

# Installation Guide

December 2019



## Sommaire

<b>1. About KSL for Salesforce®</b>	<b>3</b>
1.1. Foreword concerning the KSL plug-on	3
1.2. KSL for Salesforce®	3
1.3. Use cases	6
<b>2. Installing KSL Suite from Salesforce® AppExchange©</b>	<b>7</b>
2.1. Technical prerequisites when installing the KSL plug-in	7
2.2. Installing from Salesforce® AppExchange®	7
<b>3. Configuring KSL for Salesforce®</b>	<b>9</b>
3.1. User configuration	9
3.2. Configuring the communication between Salesforce® and KSL servers	11
3.3. KSL components - Access to template and documents	13
<b>4. KSL For Salesforce® for administrators</b>	<b>18</b>
4.1. Setup tab	18
4.2. Content repository tab	22
<b>5. Add the Plug-In components to new objects</b>	<b>23</b>
5.1. The object creation	23
5.2. Metadata	23
5.3. The component	24
5.4. The actions	24
5.5. The Display PDF component	26
<b>6. Validation workflow example</b>	<b>27</b>
6.1. Creation of a queue	27
6.2. The validation workflow	27
<b>7. KSL for Salesforce® for template designers</b>	<b>29</b>
7.1. Mandatory KSL functional parameters	29
7.2. Other KSL functional parameters	29
7.3. Considerations on Salesforce® data sent to KSL	29
7.4. Use of XML data in KSL	29
7.5. Create on-demand email template	30
7.6. Create emailing templates	36
7.7. Proposing an e-mail or e-mailing with alternative contents	43
<b>8. Annexes</b>	<b>45</b>
8.1. Generate public and private keys	45

## 1. About KSL for Salesforce®

### 1.1. Foreword concerning the KSL plug-on

Some features described in this document are only available from the version 1.6. These versions include email-related features:

- Create, personalize and send an e-mail
- Send a KSL document by e-mail
- Create and submit an e-mailing

KSL for Salesforce® is dedicated to the management of personalized customer communications and in particular the generation of documents, emails and emailings. In this document, the term "document" applies to all communications generated from KSL.

The explanations given in this document often take the Opportunity object as an example. However, they apply to any standard object such as Quotes or Campaigns, or any custom object.

### 1.2. KSL for Salesforce®

KSL for Salesforce® is a plug-in that enables the Salesforce® solution to communicate effectively and securely with the KSL servers, to generate, edit, personalize, archive, and/or deliver documents and emails.



This plug-in provides several key functions, which are very interesting to secure document and email generation and delivery:

- Document and e-mails generation and delivery
- Document versioning and archiving
- Document and e-mail template design based on shared components

#### a. Document and e-mails generation and delivery

- Template selection by the end-user:
  - The document or e-mail generation starts with the selection of a template in the Document and e-mail templates view

- This view is a KSL component that appears in Salesforce®
  - It is configured for different Salesforce® object (for example, opportunity and campaign)
  - It presents the list of available and authorized templates in a tree structure
  - These access permissions depend on the user's profile (KSL role)
- Personalized document generation thanks to Salesforce® data:
    - Each document template is set to be generated in one or two launch modes: Document generation in PDF (*on demand*) or/and document edition i.e. with interactive modification (*interactive*)
    - It should be noted that KSL can generate a PDF document at the end of the interactive edition step
    - This interactive edition step allows the user to view and modify the document before sending it to its customer or prospect
  - Personalized e-mail bodies or e-mailings thanks to Salesforce® data:
    - Each email template is characterized by the launch mode *mail*
    - Each email template is characterized by a return mode, which is defined by the *MailReturnMode* parameter equal to 1 for sending a single email through Salesforce or 2 for creating an emailing
    - In the case of a single email, the HTML body is personalized by the user and inserted automatically in a Salesforce® email for a manual sending
    - In the case of an emailing, the HTML body of the emails is personalized by the user and KSL processes the transmission of the emails to an external email send and track service (not provided by Naelan)

## b. Document versioning and archiving

- Personalized documents created by KSL:
  - Document storage is provided by the KSL solution: document are archived on KSL servers in a SaaS mode
  - A logical link to the document is created automatically in Salesforce®
  - A PDF document generated on demand by KSL is automatically associated with the object that created it, and its version is set to 1; the user can view the document in PDF format; he can send it to a validation workflow or send it by email
  - A document generated after interactive personalization is automatically linked with the object that created it, and its version is set to 1. The user can view it in PDF format, send it to a validation workflow or send it by email; he can also edit again the document and save it; in this latter case, the document version is incremented automatically by KSL. It should be noted that all PDF versions of the document remain accessible, but only the latest version of the document is editable
- Personalized e-mail bodies created by KSL:
  - The e-mail body of a single email generated after an interactive personalization, is not archived by KSL

- This e-mail body is automatically inserted into a Salesforce® e-mail that is sent and saved in the activities of the object in Salesforce®
- Personalized e-mailings created by KSL:
  - E-mailings are not accessible from Salesforce® after they are created and launched.
  - An task is created in the Salesforce® object for this e-mailing as soon as it is transmitted to the KSL server
  - The e-mailing is sent by the KSL server to an external email send and track service asynchronously
- Sending by e-mail a personalized PDF document created by KSL:
  - A personalized PDF document PDF can be send by e-mail
  - In this case, a Salesforce® activity is created by KSL containing the message sent: the sender, the recipients, the subject and the message body written by the user and its sending time
  - This activity is associated to the object used to create the document
  - The attachments and especially the KSL PDF document are accessible in the Attachments view
  - The documents in attachments are stored in Salesforce® and not in KSL: when sending a KSL document by e-mail, the document is archived in KSL when it is created and stored in parallel as an attachment of the e-mail in Salesforce®
- Salesforce® data allowing to personalize documents and emails:
  - These data come from different Salesforce® objects
  - They are transmitted in XML format automatically and seamlessly
  - Document and email templates are based on an XSD schema corresponding to the XML data
  - XSD schema and XML sample can be created by the Salesforce® administrator thanks to the KSL Administration application

### c. Document and e-mail template design based on shared components

- A centralized repository containing resources shared by templates
  - This repository is centralized and collaborative
  - It contains all the components necessary to document and e-mail templates: text zones, groups of text zones, images, PDF appendices and style sheets
  - It contains all the document and e-mail templates
  - Each component (resource) is shared by the templates
  - The repository management cans be done from Salesforce® thanks to KSL Administration application or thanks to KSL Studio
  - Each text content can be multi-brands, multilingual and / or multi-institution

- Component and E-mail template design via KSL Administration
  - For designing components, please refer to the KSL Office user guide
  - For designing e-mail templates, please refer to the KSL Email designer user guide
  - For designing document templates, refer to the information below
  
- Document template design thanks to KSL Studio tool:
  - KSL Studio is a Windows specific design tool you have to install for creating your templates
  - This tool also allows to manage the repository components
  - This is a collaborative tool for creating simple or complex components, rules and document templates from an XSD schema in connection with the KSL centralized repository

### 1.3. Use cases

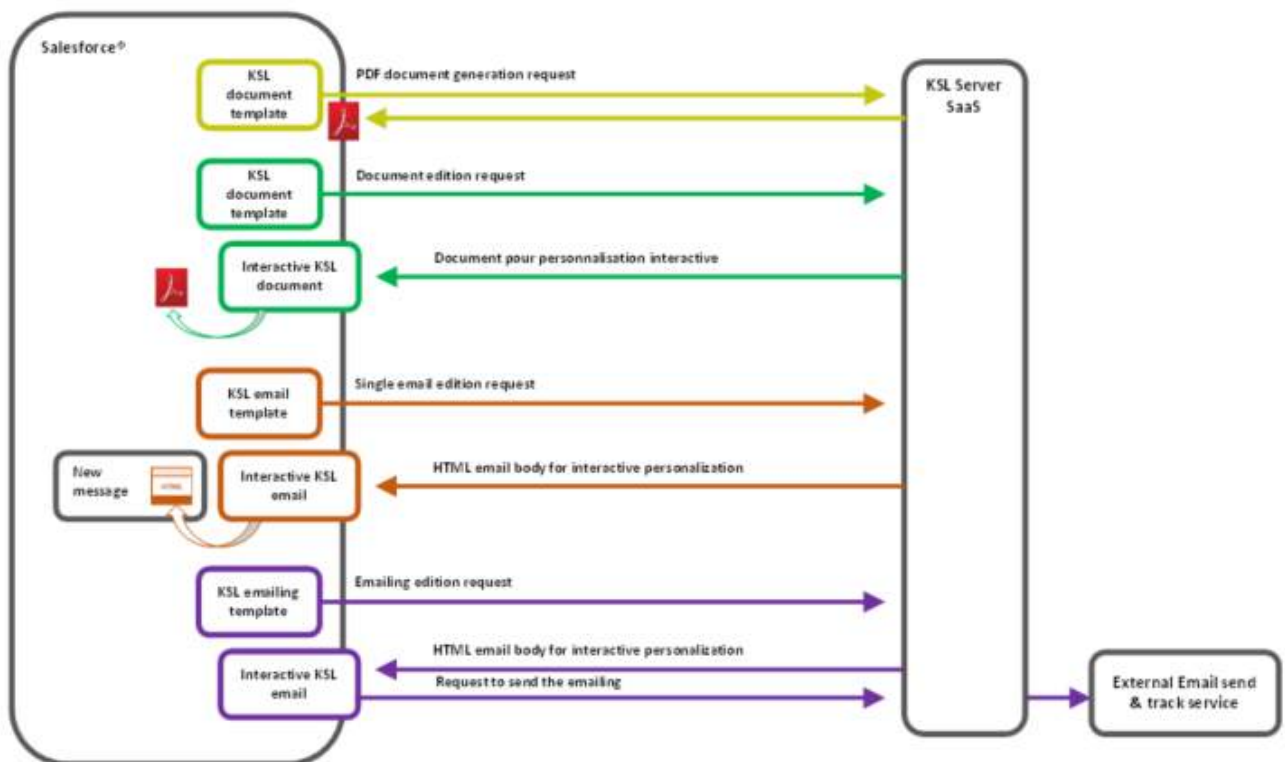


Figure above: Simplified schema of the different generation option

## 2. Installing KSL Suite from Salesforce® AppExchange®

### 2.1. Technical prerequisites when installing the KSL plug-in

To install and use KSL for Salesforce®, one of the following Salesforce edition is required:

- Enterprise Edition (EE)
- Unlimited Edition (UE)
- Developer Edition(DE)
- Performance Edition
- Professional Edition (with API access option)

Note that the Essential edition (EE) is not supported, because this version does not have API access.

Other prerequisites:

Salesforce® Products            ■ Sales Cloud

---

KSL Products                    ■ KSL for Salesforce (plug-in)  
    ■ KSL Server

---

KSL Environment                ■ Remote site URL  
    ■ Office app access URL  
    ■ KSL project name  
    ■ Public Key (oAuth2 connection)  
    ■ Private Key (oAuth2 connection)

### 2.2. Installing from Salesforce® AppExchange®

KSL plug-in is available in AppExchange®.

To install the KSL for Salesforce® plug-in:

1. Search the KSL for Salesforce plug-in in AppExchange®
2. Click on Get It Now and follow the instructions to install KSL Suite for Salesforce®
3. Specify where you want to install the KSL for Salesforce® plug-in:
  - Click Install in production to install in a production environment.
  - Click Install in sandbox to install in a Sandbox.
4. Read and accept the terms and conditions:
  - Select I have read and agree to the terms and conditions.
  - Click Confirm and Install.

As a best practice, and to avoid access issues for end users, we recommend selecting Install for Admins Only. It enables you to use the permission set delivered in the package to grant the rights to the appropriate users.

5. Then click Install:

- You must provide System Administrator credentials to install an application from AppExchange®.
- After installing, you must configure the KSL Suite application, including integration settings for third-party applications.



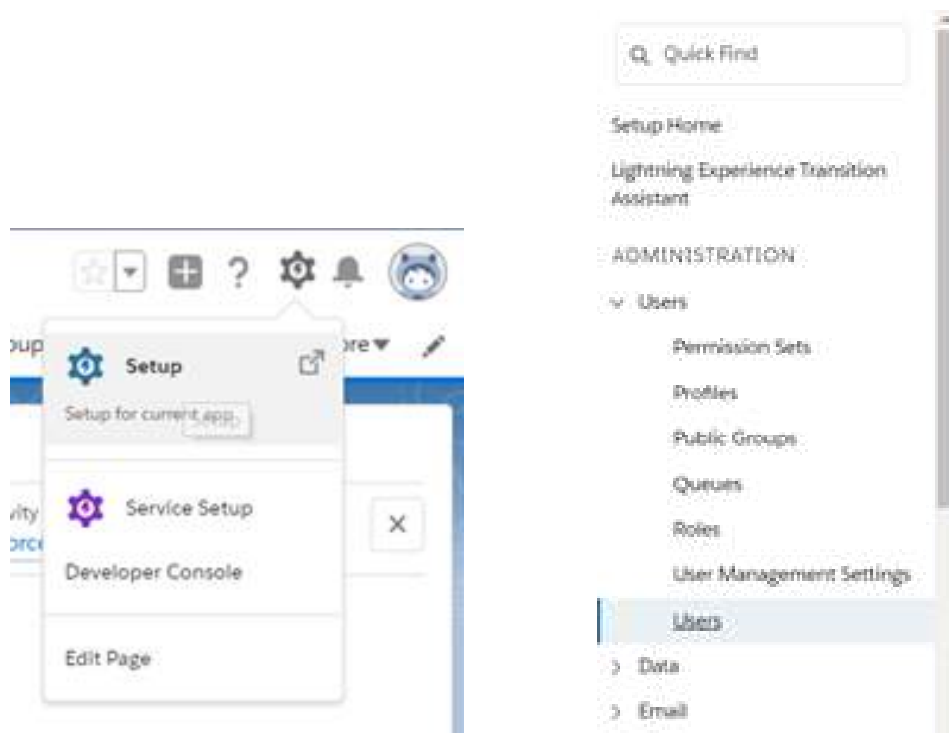
## 3. Configuring KSL for Salesforce®

### 3.1. User configuration

No user has to be created in KSL: KSL for Salesforce® relies on the users already created in Salesforce®.

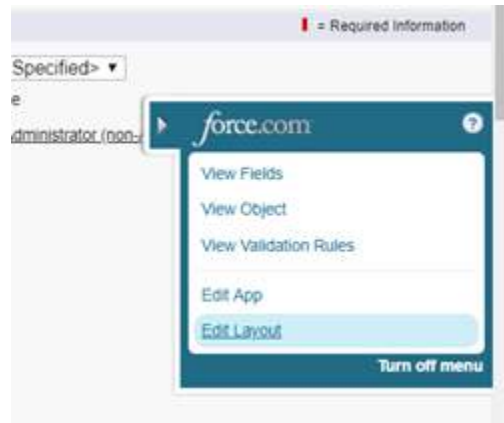
The first step in the configuration of KSL for Salesforce is to configure existing users by granting them rights on KSL For Salesforce® app.

- Grant KSL Suite role to give users rights to create, modify, generate or delete documents on KSL server.
- Select Setup to access to the user administration, then select ADMINISTRATION > Users > Users.

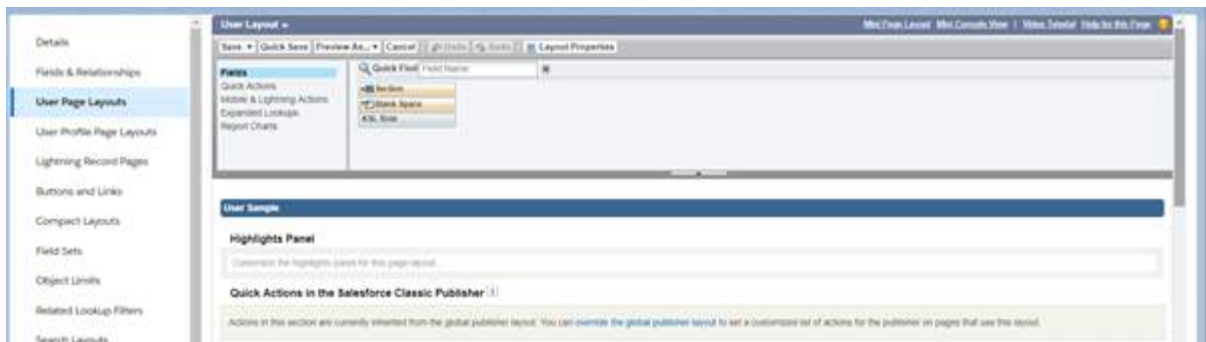


- Click Edit to select a user.

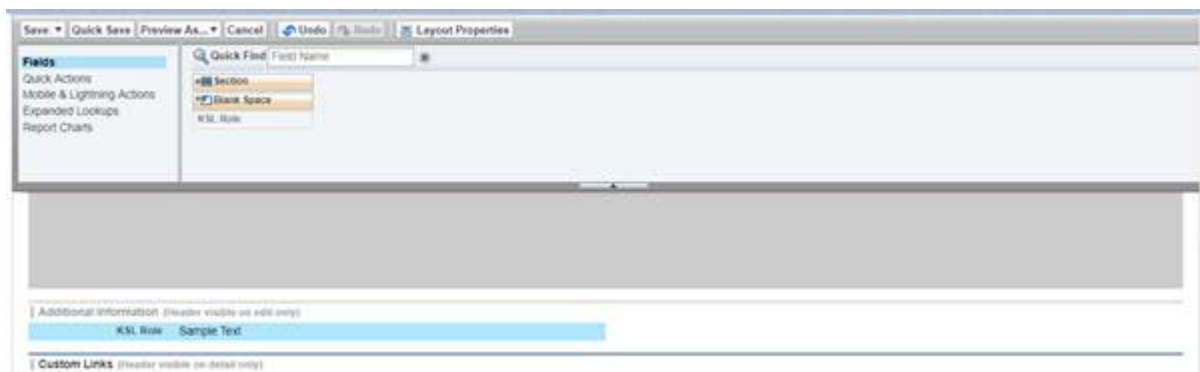
Action	Full Name ↑	Alias	Username
<a href="#">Edit</a>   <a href="#">Login</a>	BONNETON_PASCAL	PRONN	salesforce@naelan.com.testlogin
<a href="#">Edit</a> - Record 2 -	BONNETON_PASCAL	Chatter	chatty_00d110000000otaeau_fm55dezvna@chatter.salesforce.com
<a href="#">Edit</a>	unused	unused	unused@unused.fr.testlogin
<a href="#">Edit</a>   <a href="#">Login</a>	User_Admin	AllUser	coa.n3oxpdwvwhh_ds6imvdu/vzo@naelan.com.testlogin
<a href="#">Edit</a>   <a href="#">Login</a>	Vérité_Corine	cverite	verite@naelan.com.testlogin



- Click on the arrow at the right of the screen and select Edit Layout. The user layout page is displayed (User Page Layouts).



- Add the field KSL Role (KSL\_Role\_\_c) to your User Page Layout.
- Drag and drop the KSL Role available field under Additional information and Save.



- On the user's details, set the KSL Role based on the rights you want to give him; this role corresponds strictly to the group set in KSL Admin (not in KSL Administration).

The screenshot shows the 'Users' setup page in Salesforce. The 'Additional Information' section contains a 'KSL Role' dropdown menu, which is highlighted with a red arrow. Other sections include 'Mailing Address', 'Single Sign On Information', 'Locale Settings', and 'Approver Settings'.

- In Users > Permissions Sets, select KSL User, click Manage assignments, and assign it to each user that will use the KSL for Salesforce® features.
- Select KSL Admin, click Manage assignments and select each user that will use KSL for Salesforce®. These users will have access to the resource repository and will be able to generate XSD/XML data files.

*Note that KSL Admin permissions are an extension of KSL User. A user with this permission level must be a KSL User to access to the KSL plug-in features.*



## 3.2. Configuring the communication between Salesforce® and KSL servers

### a. Remote site

KSL for Salesforce® communicates with a KSL server; this server processes coming from Salesforce® requests and generates documents and emails. To allow this communication, you must declare a remote

site in Salesforce® with the remote server information.

In Setup, look for Remote Site Settings.

Create a New Remote Site:

- Remote Site Name: name of the remote site
- Namespace prefix: ksl
- Remote Site URL: (example: <https://hostname:XXX> with XXX the https port number)

#### Remote Site Details

Remote Site Detail

Remote Site Name: KSLServer

Namespace Prefix: ksl

Remote Site URL: [REDACTED]

Disable Protocol Security:

Description: Active

Created By: User User, 18/02/2019 15:52

Modified By: User User, 18/02/2019 15:52

Buttons: Edit, Delete, Close

#### b. Connexion parameters

The connexion parameters must be completed in the KSL Administration part, to allowed KSL Server to communicate with Salesforce®. More information in the paragraph 4.1.1.

#### c. Custom Metadata - Allowed Objects

Generating a data schema (.xsd) to create a document template refers to this interface that lists standard Salesforce® objects by default.

In Setup, look for Metadata Types.

On the Custom MetadataTypes page, on the allowed objects line, Clic on Manage records, then Edit.

All Custom Metadata Types Help for this Page ?

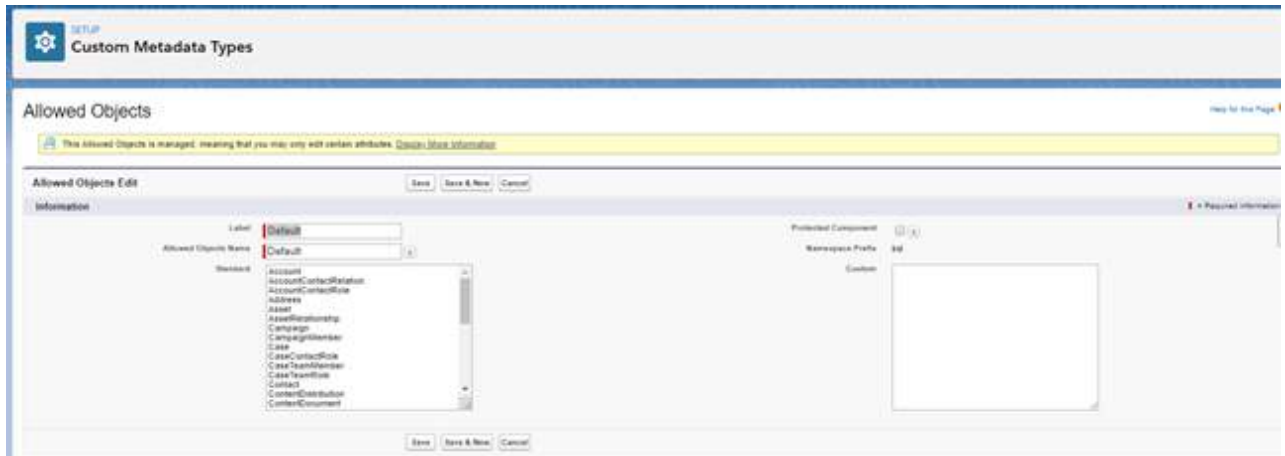
Custom metadata types enable you to create your own setup objects whose records are metadata rather than data. These are typically used to define application configurations that need to be migrated from one environment to another, or packaged and installed.

Rather than building apps from data records in custom objects or custom settings, you can create custom metadata types and add metadata records, with all the manageability that comes with metadata: package, deploy, and upgrade. Querying custom metadata records doesn't count against SOQL limits.

Action	Label	Namespace Prefix	Visibility	Api Name	Record Size	Description
<a href="#">Manage Records</a>	Allowed Objects	ksl	Public	ksl__AllowedObjects__mdt	651	
<a href="#">Manage Records</a>	Auth Configuration	ksl	Public	ksl__AuthConfiguration__mdt	1416	
<a href="#">Manage Records</a>	KidObjet Description	ksl	Public	ksl__KidObjet__mdt	2181	

If you want to use a custom object, add the API of the object to the Custom field. You can also add a Salesforce® standard object to this list if it is not present in the standard list.

Separate each custom object by a carriage return.



You can find more information about the XSD Generation in the paragraph 4.1. XSD schema generator tab

#### d. Custom Objects

By default, KSL For Salesforce® provides two objects linked to the Opportunity objects to enable you to create documents from opportunities. The KSL Document objects linked to the Opportunity object displays the interactive KSL document, whereas the KSL Document (PDF) object displays the PDF document stored in KSL but accessible from Salesforce®.

You can find more information about these objects in the following paragraphs: 5.2. KSL Document and 5.3. KSL Document (PDF).

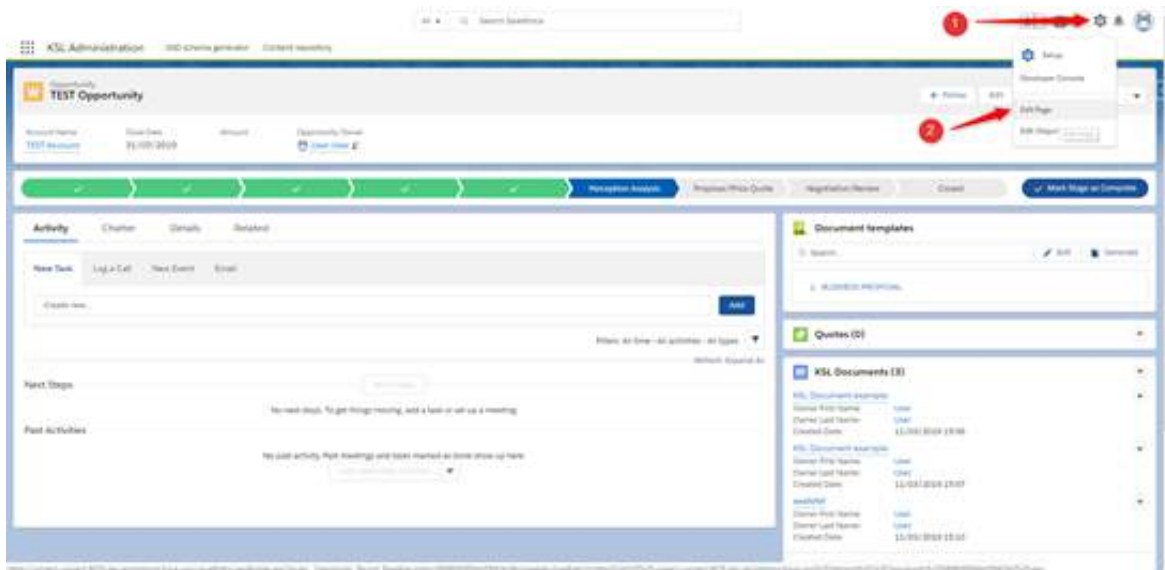
### 3.3. KSL components - Access to template and documents

To be able to generate KSL Document from Salesforce® objects, you need to configure your object layout to display the KSL For Salesforce® elements.

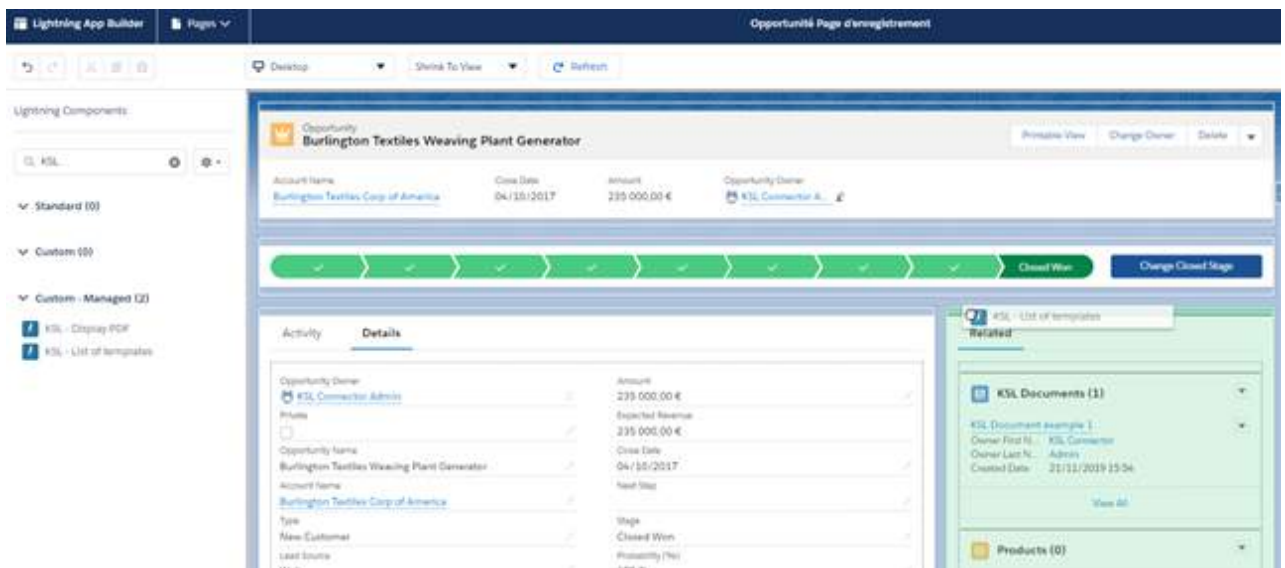
The component Document templates will display the templates available in the KSL Project configured in the custom metaData type Auth. Configuration. The user will then be able to create a KSL document based on a chosen template and its launching mode: e.g. PDF on demand generation or document interactive edition.

The following example guides you to display and configure the component on the Opportunity Object page layout. The process is the same for any standard or custom object in Salesforce.

- Go to the detail page of an Opportunity (Opportunities tab), click on the setup icon and then click on Edit page.

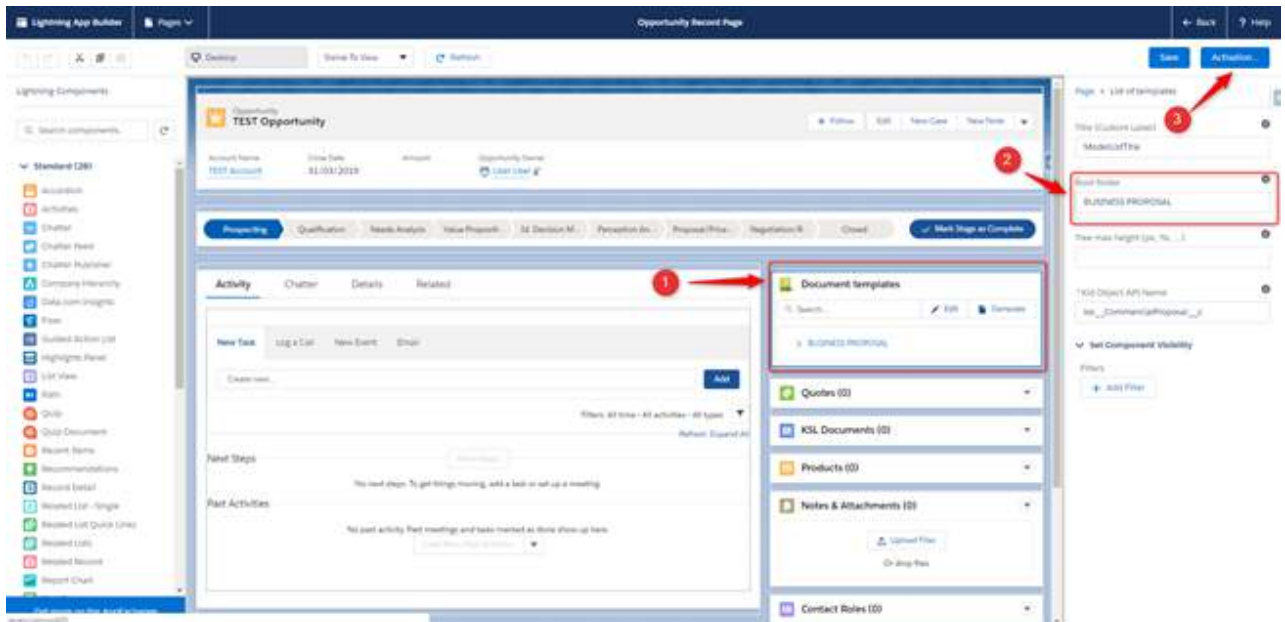


- From the Custom Managed list (on the left), drag and drop the lightning component KSL - List of templates where you want in the page.



- In the option layout of the component (on the right), configure the following options:
  - Title: name of the component record. The entered value must be a Custom Label. The Custom Label ModelListTitle takes the object title (KSL Documents and Email) with its translations.
  - Root folder: It is possible to filter the classification tree of the KSL project where templates and resources are stored, by indicating the root node you want to display for the object. Your user must, of course, have the necessary permissions on the node and its potential branches. By default, this parameter is set to OPPORTUNITY for the Opportunity object and CAMPAIGN for the Campaign object.

- Tree max height (optional): Maximum height of the component on the record page (pixel, percentage,...)
- Kid Object API Name: API name of the Salesforce® object that represents the KSL Document in Salesforce®. By default, KSL For Salesforce® provides the object KSL Document (ksl\_\_CommercialProposal\_\_c) linked to the Opportunity object.



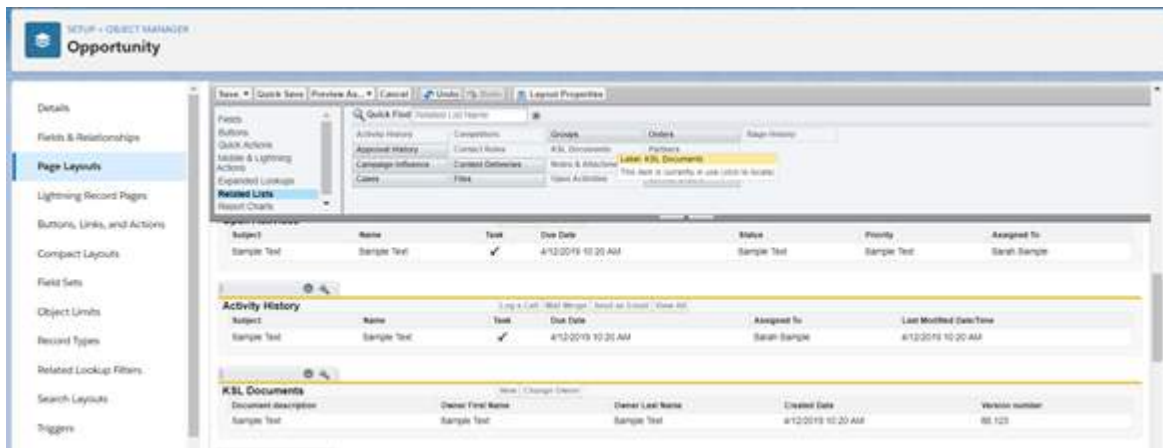
- Définir la visibilité du composant permet de conditionner l'apparition/disparition de la liste des modèles. Avec un filtre sur une étape de l'opportunité par exemple : En déposant le composant "KSL - List of templates" plusieurs fois sur la page, avec un nœud du plan de classement différent (root folder) , cela vous permet d'affiner les modèles de document accessibles et surtout opportuns sur l'étape en cours.
- Set Component Visibility condition the appearance / disappearance of the document templates. With a filter on a step of the opportunity for example: By depositing the component "KSL - List of templates" several times on the page, with a different root folder, this allows you to refine the access of the appropriate document templates on the current step.

You have to activate (Activation) the lightning page before saving (Save) before closing the window.

To add the list of the available KSL documents of the opportunity,

- From configuration > Setup
- Tab Object Manager > Opportunity > Page Layouts
- Click Opportunity Layout

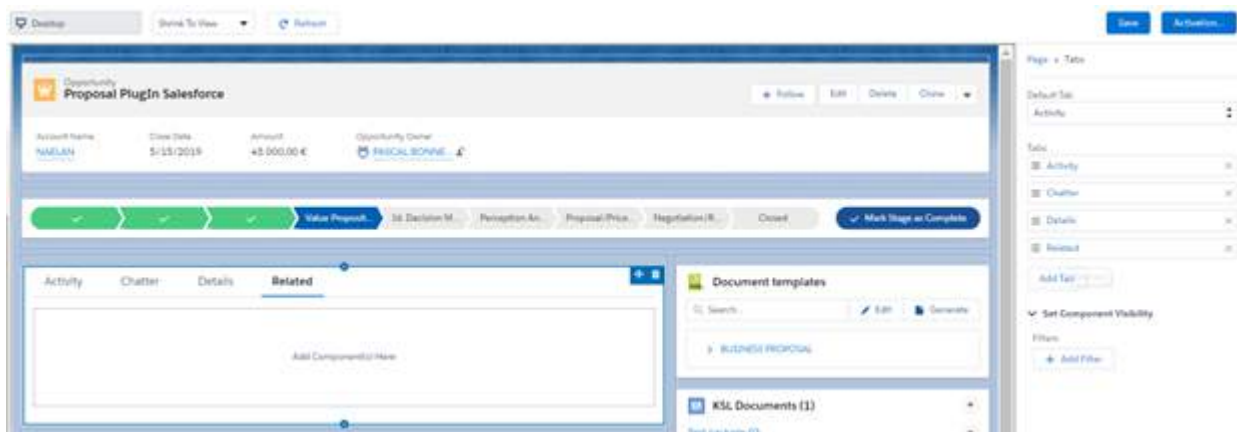




- Select Related Lists
- Select Ksl Documents and drag and drop the list at the 1<sup>st</sup> position in the existing related lists for the opportunity ; click on Save ; the list appears in the opportunity on the left side, under the list of document templates

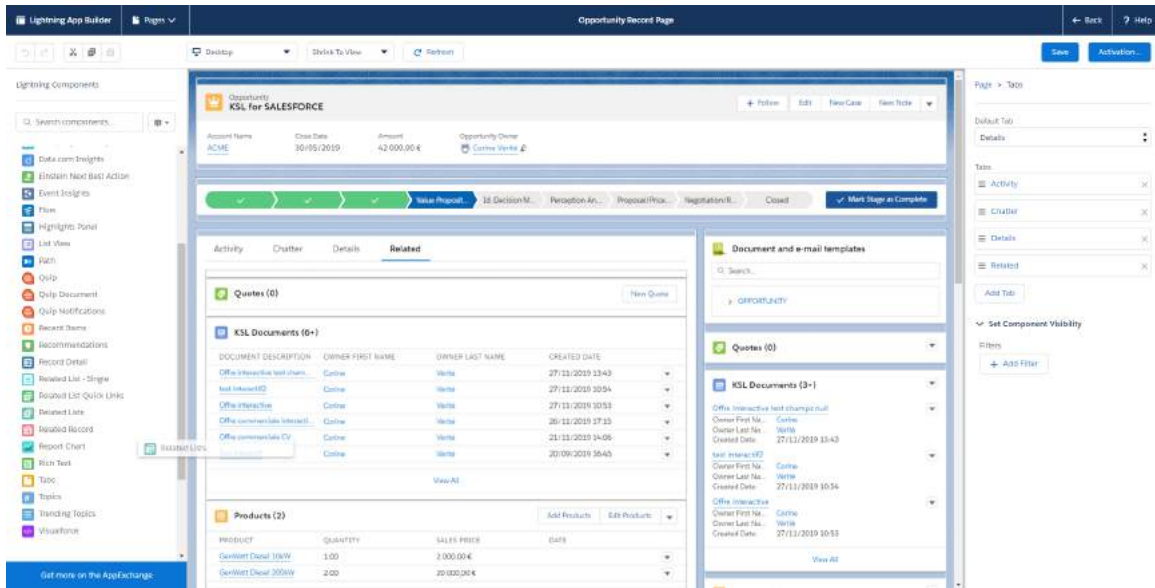
To add the Related tab in the page body and consequently displaying the document list:

- Configuration icon, then Edit Page
- On the Tab screen area, the tab list appears on the right
- Click Add Tab; a new Detail tab is added to the list; rename the tab by clicking on Detail and select Related from the list provided. The new tab Related appears on the opportunity body.



- On the Related tab, select Related Lists from the lightning component list (available at the left of the screen), and drag and drop it in the tab body. The complete list of « related » elements, and especially the KSL documents, is available in the opportunity body. Click Save and Activation.

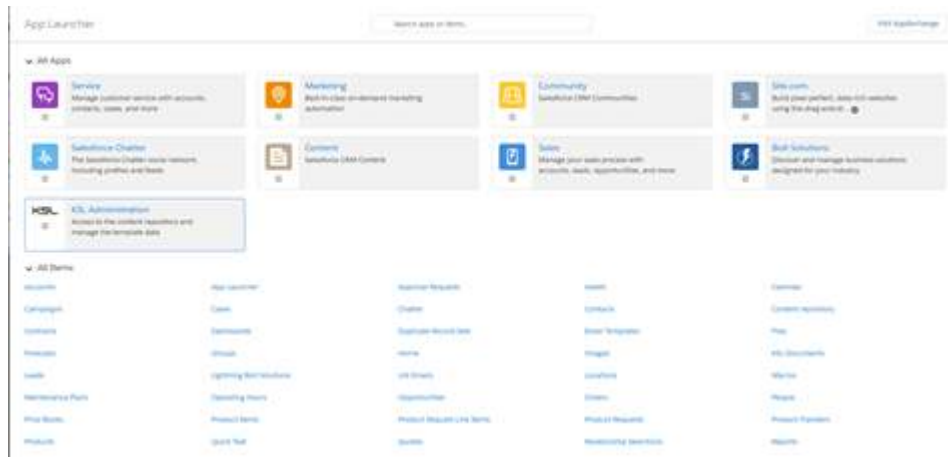




*Note that this display can constitute a repetition with the right column; this use case will have to be detailed with the business users.*

## 4. KSL For Salesforce® for administrators

Select the application KSL Administration from the Salesforce® App Launcher menu.



This KSL Administration application only concerns KSL for Salesforce® plugin et does not response to the same needs as KSL Admin application, which is used to setup and supervise a KSL Server.

KSL Administration includes 2 tabs:

- Setup
- Content repository

### 4.1. Setup tab

The Setup tab contains 2 sub-tabs:

- Auth. Configuration
- XSD Generator

#### a. Auth. Configuration

This tab allows to configure the communication between Salesforce® and KSL server :

- KSL Server endpoint: URL which allows to communicate with KSL Office application ; the URL must end with the reference of KSL Office application, example: https://host21.naelan.fr:8443/ksl\_office/
- KSL Project Name: name of the KSL project which hosts templates and resources. This project will be present on the KSL remote server, (example: KSL\_FINANCE)
- Token - Audience field override: this field can be completed, if necessary, with the audience field defined in the KSL parameters, usually an URL (example: https://test.salesforce.com).
- Token - Public Key: a couple “public/private key” is necessary for oAuth2 authentication. This field contains the same public key than the one configured on the remote KSL web server. This key is used by the KSL server to control that the Salesforce configuration is authorized.
- Token - Private Key: a SHA256 256bytes private key which is used by the KSL remote server to decrypt the identification JWT token. This key is hidden after saving the settings and must be entered at each parameters update.

Additional informations are given on tools that can be used to generate public and private keys in the appendix of this document.

## b. XSD Generator

This tab allows the administrator to generate an XSD schema associated with Salesforce® objects. It also allows you to generate a test data file to initiate the design of a document or e-mail template.

XSD use

This XSD is used by the KSL Studio design tool to build the document templates based on the data of the selected objects. A template is associated with an XSD. The same XSD can be used by several templates.

When generating a document, the KSL plug-in asks the KSL server for the XSD schema associated with the selected template; the plug-in then retrieves the necessary data in the Salesforce® objects, and instantly creates a stream in XML format with this data. This XML conforms to the XSD and allows the KSL server to provide the field values to generate the document.

*Note: this exchange makes it possible to detect as soon as possible any discrepancies between the data available in Salesforce® and the data expected by the document template; without this mechanism, deleting for example a Salesforce® field could cause a problem when generating the document, or even generate a document with erroneous contents. In the event of such an offset, an error is displayed in Salesforce® when the document is generated.*

## XSD generation

The XSD generator allows you to select the main object of your XSD. The builder automatically proposes the direct relationship (Parent) and indirect relationship (Child) objects that exist for this object in your Salesforce® environment. The checkboxes allow you to choose the objects you want to include in your XSD, in relation to the fields you will use in your KSL document templates.

The objects available in the menu are configured in the Metadata Types, Allowed Object (more information in the chapter 3.2.c)

The screenshot displays the 'Génération d'un XSD' configuration page in the KSL Administration tool. The main object is 'Opportunité'. The interface is divided into three main sections:

- Relations Directes:** A table listing direct relationships. The 'Account' relationship is selected with a checkmark.
- Sous-Relations:** A table listing indirect relationships. 'OpportunityContactRole' and 'OpportunityLineItem' are selected with checkmarks.
- XSD Preview:** A code editor showing the generated XML Schema Definition (XSD) for the 'Opportunity' object, including elements like 'Id', 'IsDeleted', 'AccountId', 'Name', 'StageName', 'Amount', 'Probability', 'ExpectedRevenue', 'TotalOpportunityQuantity', 'CloseDate', 'Type', and 'NextStep'.

*Note: You cannot select the same types of objects more than once in the same relationship tree. For example, if you select Account, you will not be able to select the parent Account because they belong to the same type of object in the same tree; KSL document templates will not allow it.*

The Generation tab of an XSD includes several buttons for generating XSD and XML:

- The Refresh XSD button

Click this button after selecting the objects you want to insert into your XSD in order for the builder to generate XSD file in the right window. This window contains all the fields of the selected objects.

- The Download XSD button

This button allows downloading the XSD file and using it in KSL Studio to create a document template. It will not work if you have not previously clicked Refresh XSD.

When you create a document template based on this XSD, this XSD is linked to the template. With each KSL document generation (from a template), Salesforce® ask for this XSD and use it to retrieve the values of the Salesforce® object fields described in the schema.

Special attention when changing of the data schema:

- If you create a new field in an object used in a KSL document template:
  - If you want to use this field in your document, generate a new XSD, download it, update the XSD component in the KSL repository template and update the KSL template using it.
  - If you do not want to use the field, do not do anything. Users will still have the option of generating or customizing the KSL document in Salesforce® (the new field will be ignored).
- If you delete a field of an object used in a KSL document template:
  - Generate and download a new XSD; download it, update the XSD component in the KSL repository template and update the KSL template using it.
  - The field should also be removed from the document layout. If you do not perform this update, an error will be returned to the users who will no longer be able to use the document template in Salesforce®.
- If some objects present in your XSD are not finally associated with the main object (opportunity for example)
  - The XML generated will not be consistent with the XSD, which will cause a document generation error.
  - However, it is possible to make these objects optional by manually modifying the XSD (minOccurs = 0). Note that fields from potentially missing objects will be empty.

*Note that it is possible to delete some fields of the XSD that would not be necessary for the template in order to reduce the size of the data stream exchanged between Salesforce® and the KSL servers, and also to limit the number of fields accessible from the KSL Studio tool to simplify the selection of fields during template design.*

- The Download XML button (version 1.6 and higher of the KSL plug-in )

This button is used to generate and download a test data file compliant with the XSD schema. It will not work if you have not previously clicked Refresh XSD.

Each data is surrounded by XML tags:

- A text field data is the field name
- A date field data is a random date
- An integer field data is a random integer
- A decimal field data is a random decimal number
- A boolean field data is set to "true" or "false"

Examples:

- <BillingStreet>BillingStreet</BillingStreet>
- <LastActivityDate>2020-09-12 00:00:00</LastActivityDa
- <NumberOfWonOpportunities>83</NumberOfWonOpportunities>
- <Quantity>58.04</Quantity>
- <IsDeleted>true</IsDeleted>

## 4.2. Content repository tab

This tab opens in KSL Office on a new tab, the repository containing shared components. The functions of this application are detailed in KSL Office user guide.



## 5. Add the Plug-In components to new objects

### 5.1. The object creation

By default, KSL For Salesforce® proposes to create documents from the Opportunity object. But you can potentially use KSL For Salesforce® for all of your Salesforce® or custom objects. This chapter explains how to configure the application in this case.

You will need to create two custom objects for each Salesforce® object where you want to add the KSL Document templates component:

- A KSL interactive document object linked to your object; he will present the reference to the interactive KSL document - KSL Document for example.
- A KSL document version object also linked to the object; it will present the PDF versions of your interactive document - KSL Document (PDF) for example.

For the KSL interactive document object, you must create at least the two following fields:

- An external identifier text field to store the identifier of the KSL document.
- A text field to store the name of the template used for your document.
- A search relationship field related to the object from which you are producing your document.

For the KSL document version object, you must create at least the following fields:

- A master-Detail relationship field related to the custom object described above.
- An external identifier text field to store the version identifier of the KSL document.
- A numeric/integer field to store the version number of the KSL document

If you want to see the version number in the KSL interactive document object, you can add a summation summary field (COUNT) from the KSL document version object.

You are then free to add any other field to these two objects.

### 5.2. Metadata

After creating the previous object, you must declare it in the KidObject Description metadata.

- In Setup, locate the Custom MetaData Types topic
- Click Manage Records on KidObject Description to add a new record.

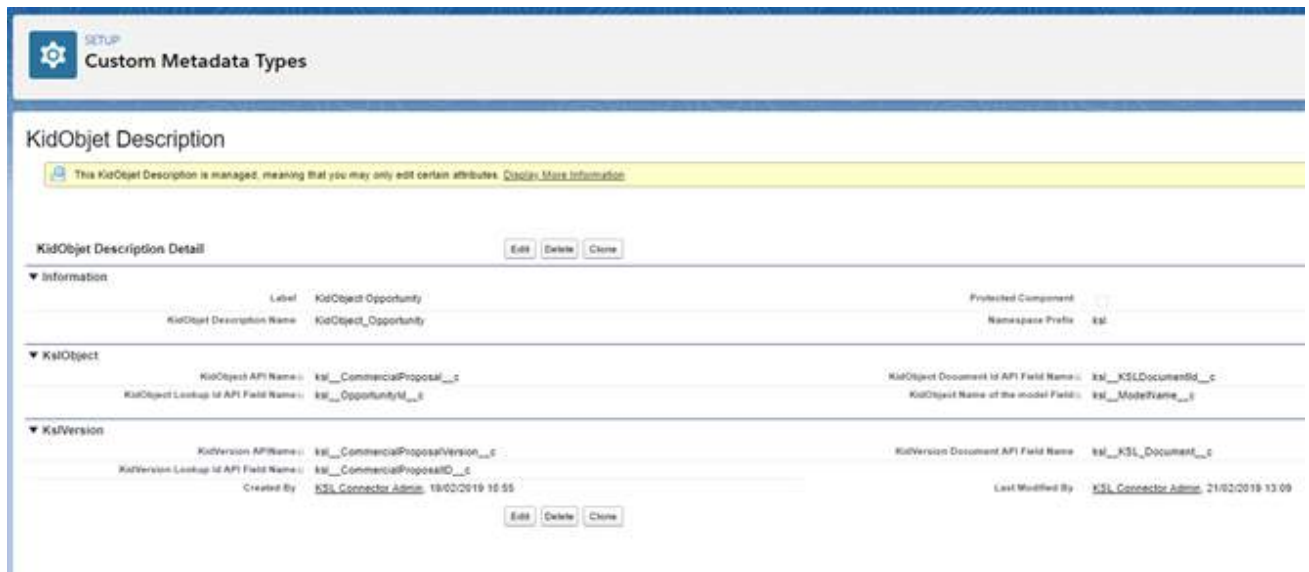
In this metadata, you will need to store the structure information of your KSL interactive document object and KSL document version, for the previously created objects.

The following information is needed:

- KidObject API Name: API name of the KSL interactive object created previously

- KidObject Lookup Id API Field Name: API field name for the search relationship of your KSL interactive document
- KidObject Document Id API Field Name: API field name that store the KSL document Id for the object KSL interactive document
- KidObject Name of the model Field: API field name that store the name of the template used to create the interactive document.
- KidVersion APIName: API field name for the KSL object previously created
- KidVersion Lookup Id API Field Name: API field name for the search relationship of your KSL object document.
- KidVersion Document API Field Name: API field name that store the PDF identifier for the KSL document object.
- KslVersion Version API Field Name : API field name that store the version number of the PDF for le KSL Document object.

You will find an example of KidObject description for KSL Document and KSL Document (PDF), which allows you to use KSL For Salesforce® on opportunities:



### 5.3. The component

After completing the previous steps, you can follow the instructions in section 3.3 The KSL components, to add the component KSL - List of templates to the object from which you want to produce KSL documents.

### 5.4. The actions


To access the Ksl For Salesforce® features on new objects, you must create the following actions:

Configuration icon > Object Manager




Select the KSL interactive document object mentioned above,  
choose Buttons, Links and Actions, then Create Action:

- Delete KSL: choose the RemoveDocument Lightning component to remove the KSL interactive document and the KSL document, as well as the Salesforce® record.

Action Detail		<a href="#">Edit</a>	<a href="#">Delete</a>
Label	Delete KSL	Object Name	<a href="#">Commercial Proposal</a>
Standard Label Type		Action Type	Lightning Component
Name	Delete_KSL_Document	Icon	
Namespace Prefix	ksl		
Description			
Lightning Component	<a href="#">RemoveDocument</a>		
Height	70px		
Created By	<a href="#">KSL Connector Admin</a> , 19/02/2019 10:55	Modified By	<a href="#">Blandine ALVAREZ</a> , 22/02/2019 14:45
		<a href="#">Edit</a>	<a href="#">Delete</a>

- Edit KSL: choose the EditKSLDocument lightning component to open the KSL window and customize the KSL interactive document.

Action Detail		<a href="#">Edit</a>	<a href="#">Delete</a>
Label	Edit Document	Object Name	<a href="#">Commercial Proposal</a>
Standard Label Type		Action Type	Lightning Component
Name	EditKSLDoc	Icon	
Namespace Prefix	ksl		
Description	Open Launck Kid to edit the interactif document		
Lightning Component	<a href="#">EditKSLDocument</a>		
Height	70px		
Created By	<a href="#">KSL Connector Admin</a> , 19/02/2019 10:55	Modified By	<a href="#">KSL Connector Admin</a> , 20/02/2019 14:45
		<a href="#">Edit</a>	<a href="#">Delete</a>

- "Send by e-mail" : Choose the EmailLauncher lightning component to open the e-mail send box.

Action Detail		<a href="#">Edit</a>	<a href="#">Delete</a>
Label	Send by email	Object Name	<a href="#">KSL Document</a>
Standard Label Type		Action Type	Lightning Component
Name	SendByEmail	Icon	
Namespace Prefix	ksl		
Description			
Lightning Component	<a href="#">EmailLauncher</a>		
Height	120px		
Created By	<a href="#">KSL Connector Admin</a> , 24/10/2019 15:53	Modified By	<a href="#">KSL Connector Admin</a> , 24/10/2019 15:53
		<a href="#">Edit</a>	<a href="#">Delete</a>

Select the KSL document object mentioned above,  
choose Buttons, Links and Actions, then Create Action:

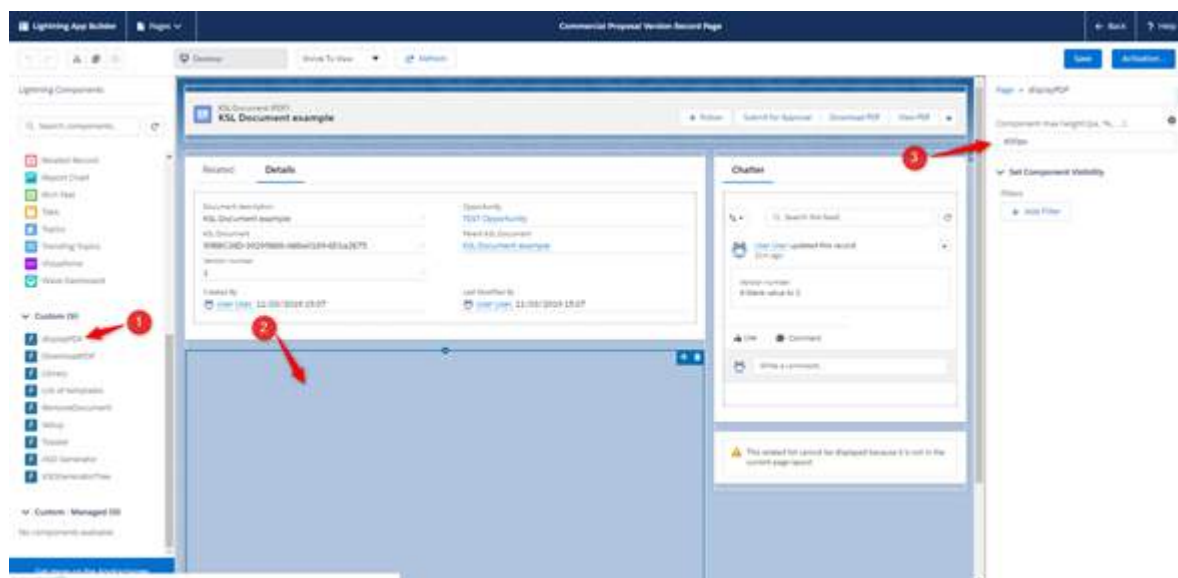
- Delete KSL: choose the RemoveDocument Lightning component to remove the KSL document and the Salesforce® record.
- View PDF: choose the ViewPDF Lightning component to open the PDF file in another browser tab.
- "Send by e-mail" : Choose the EmailLauncher lightning component to open the e-mail send box.

...and you can add the previous action to the layout of the objects.

## 5.5. The Display PDF component

KSL For Salesforce® allows you to view the PDF directly on the page of the KSL document object.

Add the displayPDF component to the presentation of the Lightning page associated to your object. You will be able to choose the height of the component in the options.



## 6. Validation workflow example

In Salesforce®, you can configure validation workflow to validate some records and especially KSL documents.

In this section, you will find a simple example of a validation workflow that you can configure for the KSL (PDF) document. For more information on validation workflow, you can read the Salesforce® documentation.

### 6.1. Creation of a queue

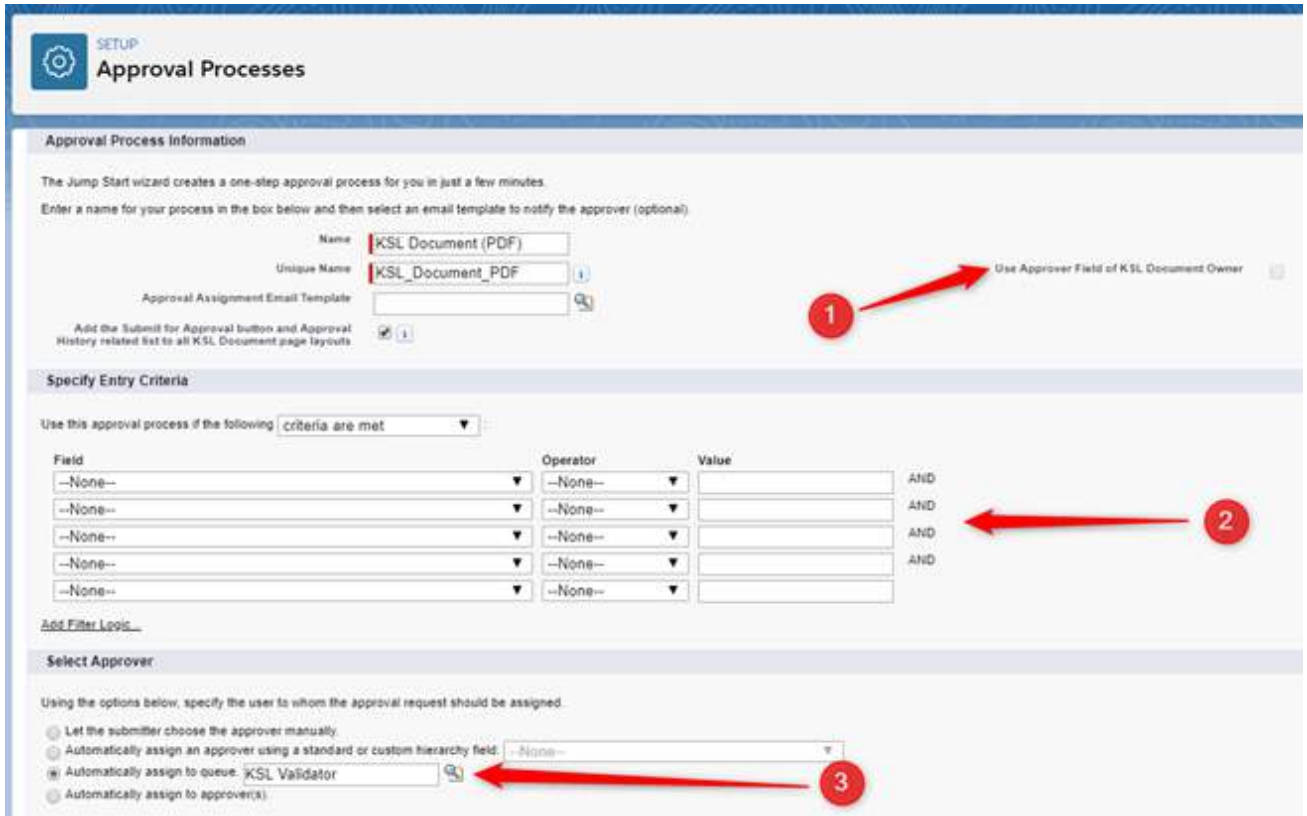
If you have multiple users assigned to validate the document, you can create a queue to manage it. In Setup, find Queues and click the button New.

The screenshot shows the 'Edit Queue' interface in Salesforce. At the top, it says 'SETUP Queues' and 'Edit Queue KSL Validator'. There are 'Save' and 'Cancel' buttons. The 'Queue Name and Email Address' section includes a 'Label' field with 'KSL Validator', a 'Queue Name' field with 'KSL\_Validator', and a 'Queue Email' field. A red arrow labeled '1' points to the Queue Name field. Below this is a 'Supported Objects' section with a description: 'Select the objects you want to assign to this queue. Individual records for those objects can then be owned by this queue.' It features two lists: 'Available Objects' (Case, Goal, Image, Knowledge Article Version, Lead, Location, Macro, Maintenance Plan, Metric, Order, Product Item, Product Request, Product Transfer, Quick Text) and 'Selected Objects' (KSL Document). A red arrow labeled '2' points to the 'Selected Objects' list.

- Choose the name of your queue
- Choose the objects that will be assigned to this queue. In our example KSL Document.
- Choose the members of the queue. These can be users, roles, or public groups.

### 6.2. The validation workflow

In Setup, search for Approval Processes. Choose the object on which you want to create your validation workflow, in this example KSL Document (PDF), and then Create New Approval Process.



- Choose the name of your validation workflow and the e-mail template to send to the approver if necessary.
- Specify whether KSL documents (PDF) are automatically transmitted to the validation workflow and the condition. Otherwise, the user may manually submit his KSL document (PDF) to the validation workflow.
- Select the approval of the KSL Document (PDF). In this example, the record is assigned to the previously created queue.

On the Approval Process (Layout) screen, you can add new approval steps and choose whether the Salesforce record (the document in the context) should be locked when it is approved or rejected.

## 7. KSL for Salesforce® for template designers

### 7.1. Mandatory KSL functional parameters

When creating new KSL templates, i.e. graphical editions of documents and the associated editing services, and e-mail templates, it is necessary to create the following 3 functional parameters of string type:

- P\_FILE: This parameter is used to identify the XML stream transmitted by the plug-in to the editing service. It must be initialized to 250 characters
- P\_LABEL: this parameter is used to transmit the description of the document; it corresponds to the field Document description - see chapter 5.1. and 5.2; it can be initialized to 50 characters
- language: This field passes the language used by the user in their Salesforce® application; this field is for example "FR" or "en\_US"; we recommend setting it to 10 characters.

### 7.2. Other KSL functional parameters

Other KSL functional parameters can be added for use in the Document parameters view of the interactive document or email (for example, to allow the user to select a table of contents type or to select optional annexes).

These other parameters can not be valued by Salesforce® through parameters; to meet this need, additional data will have to be added to the transmitted XML stream.

### 7.3. Considerations on Salesforce® data sent to KSL

Document and email generation with KSL relies on XML data conformed to an XSD schema.

Examples of XSD schemas and XML streams can be generated using the KSL Administration application. On this point, KSL Administration is only a practical tool that facilitates the work of the KSL template designer. See chapter KSL For Salesforce® for Administrators for more information.

### 7.4. Use of XML data in KSL

Before creating templates, be sure to set some of the following Advanced run parameters options in KSL Studio:

- XML file decimal character
- XML file thousands separator
- Date format in XML file

See the KSL Studio User Guide for more details.

This setting must also be reported in the project on KSL Server. Please contact the Naelan teams for this

operation.

## 7.5. Create on-demand email template

To create on-demand e-mail templates, proceed as follows. This process is explained using the object Opportunity but any other object can be used.

*Note: The following chapter is dedicated to the particular case of e-mailings.*

### a. Step 1: Generate the XSD and XML streams corresponding to an e-mail

In Ksl Administration, generate and download the XSD schema corresponding to the data personalizing the e-mail (and if necessary an associated XML test data file). This XSD schema will be used by the KSL service(s) in charge of creating the HTML bodies of e-mails.

*Example: suppose that the body of the emails to be generated has to include the data corresponding to the Opportunity name and amount, but also lastname, the firstname of the first Contact Roles of the opportunity.*

*In this context, it is necessary to generate the XSD and XML files from the Opportunity object which contains the Name and Amount elements, but also the elements LastName, FirstName and Salutation of the OpportunityContactRole node of the opportunity.*

Example of XML data corresponding to this XSD schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<Opportunity>
  <Name>My opportunity</Name>
  <Amount>10000,00</Amount>
  ...
  <OpportunityContactRole>
    <ContactId>
      <LastName>MARTIN</LastName>
      <FirstName>John</FirstName>
      <Salutation>Mr</Salutation>
    ...
  
```

### b. Step 2: create a mapping file

The mapping file is used to create variables inserted in the body of an e-mail:

- These variables can be selected and inserted by the designer or the content administrator in the body of each e-mail template within text zones.

- These variables can be selected and inserted by the Salesforce user when he edits the body of his e-mail
- These variables are automatically substituted during the e-mail body generation with data from the XML stream coming from Salesforce
- In KSL Admin, create a mapping file with a .map extension:
  - The mapping file is a text file
  - It defines each field used by an e-mail (field used in the subject, the body of the e-mail, or the recipient fields if it is an e-mailing)
  - Each field is defined by a line of the file

Each line is composed of 3 strings separated by the character ";"

- The 1<sup>st</sup> string is the name of the field that will be used in the e-mail: if used in the template, this field will be the one displayed in the interactive personalization interface, otherwise the end user will be able to insert it in a text zone of the e-mail in this interface
- The 2<sup>nd</sup> string is the element of the XML stream that contains the value to display (single-child node)
- The 3<sup>rd</sup> string is the element of the XML, which is the of the element defined just above, that contains the value to display

*Example with the <SF\_Email\_Mapping.map> mapping file:*

```
OPPORTUNITY_NAME;Name;Opportunity;  
OPPORTUNITY_AMOUNT;Amount;Opportunity;  
CONTACT_LASTNAME;LastName;ContactId;  
CONTACT_FIRSTNAME;FirstName;ContactId;  
CONTACT_SALUTATION;Salutation;ContactId;
```

Example:

*The XML stream and the mapping file given above will allow you to automatically substitute the e-mail variables with the Salesforce data with the following result:*

- *OPPORTUNITY\_NAME = My opportunity*
- *OPPORTUNITY\_AMOUNT = 10000,00*
- *CONTACT\_LASTNAME = MARTIN*
- *CONTACT\_FIRSTNAME = John*
- *CONTACT\_SALUTATION = Mr*

**Important notes:**

- *Note that only the parent node of each element is to be defined*
- *It is possible to define several fields that point to the same element*
- *If one of the elements is iterative, the field takes the value of the element of the first iteration*
- *Each line ends with a line break (including the last line)*

- In the case where the parent element exists several times in the flow, it is possible to specify the parent node of the parent node (example: SALUTATION\_CONTACT; Salutation; OpportunityContactRole.ContactId;)

- From the KSL Administration application, classify this mapping file in the KSL repository:
- Select the Content repository tab
  - Add the mapping file as an external file and classify it in the tree structure

### c. Step 3: Create a style sheet for your new e-mail template

If you already have a style sheet, go directly to the next step.

Otherwise, from KSL Studio (or KSL Office from KSL Server version 8.0.2.0), you can duplicate the FS\_SALESFORCE\_EMAIL style sheet and modify it to match your graphic charter.

### d. Step 4: Create the body of your e-mail

From the Content repository tab of the KSL Administration application, create a group of text zones. This group will be the body of your e-mail template.

This group of text zones contains one or more text zones that are also created from the Content Repository application. These text zones must use the style sheet created in the previous step 3.

The variables correspond to the Salesforce data are created in the text zone that require it.

- The value of each variable comes from the XML stream sent by Salesforce
- Each variable is in string format
- The name of each variable corresponds to the 1st element of the line of the mapping file that defines it

*For the example described above, the following variables will be created:*

- *OPPORTUNITY\_NAME*
- *OPPORTUNITY\_AMOUNT*
- *CONTACT\_LASTNAME*
- *CONTACT\_FIRSTNAME*
- *CONTACT\_SALUTATION*

### e. Step 5: Create a new KSL document template for the e-mail template

In this context of creating a new e-mail template, it is necessary to create a new KSL document template from which a new KSL Server service will be created in step 6.

The procedure from KSL Studio is as follows:

- Add the XSD file generated in step 1 to the KSL project classification plan



- Duplicate the generic edition SF\_Email\_500px and add it to the KSL project classification tree
- In this new document template, select the style sheet by selecting the style sheet component created in step 3 above
- Modify the XML request for this document template by specifying the new XSD
- Delete the group in the document template
- Insert the new text zone group created in step 4
- Save and Update and release the document template

#### f. Step 6: Create the document service corresponding to the email template

- From KSL Admin, create the document service associated with the document template created in step 5
- Set a label in the description field: this description corresponds to the label displayed for this e-mail template in the list of templates presented to the Salesforce user in its Document Templates component.
- Classify this service in the project tree: the document service should be classified in a branch of the tree accessible to the Salesforce user in its Document Templates component accessible from the Opportunity object.
- Specify the preview action group: EmailActions
- Fill in the parameters of this document service using the information below

Parameter	Default value or example	Description
feuillestyles (stylesheet)	FS_SALESFORCE_EMAIL	Select the style sheet used for editing (without extension)  Deselect the option "Show this parameter in Ksl Office"
MailAccount		<i>Parameter not used in the case of a single on demand e-mail sent by Salesforce®</i>  Deselect the option "Show this parameter in Ksl Office"
MailBat	False	Set this parameter to False for this single on-demand e-mail sending case  Deselect the option "Show this parameter in Ksl Office"
MailCharset	UTF-8	Set this parameter to UTF-8  <i>Encoding of the CSV file generated from the XML stream sent by the KSL plug-in</i>  <i>Set this parameter by default to the universal UTF-8 code page; it is possible to change this encoding (example: UTF-8, Windows-1252, ISO8859-15 (Linux))</i>

		Deselect the option "Show this parameter in Ksl Office"
MailContentGroup	Example: <SF_Email_ContractConfirm.ksg>	<p><i>Optional parameter for transmitting the group of text fields used as e-mail body</i></p> <p><i>If not specified, the group inserted by default in the document template is used</i></p> <p><i>This parameter requires that the document template dynamically calls the group (default group dynamically called with the expression MailContentGroup)</i></p> <p><i>The group is defined between the characters &lt; and &gt; with the extension .ksg</i></p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
MailCsvFile	<default>	<p>This file is used only if the P_MAP_FILE file is not defined.</p> <p>The default value can be used for debugging tests without XML stream. A value between the characters &lt; and &gt; indicates that the file is in the KSL repository.</p> <p>The CSV file is a text file with one contact / recipient per line, possibly with a header line that names the columns and column separators ";";</p> <p>You need to set a default value (for example "&lt;default&gt;") even if the P_MAP_FILE is defined</p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
MailCsvHeader	Correspond to the header line of the CSV file without #	<p><i>Parameter not used in the case of a single on demand e-mail sent by Salesforce®</i></p> <p><i>This CSV is only used if P_MAP_FILE is not defined.</i></p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
MailLinkFile	<emailing_links.csv>	<p>Set if necessary the file giving the definition of hypertext links in the e-mail (see documentation on the KSL Email Designer function)</p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
MailReturnMode	1	<p>Set this parameter to the value 1.</p> <p>Deselect the option "Show this parameter in Ksl Office"</p> <p><i>For information:</i></p> <ul style="list-style-type: none"> <li>- 1: for the creation of an e-mail sent by Salesforce</li> <li>- 2: for the creation of an e-mailing sent by KSL</li> </ul>

		<i>Server or asynchronously sent by an external e-mail send &amp; track provider</i>
MailTitle	default	<i>Parameter not used in the case of a single on demand e-mail sent by Salesforce®</i>  This parameter must have a value (for example "default")  Deselect the option "Show this parameter in Ksl Office"
MailUrl	Example: <a href="https://www.naelan.fr/ksl_images/">https://www.naelan.fr/ksl_images/</a>	Set the web directory Url used for storing images (see the KSL Email Designer documentation)  Deselect the option "Show this parameter in Ksl Office"
P_FILE	Example: <SF_email_Opportunity.xml>	<i>XML data stream passed by the KSL for Salesforce plug-in</i>  <i>This default value is used only during debugging tests</i>  Deselect the option "Show this parameter in Ksl Office"
P_LABEL		<i>Parameter not used in the case of a single on demand e-mail sent by Salesforce®</i>  Deselect the option "Show this parameter in Ksl Office"
language	Example: FR	Salesforce interface language passed through the KSL plug-in  Deselect the option "Show this parameter in Ksl Office"
P_MAP_FILE	Example: <SF_email_mapping.map>	Indicate the mapping file created for the template and corresponding to its XSD  Deselect the option "Show this parameter in Ksl Office"

#### g. Step 7 (optional): create other e-mail templates based on the same data

In many business contexts, it is necessary to create a family of e-mail templates that share the same graphic design and based on the same Salesforce objects and data.

In this case, it may be more efficient to create KSL e-mail services based on the same KSL e-mail template. The procedure is then the following.

- In KSL Studio:
  - Edit the document template created in step 5

- Reserve the template and edit the properties of the text zone group by right-clicking / Properties
  - In the window that appears, specify the expression MailContentGroup the Expression field of the dynamic group call
  - Validate, save and release the edition
- From the Content repository tab of the KSL Administration application,
    - Create as many groups of text boxes as desired body of e-mail (1 group per model of e-mail)
  - In KSL Admin:
    - Recompile the document template above
    - Copy the document service as many times as needed (as much service as the family's email template)
    - Give a description to each copied service with the name of the template you want in Salesforce
    - Edit the functional parameters of each copied service and set the MailContentGroup parameter specifying the name of the group to call (Service 1> Group 1, Service 2> Group 2, etc.)

#### Benefits:

Thanks to this setting:

- All e-mail templates share the same document template, the same document service, and the same style sheet
- Creating and editing an e-mail body is done directly from Salesforce
- Adding a template is done in seconds by duplicating a service

#### Example:

SF\_Email\_500px service is an example of service shared by the 3 e-mail templates implemented by the following 3 services:

- SF\_Email\_ContractConfirm\_500px
- SF\_Email\_ContractTermin\_500px
- SF\_Email\_TrainingConfirm\_500px

## 7.6. Create emailing templates

To create e-mailing templates, proceed as follows. This process is explained using the object Campaign but any other object can be used.

*Note: The previous chapter is dedicated to the particular case of on-demand single e-mail.*

### a. Step 1b: Generate the XSD and XML streams corresponding to the e-mailing

In Ksl Administration, generate and download the XSD schema corresponding to the data personalizing

the e-mailing (and if necessary an associated XML test data file). This XSD schema will be used by the KSL service(s) in charge of creating the HTML bodies of the e-mails.

Example:

- Suppose that the body of the emails to be generated has to include the data corresponding to the fields *Salutation*, *LastName* and *FirstName* of the targeted contacts and the *Campaign* name.
- Moreover, the E-mail address of each targeted contact must also be used by the e-mailing, since the KSL server is responsible for sending, or transmitting e-mails generated to an external send & track provider.

In this context, it is necessary to generate the XSD and XML files from the *Campaign* object which contains the *Description* element, but also from the *CampaignMember* sub-object which contains the elements *LastName*, *FirstName*, *Salutation* and *Email*.

Example of XML data corresponding to this XSD schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<Campaign>
  <Description>My emailing</Description>
  ...
  <CampaignMember>
    <LastName>MARTIN</LastName>
    <FirstName>John</FirstName>
    <Salutation>Mr</Salutation>
    <Email>martin@naelan.com</Email>
    ...
  </CampaignMember>
  <CampaignMember>
    <LastName>SIMONS</LastName>
    <FirstName>Marie</FirstName>
    <Salutation>Mrs</Salutation>
    <Email>simons@naelan.com</Email>
    ...
  </CampaignMember>
  ...
</CampaignMember>
```

## b. Step 2b: create a mapping file

The mapping file is used to create variables inserted in the body of an e-mailing:

- These variables can be selected and inserted by the designer or the content administrator in the body of each e-mailing template within text zones.
- These variables can be selected and inserted by the Salesforce user when he edits the body of the e-mailing

- These variables are automatically substituted during the e-mail bodies generation with data from the XML stream coming from Salesforce
- In KSL Admin, create a mapping file with a .map extension:
  - The mapping file is a text file
  - It defines each field used by an e-mail (field used in the subject, the body of the e-mail, or the recipient fields if it is an e-mailing)
  - Each field is defined by a line of the file

*Each line is composed of 3 strings separated by the character ";"*

- *The 1<sup>st</sup> string is the name of the field that will be used in the e-mail: if used in the template, this field will be the one displayed in the interactive personalization interface, otherwise the end user will be able to insert it in a text zone of the e-mail in this interface*
- *The 2<sup>nd</sup> string is the element of the XML stream that contains the value to display (single-child node)*
- *The 3<sup>rd</sup> string is the element of the XML, which is the of the element defined just above, that contains the value to display*

*Example with the <SF\_Email\_Mapping.map> mapping file:*

```
CAMPAGNE_NAME;Description;Campaign;
CONTACT_LASTNAME;LastName;CampaignMember;
CONTACT_FIRSTNAME;FirstName;CampaignMember;
CONTACT_SALUTATION;Salutation;CampaignMember;
Email;Email;CampaignMember;
```

Example:

*The XML stream and the mapping file given above will allow you to automatically substitute the e-mail variables with the Salesforce data with the following result:*

*For the 1<sup>st</sup> e-mail:*

- *CAMPAGNE\_NAME = My emailing*
- *CONTACT\_LASTNAME = MARTIN*
- *CONTACT\_FIRSTNAME = John*
- *CONTACT\_SALUTATION = Mr*
- *Email = martin@naelan.com*

*Pour le 2<sup>nd</sup> e-mail:*

- *CAMPAGNE\_NAME = My emailing*
- *CONTACT\_LASTNAME = SIMONS*
- *CONTACT\_FIRSTNAME = Marie*
- *CONTACT\_SALUTATION = Mrs*

- Email = *simons@naelan.com*

**Important notes:**

- Note that only the parent node of each element is to be defined
  - It is possible to define several fields that point to the same element
  - If one of the elements is iterative, the field takes the value of the element of the first iteration
  - Each line ends with a line break (including the last line)
  - In the case where the parent element exists several times in the flow, it is possible to specify the parent node of the parent node (example: SALUTATION\_CONTACT; Salutation; OpportunityContactRole.ContactId;)
- From the KSL Administration application, classify this mapping file in the KSL repository:
    - Select the Content repository tab
    - Add the mapping file as an external file and classify it in the tree structure

**c. Step 3b: Create a style sheet for your new e-mailing template**

If you already have a style sheet, go directly to the next step.

Otherwise, from KSL Studio (or KSL Office from KSL Server version 8.0.2.0), you can duplicate the FS\_SALESFORCE\_EMAIL style sheet and modify it to match your graphic charter.

**d. Step 4b: Create the body of your e-mailing**

From the Content repository tab of the KSL Administration application, create a group of text zones. This group will be the body of your e-mailing template.

This group of text zones contains one or more text zones that are also created from the Content Repository application. These text zones must use the style sheet created in the previous step 3.

The variables correspond to the Salesforce data are created in the text zone that require it.

- The value of each variable comes from the XML stream sent by Salesforce
- Each variable is in string format
- The name of each variable corresponds to the 1<sup>st</sup> element of the line of the mapping file that defines it

*For the example described above, the following variables will be created:*

- *CAMPAGNE\_NAME*
- *CONTACT\_LASTNAME*
- *CONTACT\_FIRSTNAME*
- *CONTACT\_SALUTATION*
- *Email*

### e. Step 5b: Create a new KSL document template for the e-mailing template

In this context of creating a new e-mailing template, it is necessary to create a new KSL document template from which a new KSL Server service will be created in step 6.

The procedure from KSL Studio is as follows:

- Add the XSD file generated in step 1 to the KSL project classification plan
- Duplicate the generic edition SF\_Emailing\_500px and add it to the KSL project classification tree
- In this new document template, select the style sheet by selecting the style sheet component created in step 3 above
- Modify the XML request for this document template by specifying the new XSD
- Delete the group in the document template
- Insert the new text zone group created in step 4
- Save and Update and release the document template

### f. Step 6b: Create the document service corresponding to the emailing template

- From KSL Admin, create the document service associated with the document template created in step 5
- Set a label in the description field: this description corresponds to the label displayed for this e-mail template in the list of templates presented to the Salesforce user in its Document Templates component.
- Classify this service in the project tree: the document service should be classified in a branch of the tree accessible to the Salesforce user in its Document Templates component accessible from the Campaign object.
- Specify the preview action group: EmailActions
- Fill in the parameters of this document service using the information below

Parameter	Default value or example	Description
feuillestyles (stylesheet)	FS_SALESFORCE_EMAIL	Select the style sheet used for editing (without extension)  Deselect the option "Show this parameter in Ksl Office"
MailAccount	MARKETING	Specify the e-mail distribution account if KSL Server is the SMTP server, otherwise left this field blank  This distribution account is used for mass e-mailing sending; it must match a distribution account created in KSL Admin  Deselect the option "Show this parameter in Ksl Office"



MailBat	True	<p>Set this parameter to True for this single on-demand e-mail sending case</p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
MailCharset	UTF-8	<p>Set this parameter to UTF-8</p> <p><i>Encoding of the CSV file generated from the XML stream sent by the KSL plug-in</i></p> <p><i>Set this parameter by default to the universal UTF-8 code page; it is possible to change this encoding (example: UTF-8, Windows-1252, ISO8859-15 (Linux))</i></p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
MailContentGroup	<p>Example:</p> <p>&lt;InvitationEvenement.ksg&gt;</p>	<p><i>Optional parameter for transmitting the group of text fields used as e-mail body</i></p> <p><i>If not specified, the group inserted by default in the document template is used</i></p> <p><i>This parameter requires that the document template dynamically calls the group (default group dynamically called with the expression MailContentGroup)</i></p> <p><i>The group is defined between the characters &lt; and &gt; with the extension .ksg</i></p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
MailCsvFile	<default>	<p>This file is used only if the P_MAP_FILE file is not defined.</p> <p>The default value can be used for debugging tests without XML stream. A value between the characters &lt; and &gt; indicates that the file is in the KSL repository.</p> <p>The CSV file is a text file with one contact / recipient per line, possibly with a header line that names the columns and column separators ";"</p> <p>You need to set a default value (for example "&lt;default&gt;") even if the P_MAP_FILE is defined</p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
MailCsvHeader		<p><i>Parameter not used in the case of an e-mailing</i></p> <p><i>This CSV is only used if P_MAP_FILE is not defined.</i></p> <p>Deselect the option "Show this parameter in Ksl Office"</p>

MailLinkFile	<emailing_links.csv>	<p>Set if necessary the file giving the definition of hypertext links in the e-mail (see documentation on the KSL Email Designer function)</p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
MailReturnMode	2	<p>Set this parameter to the value 2.</p> <p>Deselect the option "Show this parameter in Ksl Office"</p> <p><i>For information:</i></p> <ul style="list-style-type: none"> <li>- 1: for the creation of an e-mail sent by Salesforce</li> <li>- 2: for the creation of an e-mailing sent by KSL Server or asynchronously sent by an external e-mail send &amp; track provider</li> </ul>
MailTitle	<p>Example:</p> <p>&lt;CONTACT_SALUTATION&gt; &lt;CONTACT_LASTNAME&gt; - Visit our trade show</p>	<p>Select a text only if you want the subject of the e-mails to be predefined</p> <p>You can enter a free text in and insert some variables defined in the mapping file</p> <p>Select or deselect the option "Show this parameter in Ksl Office", depending on whether you want the user to be able to change or not, the subject used when creating e-mailing</p>
MailUrl	<p>Example:</p> <p><a href="https://www.naelan.fr/ksl_images/">https://www.naelan.fr/ksl_images/</a></p>	<p>Set the web directory Url used for storing images (see the KSL Email Designer documentation)</p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
P_FILE	<p>Example:</p> <p>&lt;SF_Emailing_Campaign.xml&gt;</p>	<p><i>XML data stream passed by the KSL for Salesforce plug-in</i></p> <p><i>This default value is used only during debugging tests</i></p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
P_LABEL		<p><i>Parameter not used in the case of a single on demand e-mail sent by Salesforce®</i></p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
language	Example: FR	<p>Salesforce interface language passed through the KSL plug-in</p> <p>Deselect the option "Show this parameter in Ksl Office"</p>
P_MAP_FILE	<p>Example:</p> <p>&lt;SF_emailing_mapping.map&gt;</p>	<p>Indicate the mapping file created for the template and corresponding to its XSD</p>

		Deselect the option "Show this parameter in Ksl Office"
--	--	---

**g. Step 7b (optional): create other e-mailing templates based on the same data**

In many business contexts, it is necessary to create a family of e-mailing templates that share the same graphic design and based on the same Salesforce objects and data.

In this case, it may be more efficient to create KSL e-mail services based on the same KSL e-mailing template. The procedure is then the following.

- In KSL Studio:
  - Edit the document template created in step 5
  - Reserve the template and edit the properties of the text zone group by right-clicking / Properties
  - In the window that appears, specify the expression MailContentGroup the Expression field of the dynamic group call
  - Validate, save and release the edition
- From the Content repository tab of the KSL Administration application,
  - Create as many groups of text boxes as desired body of e-mailing (1 group per model of e-mailing)
- In KSL Admin:
  - Recompile the document template above
  - Copy the document service as many times as needed (as much service as the family's email template)
  - Give a description to each copied service with the name of the template you want in Salesforce
  - Edit the functional parameters of each copied service and set the MailContentGroup parameter specifying the name of the group to call (Service 1> Group 1, Service 2> Group 2, etc.)

**Benefits:**

Thanks to this setting:

- All e-mailing templates share the same document template, the same document service, and the same style sheet
- Creating and editing an e-mailing body is done directly from Salesforce
- Adding a template is done in seconds by duplicating a service

## 7.7. Proposing an e-mail or e-mailing with alternative contents

KSL provides a powerful feature that allows to offer to users e-mail or e-mail templates with alternative content.

Thanks to this function, the user edits his e-mail or his e-mailing and can choose between several alternative text boxes in the KSL editor.

The SF\_Emailing\_500px service given as an example, implements this function and proposes in a single 4 e-mailings in a single template. The user selects the template "Family of e-mailings" and in a second time can select one of the 4 proposed contents corresponding to the groups:

- SF\_InvitationEvenement.ksg
- SF\_KSLEmailDesigner.ksg
- SF\_NewsletterNaelan.ksg
- SF\_Teasing\_Solutions.ksg

Refer to the KSL Office User Guide for creating these alternative groups or text boxes.

## 8. Annexes

### 8.1. Generate public and private keys

The Public Key is the identifier of the Ksl Office web application for authorization servers; it is the `clientId` configured for the KSL Suite application server (`web.xml` of the Tomcat configuration). This key is dependent on the instance of the KSL Suite server. The public key is required for the KSL server to recognize the Salesforce® server and to allow communication.

The Private Key is the equivalent of the `clientSecret`, also configured for the KSL Suite application server. This key is required for recognizing the JWT token (JSON Web Token). This 32-byte (256-bit) key corresponds to the "256" of the SHA256 protocol.

#### a. Private key creation

To generate this private key, we advise you to use the online Jamiekurtz tool: <http://jwtbuilder.jamiekurtz.com> (Signed JSON Web Token part - "Generate 32-byte Key" option).

Be careful, do not click on the Create Signed JWT #button, but on the arrow and select Generate 32-byte Key.

The value of the Key field displayed is the private key that can be copied and pasted into the KSL for Salesforce® configuration.

Previous form information is not needed to generate this key.



#### b. Public key creation

To generate this public key, we advise you to use the GuidGen online tool: <https://www.guidgen.com>.

Several other ways can be used to create this key:

- From Salesforce®
- With a tool like <https://www.ssh.com/ssh/putty/windows/puttygen>

- With a tool like <http://www.unit-conversion.info/texttools/random-string-generator/>
- With command lines (e.g. openssl)



NAELAN - Lyon  
4 rue Claude Chappe  
69370 Saint-Didier au Mont d'Or  
France  
Tél. +33 (0)4 37 59 81 40

NAELAN - Paris  
4 Place Louis Armand  
75023 Paris  
France  
Tél. +33 (0)1 72 76 80 86

[www.naelan.com](http://www.naelan.com)