How to Integrate DocuSign with Salesforce

What is DocuSign?

DocuSign is a company that provides electronic signature technology. This means you can send documents and get them signed online without printing or mailing.

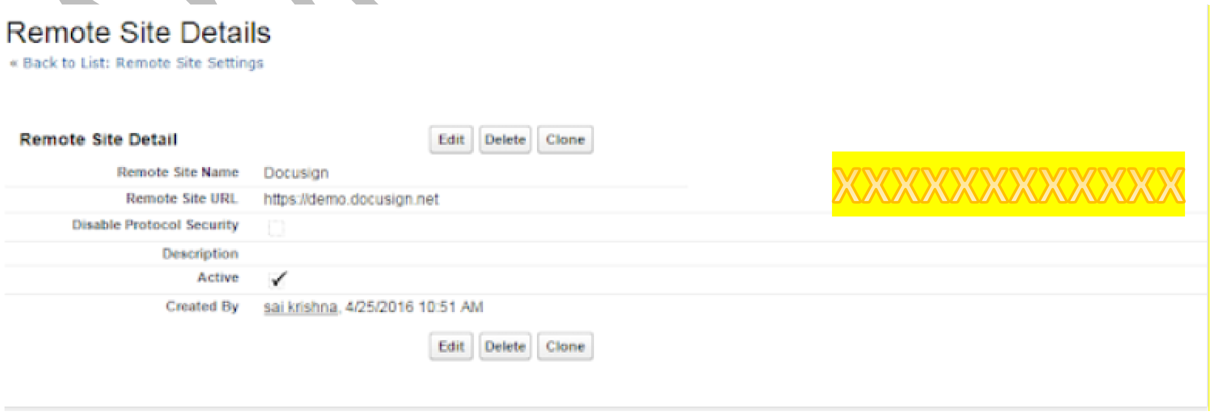
Why Use DocuSign?

Imagine a company in Australia needs to send Quote to a client named Tim. Normally, they would print the documents, mail them to Tim, and wait for him to sign and send them back. This takes a lot of time.

With DocuSign, you can send these documents electronically, and Tim can sign them online instantly. This saves a lot of time and effort.

Step 1: How to Set Up DocuSign with Salesforce

1. **Get a Salesforce Developer Account**:
   - If you don't have one, sign up for a Salesforce developer account.
2. **Create a DocuSign Sandbox Account**:
   - Sign up for a DocuSign sandbox account. https://developers.docusign.com/
3. **Log in to Salesforce Developer Account**:
   - Access your Salesforce developer account.
4. **Add DocuSign to Salesforce Authorized Endpoints**:
   - In Salesforce, go to `Setup`.
   - Click on `Security`.
   - Select `Remote Site Settings`.
   - Add **`https://demo.docusign.net`** as an authorized endpoint.



Remote Site Detail

Remote Site Name	Docusign
Remote Site URL	https://demo.docusign.net
Disable Protocol Security	
Description	
Active	✓
Created By	sai krishna, 4/25/2016 10:51 AM

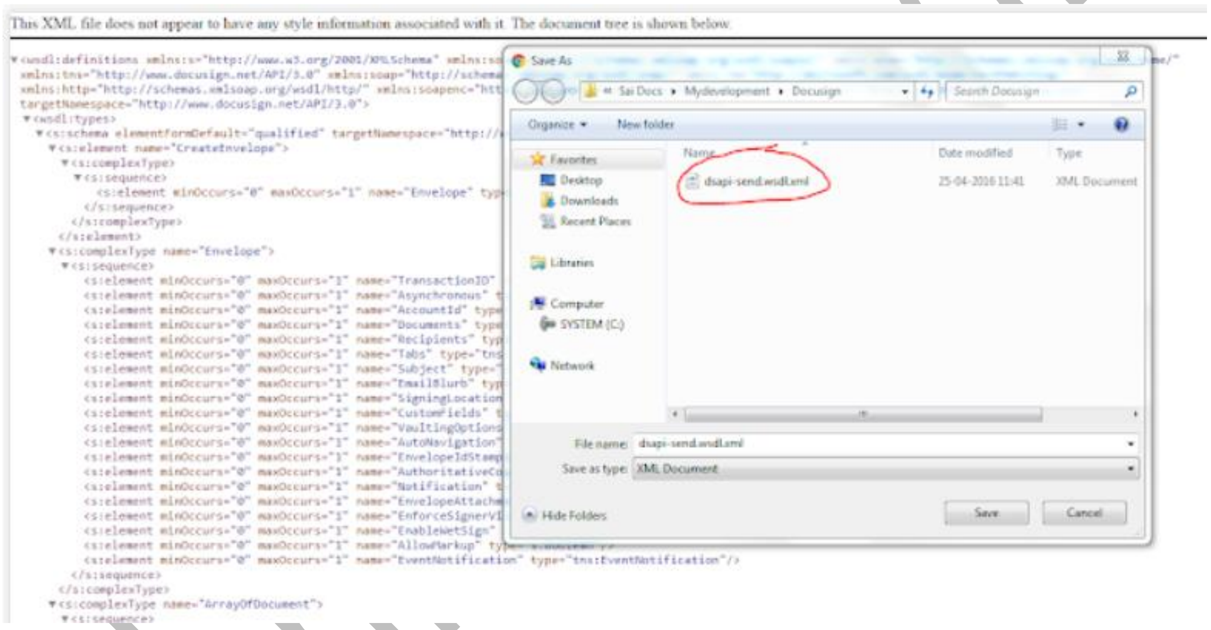
XXXXXXXXXXXX

Step 2: Creating an Apex Class to Access DocuSign API

Steps to Get the WSDL

1. **Download the WSDL File**:
    - Go to this link.
    - Right-click on the page.
    - Save the file to your desktop.

This WSDL file will be used to create an Apex class to interact with the DocuSign API.



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

```
▼<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:so
 xmlns:tns="http://www.docusign.net/API/3.0" xmlns:soap="http://schema
 xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:soapenc="htt
 targetNamespace="http://www.docusign.net/API/3.0">
  ▼<wsdl:types>
    ▼<s:schema elementFormDefault="qualified" targetNamespace="http://
      ▼<s:element name="CreateEnvelope">
        ▼<s:complexType>
          ▼<s:sequence>
              <s:element minOccurs="0" maxOccurs="1" name="Envelope" typ
            </s:sequence>
          </s:complexType>
        </s:element>
      ▼<s:complexType name="Envelope">
        ▼<s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="TransactionID"
            <s:element minOccurs="0" maxOccurs="1" name="Asynchronous" t
            <s:element minOccurs="0" maxOccurs="1" name="AccountId" type
            <s:element minOccurs="0" maxOccurs="1" name="Documents" type
            <s:element minOccurs="0" maxOccurs="1" name="Recipients" typ
            <s:element minOccurs="0" maxOccurs="1" name="Tabs" type="tns
            <s:element minOccurs="0" maxOccurs="1" name="Subject" type="
            <s:element minOccurs="0" maxOccurs="1" name="EmailBlurb" typ
            <s:element minOccurs="0" maxOccurs="1" name="SigningLocation
            <s:element minOccurs="0" maxOccurs="1" name="CustomFields" t
            <s:element minOccurs="0" maxOccurs="1" name="VaultingOptions
            <s:element minOccurs="0" maxOccurs="1" name="AutoNavigation"
            <s:element minOccurs="0" maxOccurs="1" name="EnvelopeIdStamp
            <s:element minOccurs="0" maxOccurs="1" name="AuthoritativeCo
            <s:element minOccurs="0" maxOccurs="1" name="Notification" t
            <s:element minOccurs="0" maxOccurs="1" name="EnvelopeAttache
            <s:element minOccurs="0" maxOccurs="1" name="EnforceSignerVi
            <s:element minOccurs="0" maxOccurs="1" name="EnableWetSign"
            <s:element minOccurs="0" maxOccurs="1" name="AllowMarkup" ty
            <s:element minOccurs="0" maxOccurs="1" name="EventNotification" type="tns:EventNotification"/>
          </s:sequence>
        </s:complexType>
      ▼<s:complexType name="ArrayOfDocument">
        ▼<s:sequence>
```

Save As

Sai Docs ▸ Mydevelopment ▸ Docusign

Name	Date modified	Type
dsapi-send.wsdl.xml	25-04-2016 11:41	XML Document

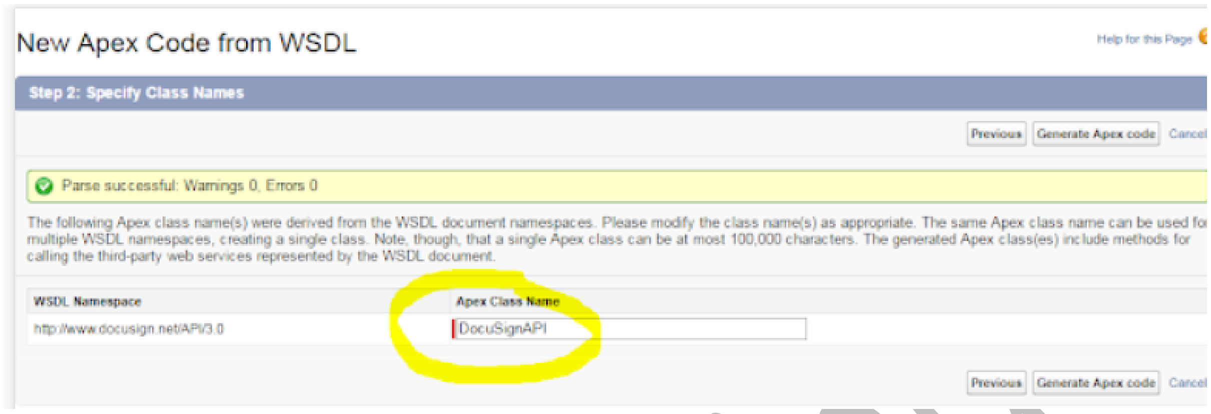
File name: dsapi-send.wsdl.xml

Save as type: XML Document

Generating an Apex Class from the WSDL

1. **Go to Apex Classes in Salesforce**:
    - Navigate to `Setup`.
    - Go to `Apex Classes`.
2. **Generate from WSDL**:
    - Click on "Generate from WSDL".
3. **Choose the WSDL File**:
    - Select the WSDL file you saved earlier.
    - Click on "Parse WSDL".
4. **Save the Class**:
    - Name the class "DocuSignAPI".
    - Click on "Save".

This will create an Apex class called "DocuSignAPI" that you can use to interact with the DocuSign API.


New Apex Code from WSDL

Step 2: Specify Class Names

Parse successful: Warnings 0, Errors 0

The following Apex class name(s) were derived from the WSDL document namespaces. Please modify the class name(s) as appropriate. The same Apex class name can be used for multiple WSDL namespaces, creating a single class. Note, though, that a single Apex class can be at most 100,000 characters. The generated Apex class(es) include methods for calling the third-party web services represented by the WSDL document.

WSDL Namespace	Apex Class Name
http://www.docusign.net/API/3.0	DocuSignAPI

Step 3: Creating a New Apex Class for DocuSign Integration

- **Generate the Apex Class**:
  - We have created an Apex class named "DocuSignAPI" using the WSDL provided by DocuSign.
- **Create a New Apex Class**:
  - Go to `Setup`.
  - Navigate to `Apex Classes`.
  - Click on "New".

  Name of Apex Class - SendToDocuSignController1

Paste the below code -

```
public class SendToDocuSignController1 {

    private final Quote quote;

    public String envelopeId { get; set; }

    private String accountId;

    private String userId;

    private String password;
    private String integratorsKey;

    private String webServiceUrl = 'https://demo.docusign.net/api/3.0/dsapi.asmx';



    public SendToDocuSignController1(ApexPages.StandardController controller) {

        this.quote = [SELECT Id, QuoteNumber, Contact__r.Email, Contact__r.FirstName,
Contact__r.LastName FROM Quote WHERE Id = :controller.getId() LIMIT 1];

        if (this.quote != null) {

            getUserDocuSignInfo();

            if (accountId != null && userId != null && password != null && integratorsKey !=
null) {

                envelopeId = 'Not sent yet';

                SendNow();

            } else {

                envelopeId = 'DocuSign credentials not configured for the user';

            }

        } else {

            envelopeId = 'Quote not found';

        }

    }



    private void getUserDocuSignInfo() {

        User currentUser = [SELECT Id, DocuSign_Account_Id__c, DocuSign_User_Id__c,
DocuSign_Password__c, DocuSign_Integrator_Key__c FROM User WHERE Id =
:UserInfo.getUserId() LIMIT 1];

        if (currentUser != null) {

            accountId = currentUser.DocuSign_Account_Id__c;

            userId = currentUser.DocuSign_User_Id__c;
            password = currentUser.DocuSign_Password__c;

            integratorsKey = currentUser.DocuSign_Integrator_Key__c;

        }

    }


    public void SendNow() {

        if (quote == null) {

            envelopeId = 'Quote not found';

            return;

        }




        ContentDocumentLink cdLink = [SELECT ContentDocumentId FROM
ContentDocumentLink WHERE LinkedEntityId = :quote.Id  LIMIT 1];

        if (cdLink == null) {

            envelopeId = 'Document not found';

            return;

        }



        ContentVersion cv = [SELECT Id, Title, VersionData FROM ContentVersion WHERE
ContentDocumentId = :cdLink.ContentDocumentId ORDER BY CreatedDate DESC LIMIT
1];



        if (cv == null) {

            envelopeId = 'Document not found';

            return;
```

```
}


DocuSignAPI.APIServiceSoap dsApiSend = new DocuSignAPI.APIServiceSoap();
dsApiSend.endpoint_x = webServiceUrl;


String auth = '<DocuSignCredentials><Username>'+ userId
    +'</Username><Password>' + password
    + '</Password><IntegratorKey>' + integratorsKey
    + '</IntegratorKey></DocuSignCredentials>';


dsApiSend.inputHttpHeaders_x = new Map<String, String>();
dsApiSend.inputHttpHeaders_x.put('X-DocuSign-Authentication', auth);


DocuSignAPI.Envelope envelope = new DocuSignAPI.Envelope();
envelope.Subject = 'Please Sign this Quote: ' + quote.QuoteNumber;
envelope.EmailBlurb = 'This is my new eSignature service,' +
    ' it allows me to get your signoff without having to fax, ' +
    'scan, retype, refile and wait forever';
envelope.AccountId  = accountId;


DocuSignAPI.Document document = new DocuSignAPI.Document();
document.ID = 1;
document.pdfBytes = EncodingUtil.base64Encode(cv.VersionData);
document.Name = cv.Title;
document.FileExtension = 'pdf';
```

```
envelope.Documents = new DocuSignAPI.ArrayOfDocument();

envelope.Documents.Document = new DocuSignAPI.Document[1];

envelope.Documents.Document[0] = document;


DocuSignAPI.Recipient recipient = new DocuSignAPI.Recipient();

recipient.ID = 1;

recipient.Type_x = 'Signer';

recipient.RoutingOrder = 1;

recipient.Email = quote.Contact__r.Email;

recipient.UserName = quote.Contact__r.FirstName + ' ' + quote.Contact__r.LastName;

recipient.RequireIDLookup = false;


envelope.Recipients = new DocuSignAPI.ArrayOfRecipient();

envelope.Recipients.Recipient = new DocuSignAPI.Recipient[1];

envelope.Recipients.Recipient[0] = recipient;


DocuSignAPI.Tab tab1 = new DocuSignAPI.Tab();

tab1.Type_x = 'SignHere';

tab1.RecipientID = 1;

tab1.DocumentID = 1;

tab1.AnchorTabItem = new DocuSignAPI.AnchorTab();

tab1.AnchorTabItem.AnchorTabString = 'By:';


DocuSignAPI.Tab tab2 = new DocuSignAPI.Tab();

tab2.Type_x = 'DateSigned';
```

```
        tab2.RecipientID = 1;

        tab2.DocumentID = 1;

        tab2.AnchorTabItem = new DocuSignAPI.AnchorTab();

        tab2.AnchorTabItem.AnchorTabString = 'Date Signed:';


        envelope.Tabs = new DocuSignAPI.ArrayOfTab();

        envelope.Tabs.Tab = new DocuSignAPI.Tab[2];

        envelope.Tabs.Tab[0] = tab1;

        envelope.Tabs.Tab[1] = tab2;


        try {

            DocuSignAPI.EnvelopeStatus es = dsApiSend.CreateAndSendEnvelope(envelope);

            if (es != null && es.EnvelopeID != null) {

                envelopeId = es.EnvelopeID;

            } else {

                envelopeId = null;

                ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR,
'Failed to get Envelope ID from DocuSign.'));

            }

        } catch (Exception e) {

            System.debug('Exception occurred while sending envelope: ' + e.getMessage());

            envelopeId = null;

            ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR,
'Failed to send the document to DocuSign. Please try again later.'));

        }

    }}
```

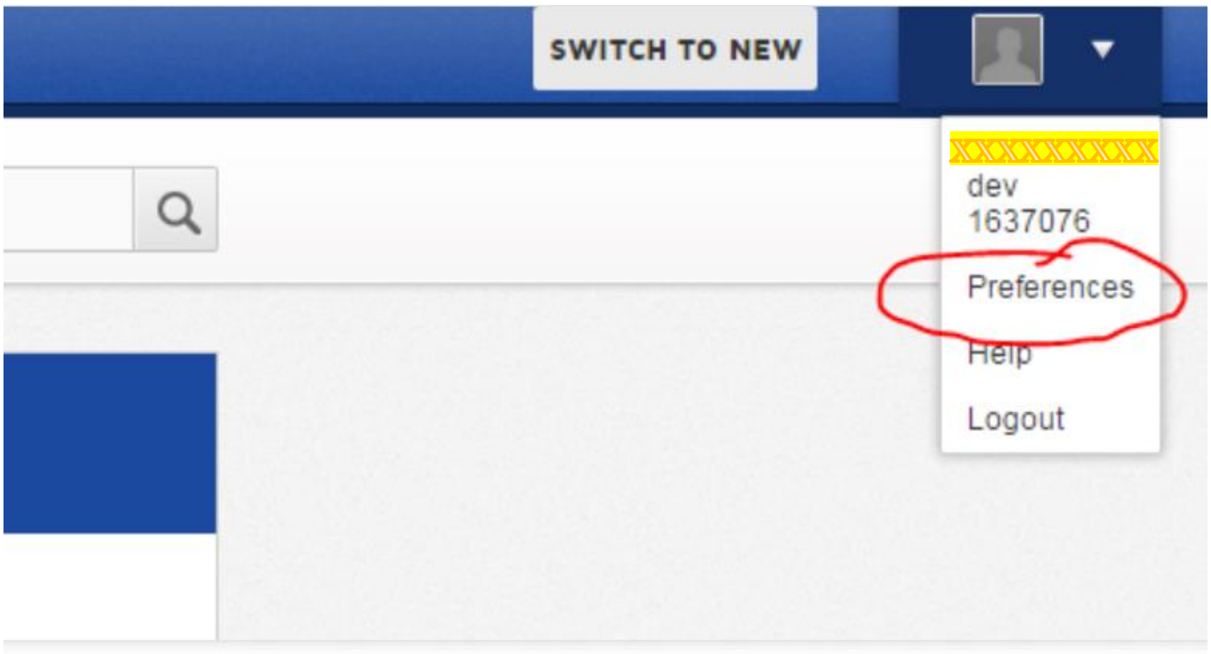
Note 1: Create a lookup relationship with the Contact object because the code above uses the contact email to send an email with the Quotes PDF attached.

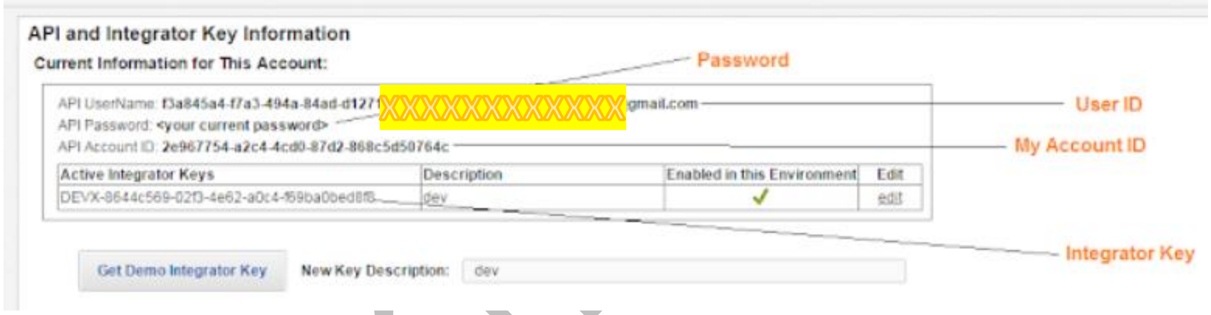
Note 2: In the User Object, add the fields DocuSignAccountId, DocuSignUserId, DocuSignPassword, and DocuSignIntegratorKey. Configure all these details for the current user so that the document is sent from the current user's DocuSign app.

Step 4: Generating Your DocuSign Integrator Key

1. **Log in to Your DocuSign Account**.
2. **Go to Preferences**:
   - Click on "Preferences".
3. **Generate Integrator Key**:
   - Scroll to the bottom and click on "API".
   - Generate your integrator key.

This key will be used to connect your Salesforce integration with DocuSign.





API and Integrator Key Information

Current Information for This Account:

API UserName: f3a845a4-f7a3-494a-84ad-d1271[redacted]gmail.com

API Password: <your current password>

API Account ID: 2e967754-a2c4-4cd0-87d2-868c5d50764c

Active Integrator Keys	Description	Enabled in this Environment	Edit
DEVX-8644c569-02f3-4e62-a0c4-f59ba0bed8f8	dev	✓	edit

New Key Description: dev

Step 5: Create a Visual Force Page to Call the Controller

We make name it as – CallDocusign1

```
<apex:page standardcontroller="Quote" extensions="SendToDocuSignController1">

<h1>Your eSignature request is being sent to DocuSign API!</h1>

<hr/>

<apex:form >

<apex:commandButton value="Send Again!" action="{!SendNow}"/>

</apex:form>
```

<hr/>

<strong>The DocuSign EnvelopeId:</strong>{!envelopeId}<br/>

</apex:page>

Step 6 : Creating a Custom Button on the Quote Object

1. **Go to Quote Object**:
   - Navigate to `Setup`.
   - Go to `Object Manager`.
   - Select `Quote`.
2. **Create a Custom Button**:
   - Click on `Buttons, Links, and Actions`.
   - Click `New Button or Link`.
   - Name the button (e.g., "Send Quote Through Docusign").
   - Set the display type to `Detail Page Button`.
   - Add the necessary URL or code for the action.

     /apex/CallDocusign1?id={!Quote.Id}

3. **Add the Button to the Quote Layout**:
   - Go to `Page Layouts`.
   - Select the layout you want to update.
   - Drag the new button to the desired location on the layout.
   - Save the layout.