README: An Embedded DocuSign Signing Experience

This readme provides a step-by-step example explaining how to set up a DocuSign ApexCode sample in your Salesforce.com account to accomplish an embedded signing experience with a single click of a button. When you are done with this walkthrough you will be able to sign a Salesforce.com Contract object with just one button click.

Before getting started you need to get a free Salesforce.com developer account at https://developer.force.com and a free DocuSign developer account at www.docusign.com/devcenter.

This walkthrough assumes familiarity with creating VisualForce pages and Apex controller classes. If Salesforce customization is new to you, starting with the walkthrough discussed in the Readme.pdf file in the Salesforce/SendToDocusign directory can help your learning curve.

1. Start out by adding DocuSign webservices to your authorized endpoints for your Salesforce.com developer account. To do this, go to Setup > Security > Remote Sites and add https://demo.docusign.net/api/3.0/dsapi.asmx.

2. Now we need to create three pages: **RenderContract**, for rendering a contract as a PDF, **EmbedDocusSign**, which makes a web service call to DocuSign and hosts the resulting document to be signed in an iFrame, and **embedPop**, for a return page when signing is complete that pops us back to the original contract page. Go to Setup->Develop->Pages, create the pages, and replace the default Visualforce Markup code with the code you'll find in the corresponding .page files in the src directory.

3. In this step we are going to create a class to access the DocuSign API. For this walkthrough we are just going to get the sending WSDL, which can be found at https://demo.docusign.net/api/3.0/Schema/dsapi-send.wsdl. You should save the WSDL file to your desktop. (Note: For complete set of WSDL files please refer to documentation on DocuSign DevCenter.)

   Next go to Setup->Develop-> Apex Classes and select “Generate from WSDL”. When the class generator asks you to supply a class name, we suggest that you overwrite the class name with **DocuSignAPI**.

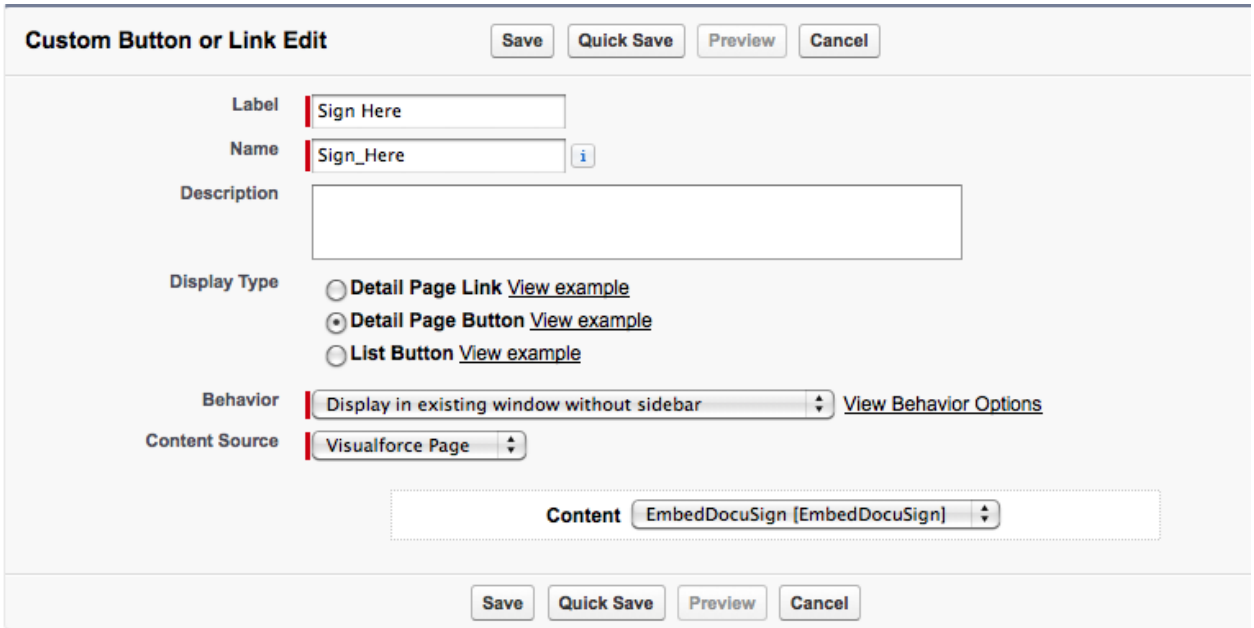
4. Using the new API endpoint, you can create a controller that manages the web service calls. Select "New" while on the Apex Classes pages. Copy the code from the EmbedToDocuSignController.cls file in the src directory into the Apex Class Edit box.

5. In order to use DocuSign API, you need to retrieve your DocuSign API credentials. Go to https://demo.docusign.net/, sign into your demo account, and then go to Preferences > API. There you will find these values that look something like this:

   Integrator's Key: **ZORO-a81ec71a-cb17-4af0-b1aa-9513115cbf02**
   API UserName: **0e212ae6-1e12-40c1-8f5a-a57458ccaa63**
   API Password: **<your current password>**
   API Account ID: **736e7948-6861-4ef3-ae71-4c56603dc14f**

   Plug these values into lines 9-12 of the controller class.

6. The last step is creating a custom button on the contract object that calls the sending page. Go to Setup > Customize > Contracts > Buttons and Links and create a **New** button. Fill in the fields like this:



7. Add the button to the page layout for the Contract object. This lets your users click on the button to get their Contracts electronically signed! They don't need to learn any other systems or follow complicated steps. The logic behind the button does all the work.