

dox42 SalesForce integration

Documentation

Léon Emmer

Summary

This document explains the integration of dox42 into SalesForce.

Content

Summary	2
Content	2
Document details	2
Creating a connected SalesForce App	3
Creating a connected SalesForce App with a service user	3
Receive Security Token	1
Create a crypto data source in the dox42 office Add-In with your SF-credentials	5
Create a XML Data source in the dox42 office Add-In which is used to generate token for the SalesForce authentication	5
How to get your data from SalesForce	7
Create an Input Parameter for the Security Token in the dox42 Add-In	7
Set up a dox42 Call from SalesForce directly using Security Tokens from SalesForce	9
Set up SalesForce labels	9
Set up SalesForce "Named Credentials"	9
Create Visualforce Page10)
Set up SalesForce Action	3

Document details

- Version: dox42 SalesForce integration V 1.0
- Author: Léon Emmer
- Date: 29 August 2019

Creating a connected SalesForce App

If you want to read SalesForce data directly from the dox42 Add-Ins, e.g. for testing, you need to provide your credentials. You may pass them as input parameters in order to avoid storing them in the Data Map, or encrypt the credentials using a dox42 Crypto Data Source.

During the dox42 call directly from SalesForce a Security Token is passed to dox42, so these Credentials are NOT necessary. In this case the credentials are stored as named credentials encrypted in SalesForce.

Creating a connected SalesForce App with a service user

Click on the "settings" gear



Click on "New Connected App"



Configure your app as following

New Connected App		
		Save
Basic Information		
Connected App Name	ExampleName	
API Name	ExampleName	
Contact Email	Example@test.com	
Contact Phone	Example Number	
Logo Image URL®	exampleLogoURL	
Icon URL®	exampleIconURL	our sample logos
	Choose one of our sample logos	
Info URL	exampleInfoURL	
Description	Example Description	
 API (Enable OAuth Settings) 		
Enable OAuth Settings		
Enable for Device Flow	\sim	
Callback URL@	https://example.dox42.com	
Use digital signatures		
Selected OAuth Scopes	Available OAuth Seence	Salastad OAuth Saaaa
	None	Access and manage your Chatter data (chatter_api) Access and manage your Eclair data (eclair_api) Access and manage your Wave data (wave_api) Access and manage your data (api) Access custom permissions (custom_permissions) Access your basic information (id, profile, email, address, phone) Allow access to your unique identifier (openid) Full access (full) Perform requests on your behalf at any time (refresh_token, offline_access) Provide access to custom applications (visualforce)
ick on "Save": Save	mor Socrot"	
py consumer key and consu		
API (Enable OAuth Settings)		
Consumerikey 3MVG96		Consumer Secret Click to r

Receive Security Token

Click on your profile and on "Settings"



Click on "Reset My Security Token"

You (service user) will get an email with your "Security Token" which you need to copy as well





Create a crypto data source in the dox42 office Add-In with your SF-credentials

Create a new crypto data source with the following field names:

clientId = Consumer Key

clientSecret= Consumer Secret

Crypto D	ata Source				×
Name:		SFcredentials			
Master	Password:			Enc	yption Key:
		Show	Enter		Show
Fields	Encrypt Values	s Data Sources			
	Name		Data Field		Value
•	usemame				
	password				
	clientId				·····
	clientSecret				·····
	securityToken	1			
					Show Values
					V OK X Cancel

Create a XML Data source in the dox42 office Add-In which is used to generate token for the SalesForce authentication

Name: "TokenGetterSource"

Use the POST method

URL: https://login.salesforce.com/services/oauth2/token?"

POST-Data:

grant_type=password&client_id=<%SFcredentials.clientId%>&client_secret=<%SFcredentials.clientSe cret%>&username=<%SFcredentials.username%>&password=<%SFcredentials.password%><%SFcred entials.securityToken%>&format=xml

Element: OAuth

Fields	ds: Read		Tag/Attribute Name/XPath		Data Field Name	Complex Type			
i leius.	▶ SubElem		SubElement	\sim	access_	access_token		access_token	
		*		~					
	XM	L/JSON E	Data Source						×
	Nar	ne:	TokenGet	tterSourc	e				Http Header
	URL/XML/JSON: POST v https://ogin.salesforce.com/services/oauth2/token?								
	Post Data: grant_type=passw			e=passw	ord&client_id=<%SFcredentials.clientId			Insert Data Field	Username/Password
			Init						
	SharePoint MultiValue Field:						V Init from XSD/XML/JSON/WSDL		
	Field	l de s						1	
	Fie	IOS:		Read		Tag/Attribute Name/XPath		Data Field Name	Complex Type
			► S	SubEleme	ent 🗸	access_token		access_token	
	•			_	~				
					Test			\checkmark	OK 🗙 Cancel

How to get your data from SalesForce

Create an Input Parameter for the Security Token in the dox42 Add-In

Input Parameter		×
Input Parameter Name: Comment:	TokenGetter	^
		~
Input type:	Select from Data	a Source 🗸
	Data Source:	TokenGetterSource ~
	Value:	access_token ~
	Display Text:	access_token ~
Lookup Link:		
Caption Lookup Link:		
		V OK X Cancel

The SalesForce syntax is based on database queries that are implemented in the URL.

To access your data you need to know from which table you want to get the data as well as the field names.

You can look them up here: <u>https://developer.salesforce.com/docs/atlas.en-us.sfFieldRef.meta/sfFieldRef/salesforce_field_reference.htm</u>

Query statement: Functions like a SQL statement. The only difference is that between the operators spaces are replaced with "+"

Copy Code:

https://[YOUR SalesForce address].salesforce.com/services/data/v46.0/query?q=[Query statement]

In our example data comes from the table Order:

dox42 automate your documents, integrate your data

XML/JSON Data	Source							
Name:	GetOrd	ler				[Http Header
URL/XML/JSON:	DN: GET v https://YOUR.salesforce.com/services/data/v46.0/guery?g=SELECT+Id,OrderNumber,Ownerld,CreatedDate,AccountId,TotalAmount+FROM+On							otalAmount+FROM+Order
					Inse	rt Data Field	ι	Isername/Password
	Init							
				SharePoint	MultiValue Field:	~	Init fro	m XSD/XML/JSON/WSDL
Element:	item							
Fields							-	
ricius.	-	Read	_	Tag/Attribute Name/XPath	Data Field Name	Complex Type	_	
		SubElement	\sim	OrderNumber	Order Number		_	
		SubElement	\sim	Ownerld	Ownerld		- 1	
		SubElement	~	CreatedDate	CreatedDate			
		SubElement	\sim	AccountId	AccountId			
		SubElement	~	TotalAmount	TotalAmount			
		SubElement	~	ld	ld			
			~					
		Ter						Canaal
		Tes	st			\checkmark	JK	

Also important is to set your Http Header as following:

Header	Value	
Authorization	Bearer <%TokenGetter%>	

Set up a dox42 Call from SalesForce directly using Security Tokens from SalesForce

Set up SalesForce labels

Use Custom Labels to store Client_Id and Client Secret of your connected App.

Go to settings and search for "Custom Labels"

	Q Custom Labels
	User Interface Custom Labels
Click on "New Custom Label":	New Custom Label
Parameter	Value
Name	C_Client_Id
Description	Client-Id
Value	[your client Id]
Click on "Save & New":	3. New
Parameter	Value
Name	C_Client_Secret
Description	Client-Secret
Value	[your client Secret]

Click on "Save": Save

Set up SalesForce "Named Credentials"

Use Named credatials in SalesForce to store your Service User Credentials encrypted.

Go to settings and search for "Named Credentials"

		dox42	automate your documents, integrate your data
	Q Named Credentials		
	✓ Security Named Credentials		
Click on "New Named Credential":	New Named Credential		

In the setup use the following parameters (You need to use a Service User):

	Save
Label®	SalesForceLogin
Name 😡	Sales_Force_Login
URL	https://login.salesforce.com
▼ Authentication	
Certificate	
Identity Type	Named Principal V
Authentication Protocol	Password Authentication 🔻
Username	
Password	Service User
▼ Callout Options	
Generate Authorization Header ()	
Allow Merge Fields in HTTP Header ()	
Allow Merge Fields in HTTP Body	
	Save

Create Visualforce Page

Click on the gear:



Click on "Developer Console":



Setup for current app
🔅 Service Setup
Developer Console
Edit Page

Click on File >> New >> Apex Class

File	▪ Edit ▪ Debu	ig 🔹 Test 👻	Workspace •	Help 🕶	<	>
	New		Þ	Apex	Class	
	Open		CTRL+0	Apex	Trigg	er
	Open Resource	CTR	L+SHIFT+O	Visua	lforce	Page
	Open Lightning F	Resources CTR	L+SHIFT+A	Visua	lforce	Component
	Open Log		CTRL+G	Statio	Reso	urce
	Open Raw Log	CTR	L+SHIFT+G	Light	ning /	pplication
	Download Log	C	FRL+ALT+G	Light	ning (Component
	Save		CTRL+S	Light	ning 1	nterface
	Save All	CTR	L+SHIFT+S	Light	ning B	evnt
	Delete	СТ	RL+DELETE	Light	ning 1	okens

[Apex Class-Name can be chosen freely]

erDoo	x42Gett Coverage	ter.vfp 🕷 Ord	erGetter.apxc ×						
		INS	ERT C	ODE	HEF	RE			
<i>.</i>									
0									
1.									
2									
3									
4									
5									
6									
7									
8									
9									
0									
1									
2									
3									
4									
2									
7									
0									
9									
	Taste	Chastmoints	Owner Editor	Manu Etato	Broomers	Buchland			
	reats	circosponits	Courty Contor	Line	Problem	Trobicais			

Copy code

(replace all [] with your data, insert your parameters in the dox42 REST link with splitting the string at the correct position and adding your parameter out of SalesForce to it):

public with sharing class [Your chosen Apex Class name] {

//Get selected Order from VisualForce Page via Constructor

private [Name of table that is in use] o1;

public [Your chosen Apex Class name] (ApexPages.StandardController stdController) {

o1 = ([Name of table that is in use])stdController.getRecord();

}

//Method that is called by the VisualForce Page "popUp"

public PageReference redirectToDox42() {

//Client Id set in a Custom Label

String clientId = Label.C_Client_Id;

//Client Secret set in a Custom Label

String clientSecret = Label.C_Client_Secret;

//String where token is extracted to

String token;

//Setting up connection for requesting token

HttpRequest req = new HttpRequest();

req.setMethod('POST');

req.setEndpoint('callout:Sales_Force_Login/services/oauth2/token');

string bdy = 'grant_type=password' +

'&client_id=' + clientId +

'&client_secret=' + clientSecret +

'&username={!\$Credential.Username}'+

'&password={!\$Credential.Password}'+

'&format=xml';

req.setBody(bdy);

Http h = new Http();

HttpResponse res = h.send(req);

//Extracting token out of XML response

String storage = res.getBody();

storage = storage.substringAfter('<access_token>');

token = storage.substringBefore('</access_token>');

//Call dox42 Server with handing over Order-Id and token

String dox42Url= '[Your dox42 Rest Link]';

PageReference newFloodOnlyUrl = new PageReference(dox42Url);

return newFloodOnlyUrl;

}

Click on File >> New >> Visualforce Page

File 🕶 Edit 🕶 Debug 🕶	Test • Workspace •	Help • < >
New	ا	Apex Class
Open	CTRL+0	Apex Trigger
Open Resource	CTRL+SHIFT+0	Visualforce Page
Open Lightning Reso	urces CTRL+SHIFT+A	Visualforce Component
Open Log	CTRL+G	Static Resource
Open Raw Log	CTRL+SHIFT+G	Lightning Application
Download Log	CTRL+ALT+G	Lightning Component
Save	CTRL+S	Lightning Interface
Save All	CTRL+SHIFT+S	Lightning Event
Delete	CTRL+DELETE	Lightning Tokens
Close	CTRL+/	
Close All	CTRL+ALT+/	

[Visualforce Page-Name can be chosen freely]

Copy code:

<apex:page standardController="[Table you want to use e.g. Order]" extensions="[Name of your Apex class e.g. OrderGetter]" action="{!redirectToDox42}">

</apex:page>

Set up SalesForce Action

Click on "Object Manager" in the settings:



Then click on your label you wish to implement dox42:

(In this example "Order")



Image	Image
Individual	Individual
Lead	Lead
Macro	Macro
Messaging Session	MessagingSession
Messaging User	MessagingEndUser
Opportunity	Opportunity
Opportunity Product	OpportunityLineItem
Order	Order
Order Product	OrderItem
Price Book	Pricebook2
Price Book Entry	PricebookEntry
Product	Product2
Product Consumption Schedule	ProductConsumptionSchedule
Quick Text	QuickText
Recommendation	Recommendation
Scorecard	Scorecard
Scorecard Association	ScorecardAssociation
Scorecard Metric	ScorecardMetric
Social Persona	SocialPersona
Task	Task

Click on "Buttons, Links and Actions" and on "New Action":



Setup:

	Save	Cancel	
Object Name	Order i	Choo	se "Custom Visualforce"
Action Type	Create a Record 🔹		
Target Object	None 🔻 i		
Standard Label Type	None V i		
Label			
Name	i		
Description	i		
Create Feed Item	🖉 i		
Success Message		i	
Icon	✓ Change Icon		
	Save	Cancel	



automate your documents, integrate your data

Object Name	Order i Choose your Visualforce
Action Type	Custom Visualforce Page
Visualforce Page	OrderDox42Getter [OrderDox42Getter] 🔹
Height	250px Set a name for your button
Standard Label Type	None
Label	Autogenerated by pressing "tab"
Name	Ontional description
Description	
icon	Change icon
	Save
	Click on Save

Next go back to the label you want to edit and click on "Page Layouts", select your layout you would like to add the action to.

(e.g. "Order Layout")



Select "Mobile & Lightning Actions"

Drag and drop your created action into this field. (It is recommended to put it on first place.)



Click on Save.

dox42 automate your documents, integrate your data

		All 🔻	Q	Search Orders and n	iore			★ 🚽 🖽 ?
Sales Home	Opportunities 🗸	Leads 🗸 Tas	us ∨ I	Files 🗸 Account	s 🗸 Contacts 🗸	Campaigns 🗸	Dashboards 🗸	Reports 🗸 * 00000100 🗸
Order 00000100							Generate Docum	ent New Contact New O
Account Name dox42 Test	Contract Number	Order Start E 23.07.2019	ate	Status Activated	Order Amount 420.000,00 €			
			\ \ <i>₹</i>		J.M.C.M. 7 1114			
	~			\rangle		Activated		V Mark Stat
Related Det	✓ ails					Activated	Activity	V Mark Stat
Related Det	tails					Activated Edit Products	Activity New Task	✓ Mark Stat
Related Det	tails ts (3)	DE	QUANTITY) ,	UNIT PRICE	Activated	Activity New Task	✓ Mark Stat
Related Det Crder Produce PRODUCT SLA: Bronze	tails ts (3) PRODUCT COL SL9020	DE	QUANTITY 2,00	,	UNIT PRICE 10.000,00 €	Activated	Activity New Task Create a	✓ Mark Stat
Related Det Criteria Det Criter	tails ts (3) PRODUCT COI SL9020 KWW GC1060	DE E	QUANTITY 2,00	, ,	UNIT PRICE 10.000,00 € 100.000,00 €	Activated Edit Products	Activity New Task Create a	Mark Stat