
3. In the `Question Id-Text` picklist, choose the question you want to ask.
4. In the `Custom Survey Prompt` field, enter the text of a brief survey prompt that displays above the survey question.
   For example: Please take a moment to give us some feedback on your experience.
5. In the `Custom Survey Prompt Translations` field, enter the translated text of `Custom Survey Prompt` to send the Survey Prompt in the Twitter or Facebook user profile language. It

must be formatted like this:

The Automatic survey detail page has five sections: Basic Settings, Facebook Survey Settings, Advanced Settings, Filters and Advanced Filters.

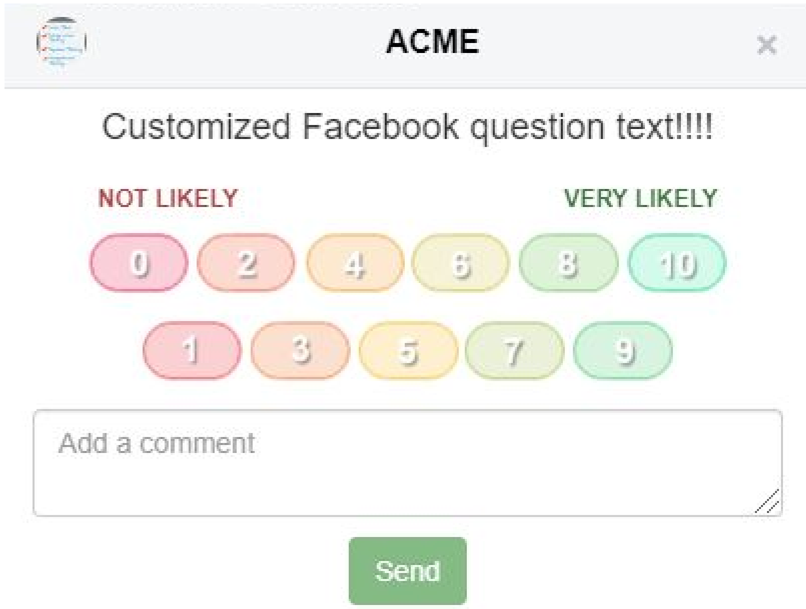


In **Facebook Survey Settings**:

- **Custom Facebook Question / Translations**: Populate with customized Facebook Question and translations for each supported language following this format:

If you populate it like this:

It will be shown like this to end user in Facebook:



- **Custom Facebook/SMS Display Name**: Populate with customized Display Name. It will be shown like this: Question text “customized display name”?
- **Facebook Survey Template URL**: URL where the image you want to show in Survey Prompt Bubble is hosted.
- **Facebook Survey Template Title**: Text you want to show as Title for the Survey Prompt Bubble.
- **Facebook Survey Template Translations**: Fill in the translations for the Facebook Survey Template Title following this format:

```
Language1:TranslatedTextMessage1
Language2:TranslatedTextMessage2
.
.
.
LanguageX:TranslatedTextMessageX
```

- **Facebook Survey Subtitle:** Text you want to show as Subtitle for the Survey Prompt Bubble.
- **Facebook Survey Subtitle Translations:** Fill in the translations for the Facebook Survey Subtitle following this format:

```
Language1:TranslatedTextMessage1
Language2:TranslatedTextMessage2
.
.
.
LanguageX:TranslatedTextMessageX
```

This is how **Facebook Survey Template Title** and **Facebook Survey Subtitle** is shown to end user in Facebook side:

- **Facebook Survey Thanks Message**: Custom text shown to a user who completed the Facebook Survey.
- **Facebook Survey Thanks Message Translations**: Fill in the translations for the Facebook Survey Thanks Message following this format:

```
Language1:TranslatedTextMessage1
Language2:TranslatedTextMessage2
.
.
.
LanguageX:TranslatedTextMessageX
```

- **Facebook Survey Already Rated Message**: Custom text shown to a user who completed Facebook Survey.
- **Facebook Survey Rated Translations**: Fill in the translations for the Facebook Survey Already Rated Message following this format:

```
Language1:TranslatedTextMessage1
Language2:TranslatedTextMessage2
.
.
.
LanguageX:TranslatedTextMessageX
```

- **Facebook Survey Expired Message**: Custom text shown to a user to inform them that Facebook Survey already expired.
- **Facebook Survey Expired Translations**: Fill in the translations for the Facebook Survey Expired Message following this format:

```
Language1:TranslatedTextMessage1
Language2:TranslatedTextMessage2
.
.
.
LanguageX:TranslatedTextMessageX
```

- **Send one survey per day**: Uncheck this checkbox if you want to send more than one survey to the same Facebook User within a 24 hours period (Checked by default).

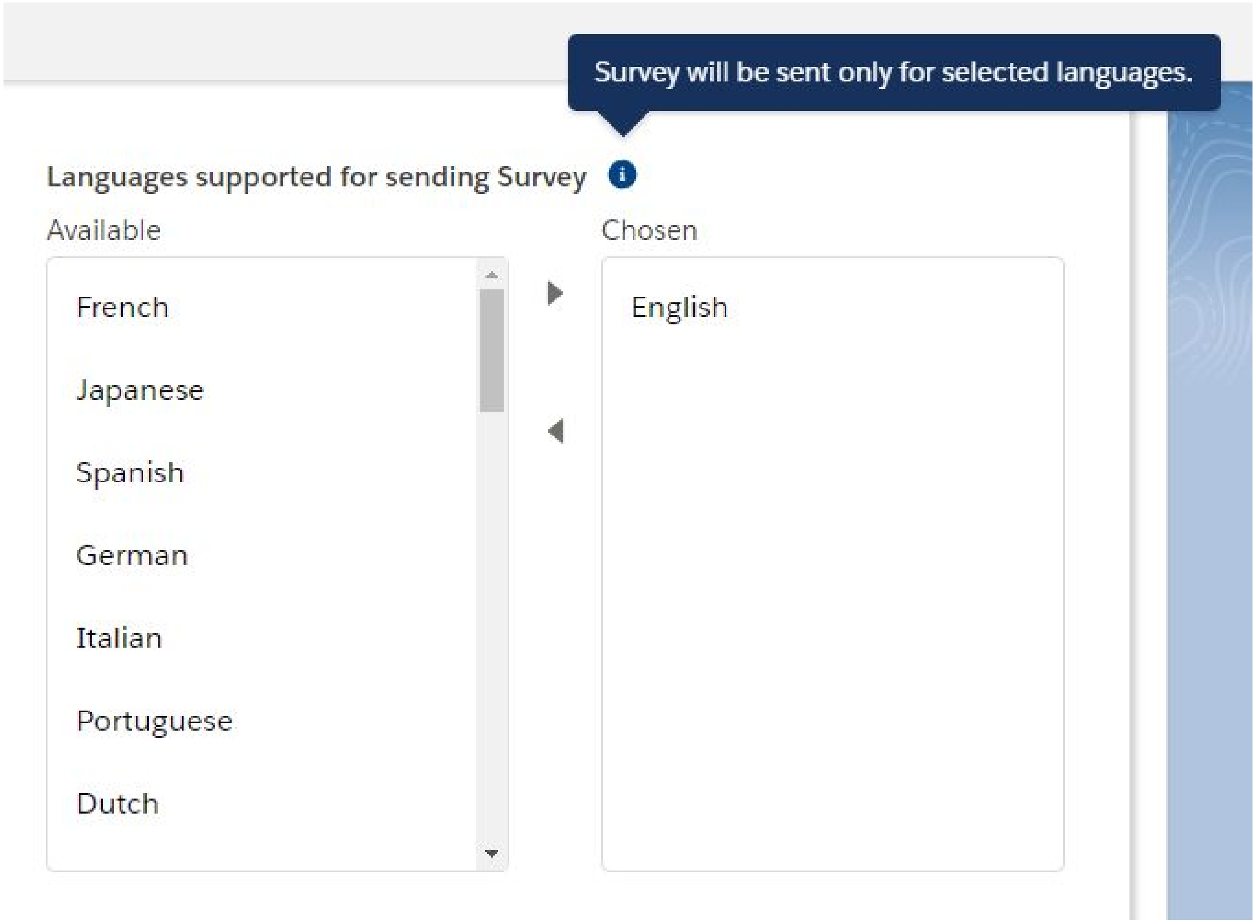
In **Advanced Settings:**

- **Hours After**: The wait time in hours before sending the automatic survey. Enter an integer from 0 to 99. Zero (0) means the survey will be sent immediately upon case closure.
- **Survey Scope**: The survey will be sent to X % of the closed cases randomly.
- **Privacy Policy URL**: Required. The URL of the sender's hosted privacy policy. Follow this format: `https://www.yourprivacypolicyURL.com`. Note that you must use “https”.
- **Send Error Email**: The specified email gets an email when error occurs.
- **Additional Error Emails**: The specified, comma separated, emails also get an email when an error happens.
- **Send survey once in X month(s)**: Send the survey only if X month(s) has passed since the last time a survey was sent to a specific
@user.

In **Filters**:

- **Don't send for selected Case Reason(s)**: Survey will not be sent from cases with selected Reason(s). Multi-select type field.
- **Don't send for selected Case Origin(s)**: Survey will not be sent from cases with selected Origin(s). Multi-select type field.
- **Don't send for selected Case Type(s)**: Survey will not be sent from cases with selected Type(s). Multi-select type field.

- **Languages supported for sending Survey**: Send the surveys in the user's profile language. Move the supported languages to the Chosen list.



In **Advanced Filters**:

- **Case Custom Fields Filtering**: Use this field to identify when you don't want to send a survey request. Enter the fields and values that would constitute not sending a survey in this format:
  `Field_API_Name:Value1;Value2;...`
  `Field2_API_NAME:Value1;Value2;...`

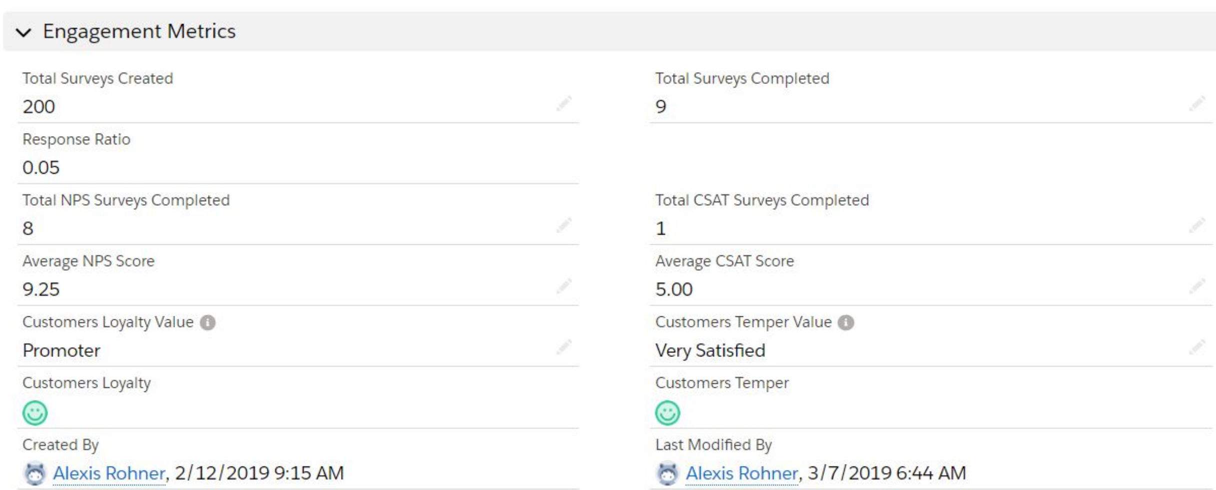
Supported field types and format for filtering:

Field type	Supported	Format for using in filter
Autonumber	**YES**	**1234**
Checkbox	**YES**	**true / false**
Currency	**YES**	**$12 -> 12**
Date	**YES**	**YYYY-mm-dd 00:00:00 -> 2019-05-03 00:00:00**
Datetime	**YES**	**YYYY-mm-dd HH:mm:ss where HH:mm:ss MUST be converted to GMT**
Email	**YES**	**something@something.com**
Geolocation	**NO**	**NO**
Lookup	**YES**	**Populate ObjectID as value**
Multipicklist	**NO**	**NO**
Number	**YES**	**1234**
Percent	**YES**	**99% -> 99**

Picklist	**YES**	**PicklistValue**
Phone	**YES**	**(555) 555-5555**
Text	**YES**	**Text**
Text Area	**YES**	**Text**
Text Area Encrypted	**YES**	*********8910 -> 12345678910**
Text Area Long	**YES**	**Text**
Text Area Rich	**YES**	**Text**
Time	**YES**	**HH:mm:ss.SSSZ -> 12:00:00.000Z, desired time MUST be converted to GMT**
URL	**YES**	**https://www.salesforce.com -> www.salesforce.com**

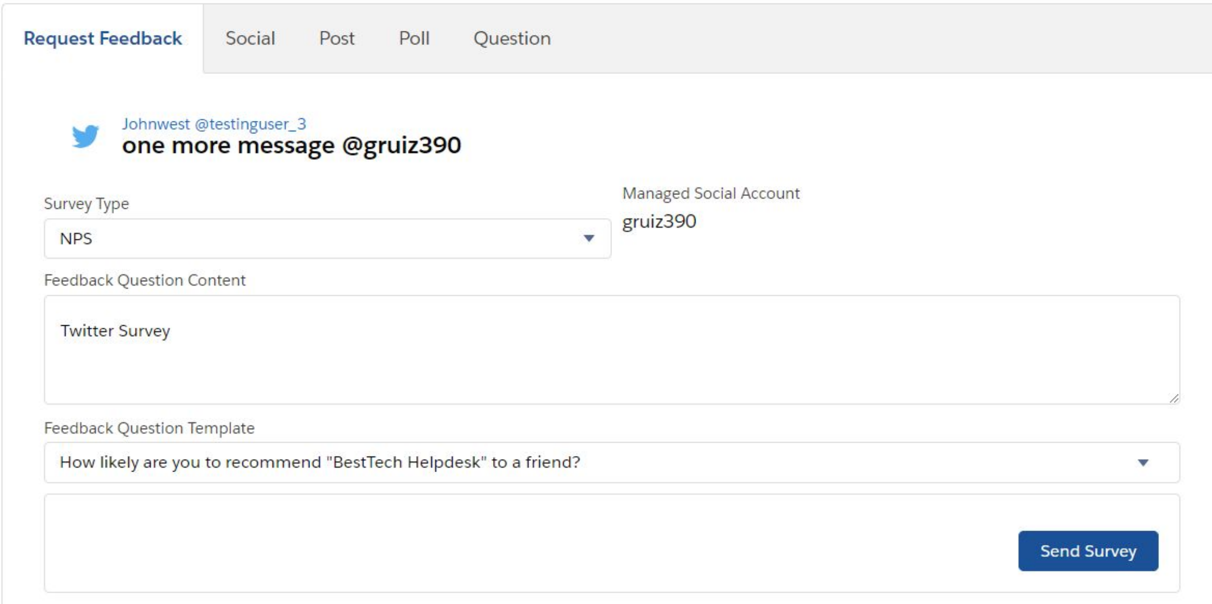
Note:

We have also added a section at the end of Automatic Survey detail page layout for you to have main metrics for each individual channel, you'll be able to see **Created Surveys**, **Completed Surveys** and **Response Ratio** and also have a sense of your customer's **Loyalty** (if using NPS surveys) and/or your customer's **Temper** (if using CSAT surveys) :



Manual Surveys

Manual survey quick action has four important fields.



- **Survey Type**: Choose the type of survey to be sent when closing the case.
- **Managed Social Account**: Social Account validated for using Surveys feature.
- **Feedback Question Content**: Fill in the text you want to be rendered as a message for introducing the survey request.
- **Feedback Question Template**: Questions will be listed according to the selected Survey Type (NPS or CSAT).

Facebook Surveys Language

To determine which language the survey is sent in, there is a priority order that our app checks before sending out Facebook Feedback survey.

- If no language is specified as Default Language for the Facebook page (Automatic Survey Details) and no Case language is specified -> Then survey will be sent according to Social Post language.
- If Feedback Survey Language is specified and Default Language is specified for the Facebook page -> Then survey will be sent according to Case Language specified by agent.
- If Case Language is not specified and Default Language for Facebook page is specified -> Then survey will be sent according to Default Language.

Default Language (Automatic Survey Detail)	**Feedback Survey Language** field (scs_twtSurveys__Language__c)	**Social Post Language**	**Survey Language**
✅	✅	✅	Case Language
✅	✅	❌	Case Language
✅	❌	❌	Default Language
✅	❌	✅	Default Language
❌	✅	✅	Case Language
❌	✅	❌	Case Language
❌	❌	✅	Social Post Language
❌	✅	❌	Case Language
❌	❌	❌	English (Default)

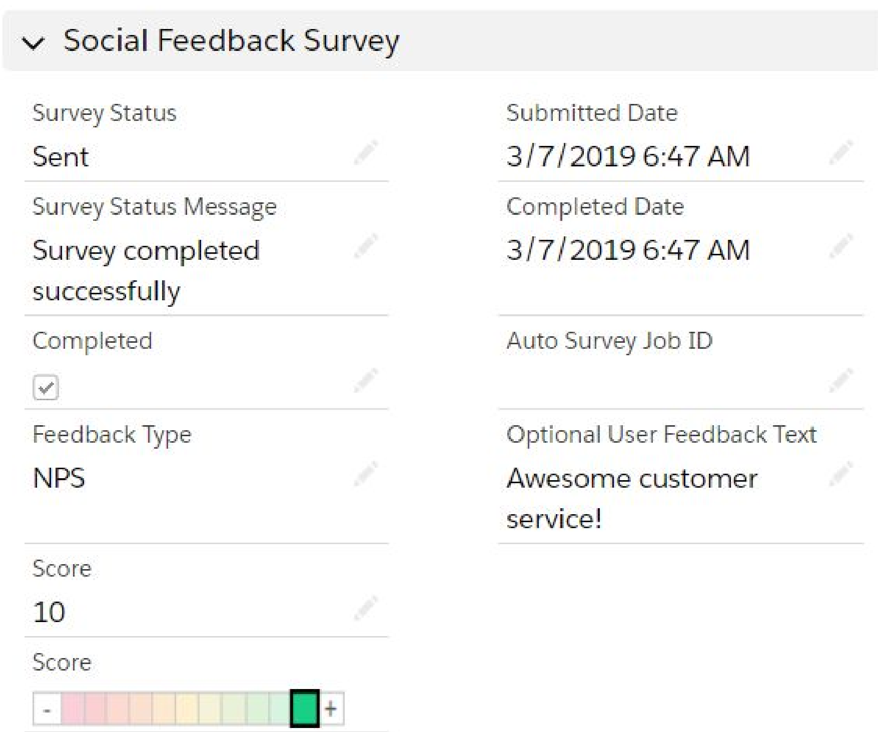
Priority Summary (from High to Low)

1. ***Case Language (using our field scs__twtSurveys__Language__c at Case level)***
2. ***Default Language field (at Automatic Survey Details level)***
3. ***Language standard field (at Social Post level)***

Survey Response Data on the Case Detail Page

The Social Feedback Survey related list provides the following information.

- **Survey Status**: Populated with Not Sent by default changes to Sent when survey request is sent to the Twitter or Facebook user or it changes to Dismissed or Expired when the survey expires.
- **Survey Status Message**: Shows error details when survey request fails.
- **Completed**: Unchecked by default changes, checked when survey response reaches Salesforce.
- **Feedback Type**: Populated with NPS/CSAT depending on the type of request sent to Twitter or Facebook user.
- **Score** (Numeric): Not populated by default, populated when response reaches Salesforce (it can be numeric or text depending on the survey type response).
- **Score** (Bar): 11 Segments / 5 segments bar greyed out by default, gets populated with value according to the survey type response.
- **Submitted Date**: The date the survey is sent.
- **Completed Date**: The date the survey is completed.
- **Auto Survey Job ID**: Populated with ID for the job when using Auto Surveys.
- **Optional User Feedback Comment**: Optional comment text provided when completing the survey.
- **Social Feedback Surveys related list**: Lists all the surveys sent from current case sorted from the newest (most recently sent) to the oldest.



Survey Status Message field possible values and its meaning

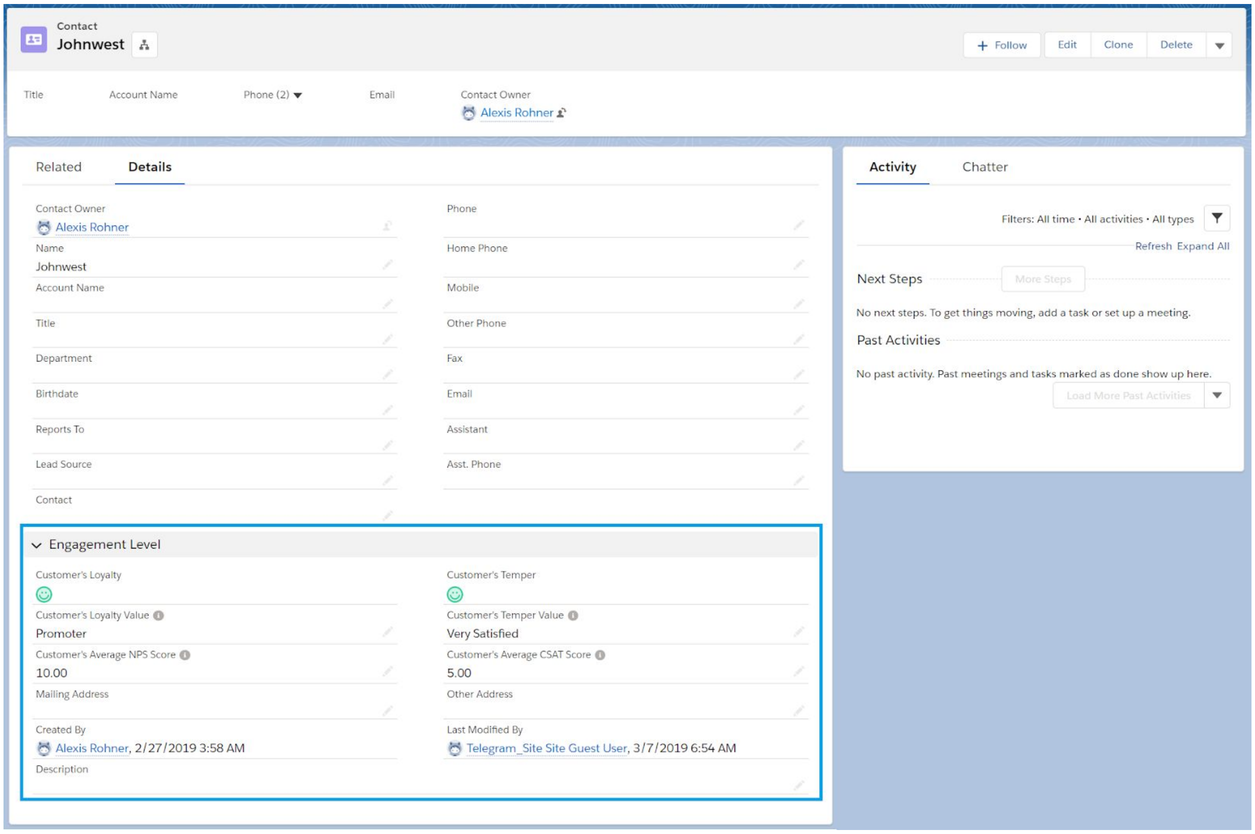
1. **24-hrs window has passed**
    - Our app is not able to send survey because more than 24hrs passed since last inbound private message from facebook user you are trying to send survey to, this is a limit on Facebook API side.
2. **FB settings: Cannot send more than 1 survey per day**
    - You have the checkbox **Send one survey per day** marked in Automatic Survey Details object related to facebook page.
3. **Recipient not available**
    - This means Facebook end user you are trying to send survey to deleted conversation or blocked your page.
4. **Error detected sending Twitter/Facebook/SMS survey**
    - This is a generic error shown most of the time when APIs are down.
5. **Failed to match against app database**
    - Our app tries to use SCS messenger id to send survey out if that fails we try to match against our heroku database (which uses Facebook webhook), if we are not able to find messenger id then we fail with this error.
6. **Filters: Language not supported**
    - Language detected in case's social posts it is not included in the supported languages list for Automatic Survey Details object related to Facebook Page/Twitter Account.
7. **Filters: Social Persona not receiving surveys**
    - This filter is applied when Social persona related to the case has **Not a Survey candidate** checkbox marked.
8. **Twitter user is not following or is blocking your account**
    - You cannot send surveys to twitter users that are not following / has blocked your account.
9. **No PM available for the user in this case**
    - You will find this message when the case does not contain any private message (just Post, comment or reply) from end user, this means we are not able to send survey for this case.
10. **Filters: X**
    - When a survey is not sent due to filters applied at Automatic Survey Details level the detailed information will be shown like this.

Fields on the Contact / Person Account Detail Page

Fields, calculated from the average of all the NPS or CSAT survey responses scores, can be added to the contact or person account detail page.

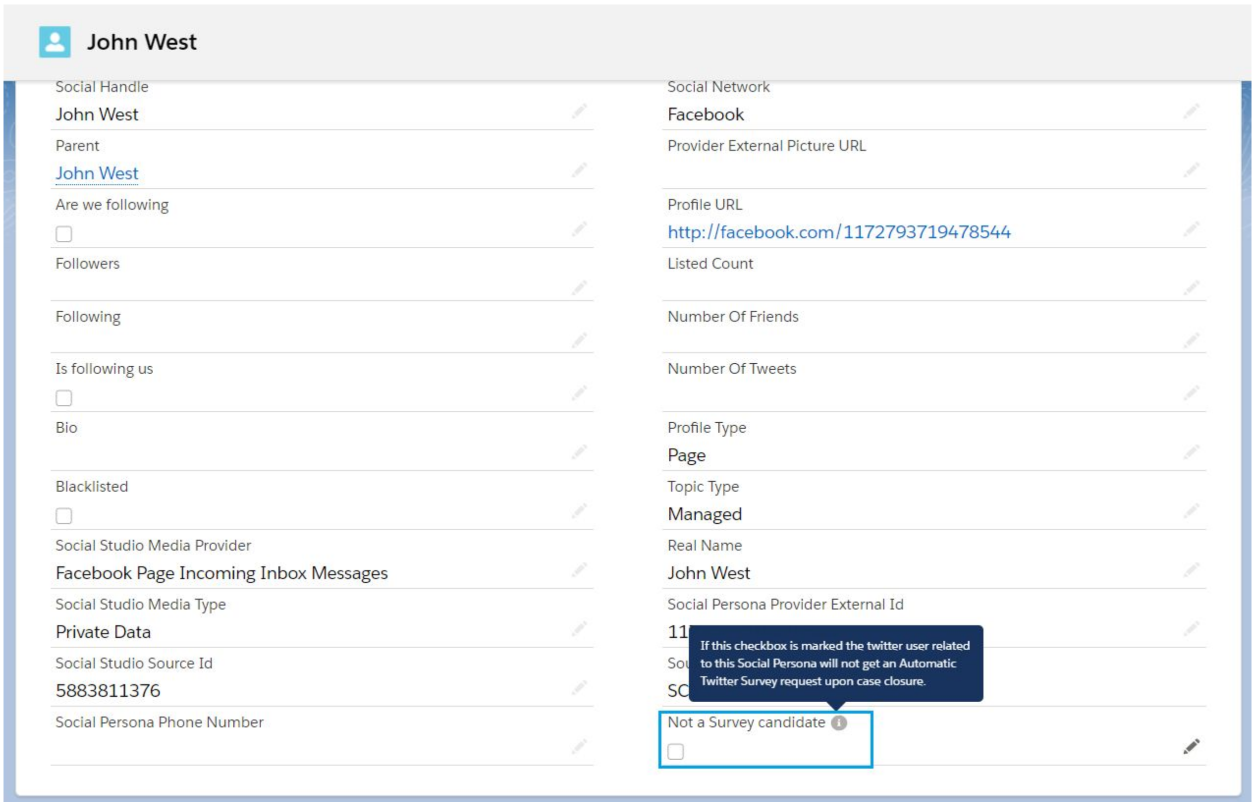
- **Customer's Loyalty Value**: Calculated from the average of the scores of the NPS surveys for the user:
  - 0-6 Detractor
  - 7-8 Passive
  - 9-10 Promoter
- **Customer's Loyalty (face)**: Populated depending on the Customer's Loyalty Value field:
  - Promoter
  - Passive
  - Detractor
- **Customer's Average NPS Score**: Calculated average from all the NPS survey scores for the current contact or person account.
- **Customer's Temper Value**: Calculated from the average of the scores of the CSAT surveys for the user:
  - 1 Very Dissatisfied
  - 2 Somewhat Dissatisfied
  - 3 Neutral
  - 4 Somewhat Satisfied
  - 5 Very Satisfied
- **Customer's Temper (face)**: Populated depending on the Customer's Temper Value field:
  - Very Dissatisfied
  - Somewhat Dissatisfied
  - Neutral
  - Somewhat Satisfied
  - Very Satisfied

- Customer's Average CSAT Score: Calculated average from all the CSAT survey scores for the current contact or person account.

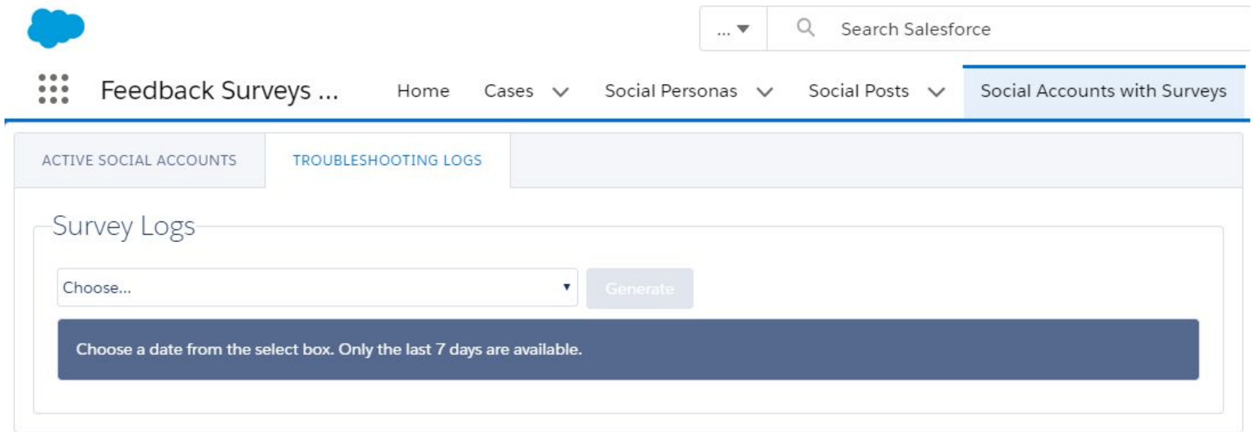


Field on the Social Persona Object

This package adds the field, **Not a Survey candidate** to the Social Persona object. If the checkbox is marked the user related to this Social Persona doesn't get an automatic survey upon case closure.



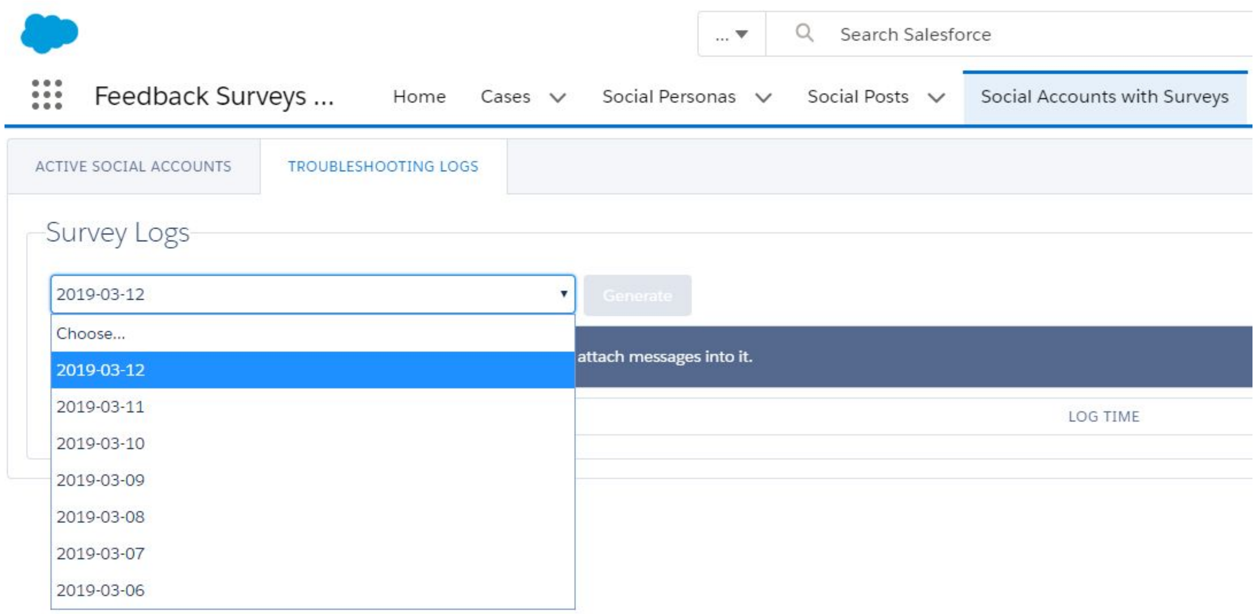
App Troubleshooting logs



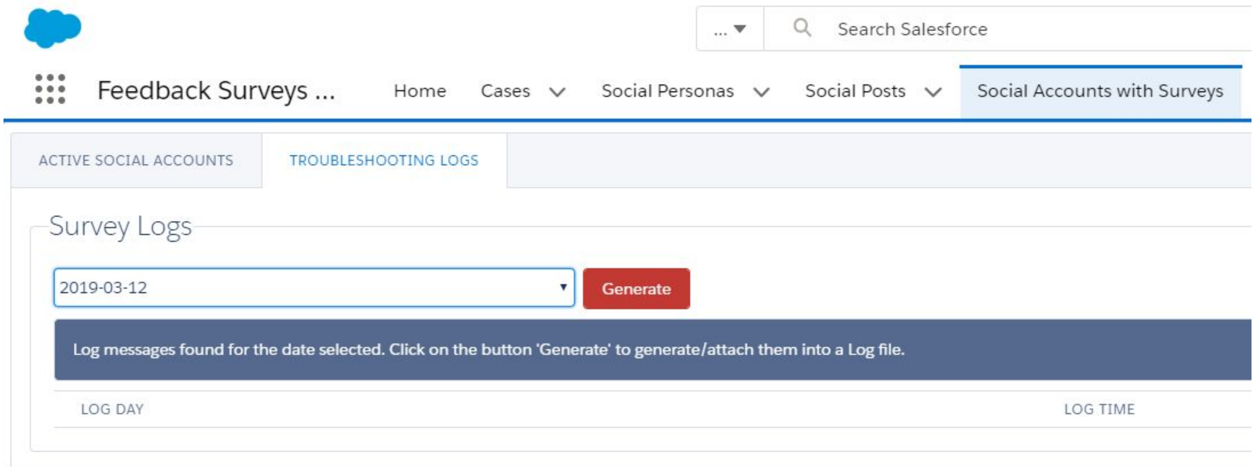
We’ve changed the way our app logs the events (Survey sending / responses)

You’ll be able to get logs for the last 7 days, just follow these steps:

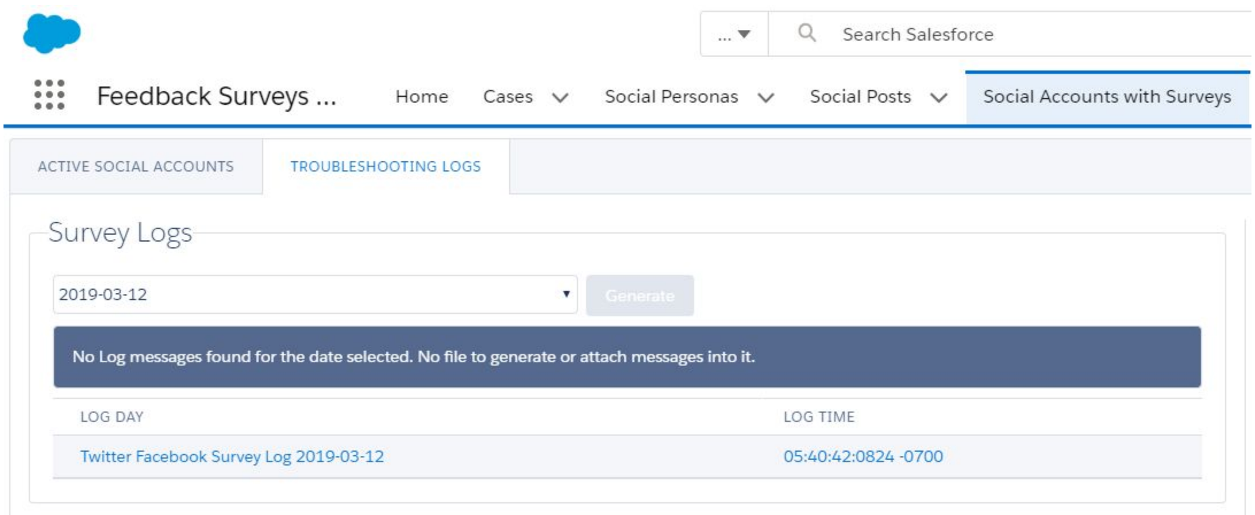
1- Choose a date from the picklist



2- If new events have occurred for selected date Generate button will become clickable



3- Click on Generate button and log file will be created (or updated with new events if present already)



4- Continue with clicking Generate button until it remains deactivated (this means no more log messages are present)

Note:

There is a Custom Setting called **Logs Generate Limit** that comes with default value 2000 our advice is to use a higher value **(*not higher than 5000 as you might hit Salesforce limits*)** if your survey volume is high.

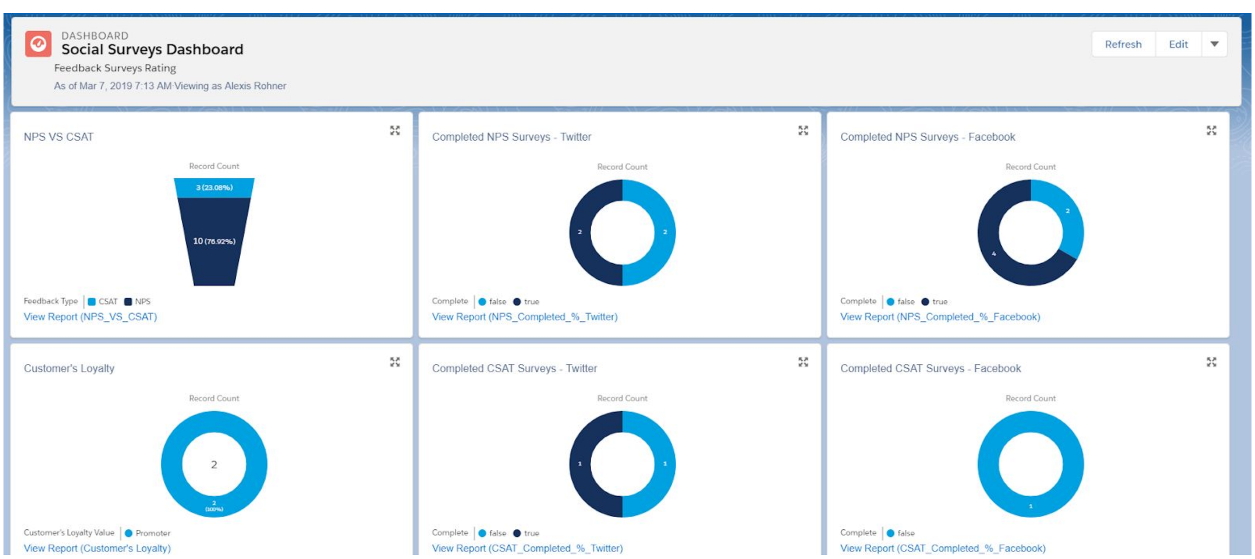
Basic Reports and Dashboards Included

Social Surveys dashboard includes the following eight reports.

- **NPS vs CSAT**: Basic report for showing the % of NPS vs the % of CSAT surveys.
- **Completed NPS Surveys – Twitter**: Basic report for showing the % of completed vs % of non-completed NPS surveys.
- **Completed NPS Surveys – Facebook**: Basic report for showing the % of completed vs % of non-completed NPS surveys.
- **Completed CSAT Surveys – Twitter**: Basic report for showing the % of completed vs % of non-completed CSAT surveys.
- **Completed CSAT Surveys – Facebook**: Basic report for showing the % of completed vs % of non-completed CSAT surveys.
- **Customer's Loyalty**: Basic report for showing the % of Promoters vs % of Passives vs % of Detractors.
- **Top Promoters**: Basic report for showing the 10 best promoters sorted by Average NPS score sorted Descending.
- **Top Detractors**: Basic report for showing the 10 worst detractors sorted by Average NPS score sorted Ascending.

Out of the ones included in dashboard we are also including following reports (related to LM channels)

- **Completed NPS Surveys - Text**: Basic report for showing the % of completed vs % of non-completed NPS surveys for SMS LiveMessage channel.
- **Completed NPS Surveys – Facebook LM**: Basic report for showing the % of completed vs % of non-completed NPS surveys for Facebook Page under LiveMessage channel.
- **Completed CSAT Surveys - Text**: Basic report for showing the % of completed vs % of non-completed CSAT surveys for SMS LiveMessage channel.
- **Completed CSAT Surveys – Facebook LM**: Basic report for showing the % of completed vs % of non-completed CSAT surveys for Facebook Page under LiveMessage channel.



How to solve...

Survey responses not pushed into Salesforce

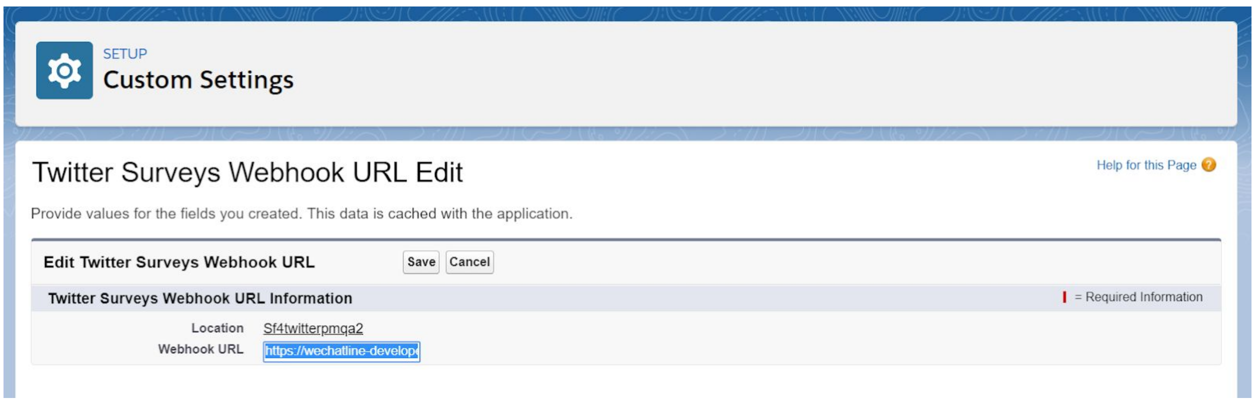
This issue it is related with having app custom setting populated wrong. There is a quick check that can be done following these steps:

1- Go to Salesforce Setup

2- Custom Settings

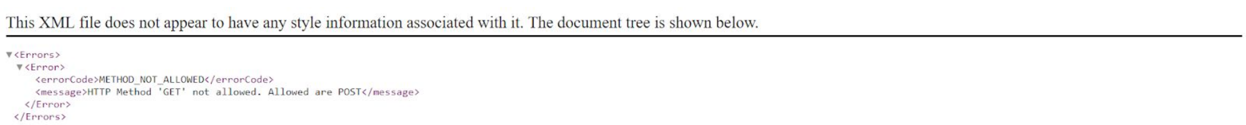
3- Click on Manage for the one called Twitter Surveys Webhook URL

4- Click on the Edit button and copy the value populated in Webhook URL



5- Paste it in browser and add: services/apexrest/scs_twtSurveys/FeedbackSurveySite

6- Hit Enter and you should get a response like this:



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<Errors>
  <Error>
    <errorCode>METHOD_NOT_ALLOWED</errorCode>
    <message>HTTP Method 'GET' not allowed. Allowed are POST</message>
  </Error>
</Errors>
```