



Copper CRM Introduces DocuSign eSignature Integration

"DocuSign offered Ruby code examples that were incredibly easy to work with."

Daniel Quimper, Copper Developer

Copper, a Google Cloud–recommended CRM (Customer Relationship Management) solution for G Suite, launched a new breed of CRM in 2014. Copper (formerly ProsperWorks) is focused on enabling straightforward user scenarios for salespeople who just want a CRM platform that is simple, easy to use, and not cluttered with features they don't need or want. Copper is integrated with [G Suite from Google](#) and is growing rapidly in various markets. In addition to their DocuSign integration, they've also built integrations with popular applications such as [Slack](#), [RingCentral](#), [QuickBooks](#), [MailChimp](#), [ZenDesk](#), [Intercom](#), and more. Copper's focus on enabling critical sales processes without the need to involve an IT department is finding success in the crowded world of CRM tools and applications.

Copper and DocuSign: The benefits of a package deal

Sales people ultimately want to sell. Selling typically involves signing a contract. For Copper, already a [DocuSign partner](#), using DocuSign to send these contracts to customers is an obvious choice. So it came as no surprise when Copper started hearing directly from their customers that they would love it if DocuSign were available for them to use from within Copper. Many of Copper's clients were already customers of DocuSign and were using it in conjunction with Copper, but without a direct integration. Their request? A simple and easy way to send contracts and track contract status directly from their favorite CRM application.

Creating a DocuSign integration prototype

The task of working out the details of Copper's DocuSign integration fell to Daniel Quimper, a developer in one of Copper's engineering pods. He was involved early in the design of the DocuSign integration, as the product manager and the user

experience (UX) designer consulted with him on the technical capabilities of the DocuSign eSignature REST API. Daniel later led the development team building the integration directly into the Copper platform.

The Copper back-end code is mostly written in [Ruby on Rails](#), so Daniel decided to use DocuSign's free, open source [Ruby SDK](#). The Ruby SDK, a RubyGem package, was instrumental in helping Daniel get the integration from a conceptual prototype into a real feature of Copper very quickly. Daniel has worked to integrate many popular APIs into Copper, so it was great to hear him say "DocuSign offered Ruby code example that was incredibly easy to work with."

Daniel spent the next development cycle looking into the various authentication models that DocuSign supports. He decided to implement the [Authorization Code Grant](#) approach because of its enhanced security. Authorization Code Grant has an added benefit: when users see the DocuSign brand on their login page, they have the extra reassurance that their information is safe and secure.

Next, Daniel used the [Ruby code examples](#) that are also provided in the [DocuSign Developer Center](#) (see figure 1). Daniel first made sure he could run the example code the way it was provided by DocuSign. That was relatively straightforward and simple based on the instructions on the Developer Center. He then moved the sample code into the Copper codebase to be used as the backbone of the integration. As he notes, "First, I used a DocuSign code example and got it to work as-is. Once it was working separately, it was trivial to bring it into our codebase."

```

9
10 #!
11 def create
12   minimum_buffer_min = 3
13   envelope_id = session[:envelope_id]? session[:envelope_id] : nil
14   token_ok = check_token(minimum_buffer_min)
15
16   if token_ok
17     # 2. Call the worker method
18     # More data validation would be a good idea here
19     # Strip anything other than characters listed
20     args = {
21       :account_id => session['ds_account_id'],
22       :base_path => session['ds_base_path'],
23       :access_token => session['ds_access_token'],
24       :envelope_id => envelope_id,
25       :starting_view => request.params['starting_view'],
26       :ds_return_url => "#{Rails.application.config.app_url}/ds_common-return"
27     }
28     begin
29       results = worker args
30       redirect_to results['redirect_url!']
31     rescue DocuSign::ApiError => e
32       error = JSON.parse e.response_body
33       @error_code = error['errorCode']
34       @error_message = error['message']
35       render "ds_common/error"
36     end
37   else
38     flash[:messages] = 'Sorry, you need to re-authenticate.'
39     # We could store the parameters of the requested operation
40     # so it could be restarted automatically.
41     # But since it should be rare to have a token issue here,
42     # we'll make the user re-enter the form data after
43     # authentication.
44     redirect_to '/ds/mustAuthenticate'
45   end
46 end
47
48 # ***DS.snippet.0.start
49 def worker(args)
50   puts "\e[32m" + "args[:envelope_id]=#{(args[:envelope_id]).inspect}" + "\e[39m"
51
52   # Step 1. Create the NDSE view request object
  
```

Figure 1. Starting with the DocuSign Ruby code examples.

Soldering DocuSign onto Copper

Setting it up

Before customers can use Copper’s new DocuSign integration, they first need to enable and configure it. The integration can be enabled in the Copper Integrations page. There they will see the DocuSign tile shown in Figure 2.

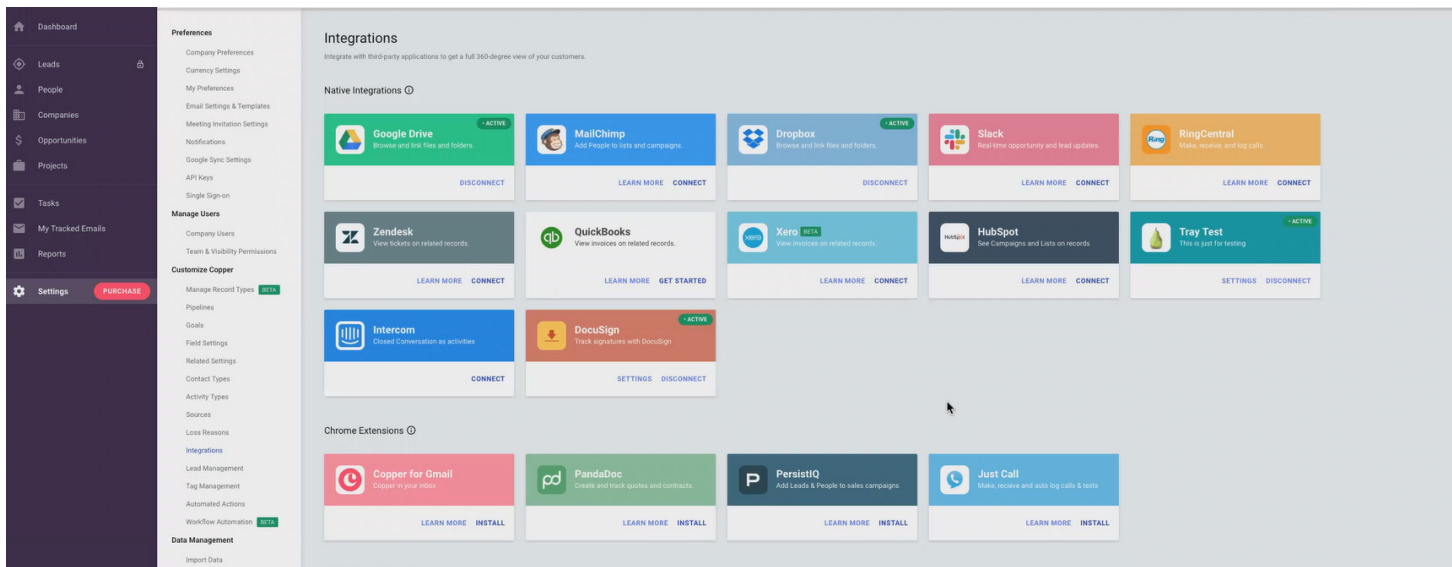


Figure 2. Configuring Copper integrations.

The important thing to decide is which types of Copper CRM records (such as Opportunities, Accounts, Leads, and more) will include the DocuSign UI pane and functionality. The users can also select the default recipients that will be automatically added to their DocuSign envelopes in the DocuSign Settings page, as shown in Figure 3.

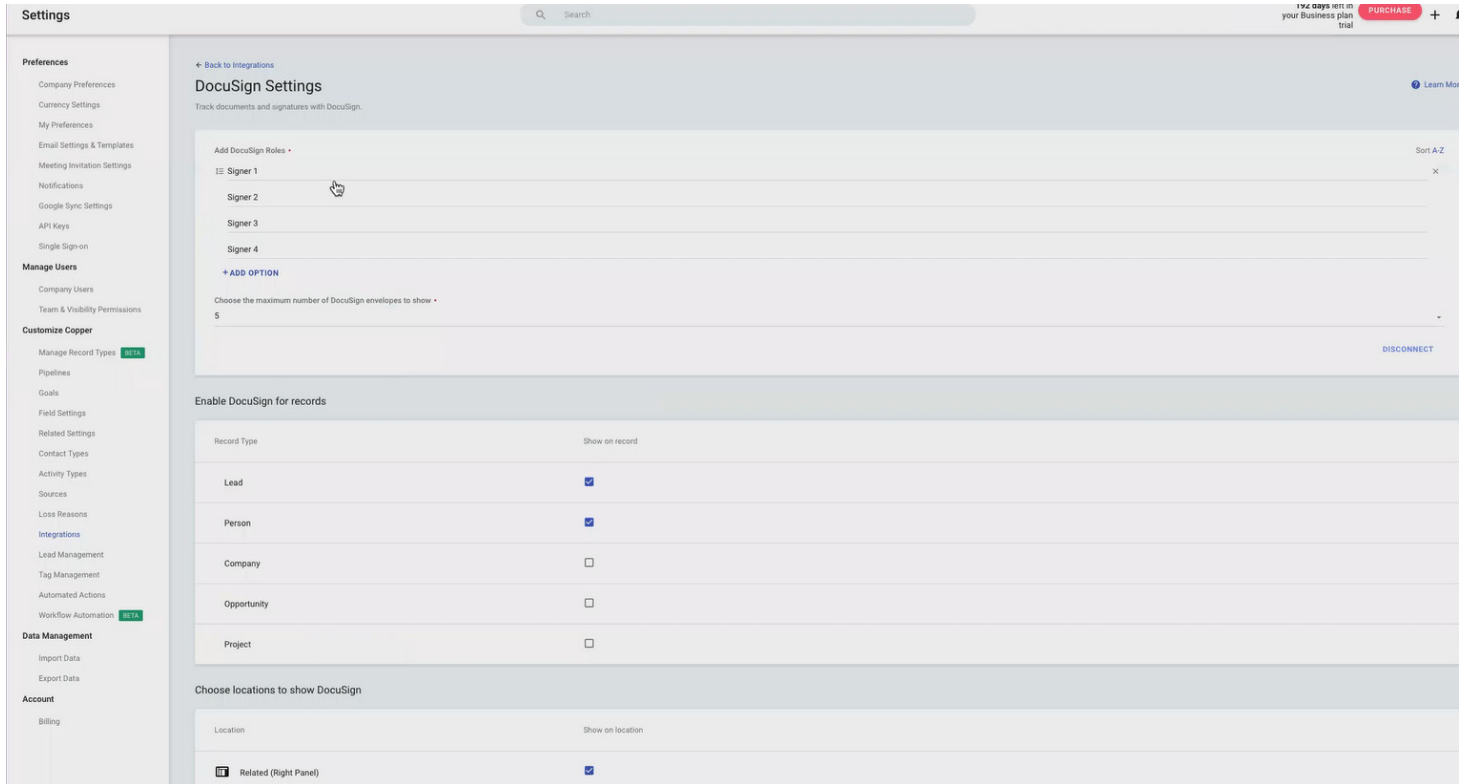


Figure 3. Configuring DocuSign settings in Copper.

Once the users enable and configure the DocuSign integration for desired CRM record types, they're ready to start using the DocuSign integration. When they open any record of the selected type inside the Copper app, users will now see a new DocuSign tab on the right hand side (see Figure 4) that includes the functionality of the DocuSign integration.

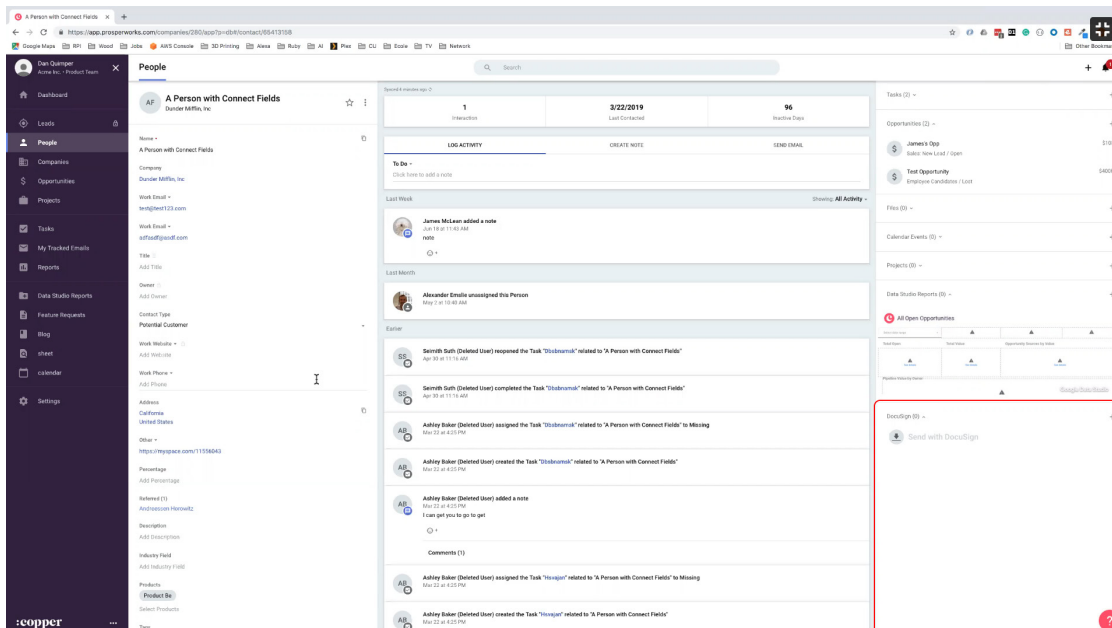


Figure 4. Accessing the DocuSign tab inside the Copper CRM record.

Logging in

Copper chose to implement DocuSign's Authorization Code Grant flow, users accessing the DocuSign integration will need to authenticate with DocuSign the first time they access the Copper app. Copper users will be redirected to the DocuSign login screen where they enter the credentials for their DocuSign account (or, if their organization uses single sign-on, they can use the organization's chosen identity provider). Once the users log in, they'll be redirected back to Copper. During the authentication process, the access token representing their identity is stored inside the Copper database together with a [refresh token](#) that allows Copper to obtain a new access token as needed. Copper users will never need to log into the DocuSign integration again, even if they use different devices or different browsers to access Copper.

Accessing DocuSign functionality

When Copper users click the + icon at the right of the DocuSign tab, they will see a modal pop-up dialog box (see Figure 5) that begins the process of sending a new envelope associated with the current Copper CRM record. This pop-up dialog lets users select the document(s) they wish to send for signature as well as the list of recipients (these are either signers or others who need to complete an action). These recipients are selected by default based on the DocuSign integration settings, but the users can still modify the list of recipients before they send the envelope. The users can either choose documents that are stored in the Copper CRM record, upload documents from their computer or device, or retrieve them from Google Drive.

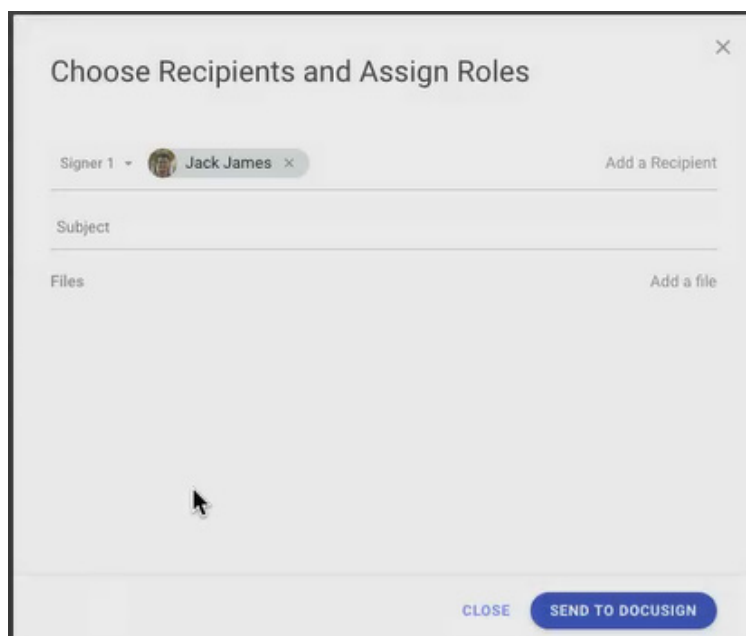


Figure 5. Sending a DocuSign envelope from inside Copper.

When Copper users click the SEND TO DOCUSIGN button (see Figure 5), they'll be redirected to DocuSign's [embedded sending](#) experience. This will let them either use an existing [template](#), if they have set up DocuSign [templates that match](#) the documents they've selected, or tag documents manually before sending them for signature. [Tagging a document](#) in DocuSign means selecting appropriate signing elements such as signatures, initials, dates, and so on to be acted upon by the recipients. When they've finished sending the envelope on the DocuSign side, they will be redirected back to their Copper CRM record.

Once the users have sent their documents using DocuSign in Copper, they'll see a new DocuSign status record added to the DocuSign tab (see Figure 6). That entry initially shows the envelope's status as Sent, if the envelope was successfully sent from DocuSign. The

record will later be updated to reflect applicable [envelope status changes](#). Copper maintains these envelope statuses for all Copper CRM records and shows the latest status in the DocuSign tab of the specific Copper CRM record, so users can get updates on their contracts.

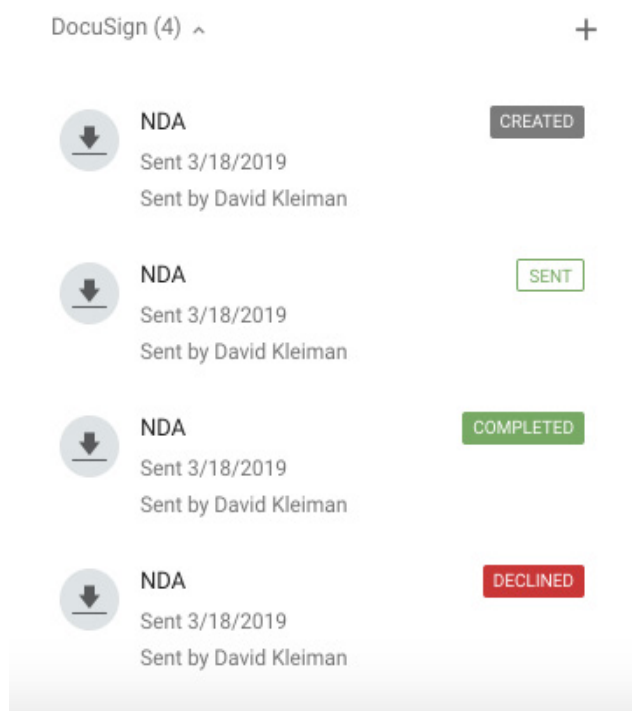


Figure 6. Checking the DocuSign status in Copper for a specific CRM record.

Connect webhook magic

A DocuSign envelope contains one or more documents and is sent to one or more recipients. Once all recipients finished signing the envelope, the envelope status is changed to Completed. How can the Copper integration know that an envelope is now completed? There are two ways to determine the status of a DocuSign envelope. The first is to make frequent API calls to check the status of every envelope. That is what we call polling. While this approach may work, it's inefficient, as it requires making many API calls for a single envelope status change.

The second method to determine the status of a DocuSign envelope is to use DocuSign's webhook notification technology, known as [DocuSign Connect](#). Connect enables web applications to receive notifications about the status of DocuSign envelopes. Daniel and his team quickly realized they needed to use DocuSign Connect so that they could update the status of envelopes inside Copper without taxing system resources with polling. Daniel researched how the DocuSign eSignature Ruby SDK makes API calls to create envelopes and what information is needed to have DocuSign call back to the Copper webhook with the required information about these envelopes. The result is a very sleek integration that dynamically updates information inside Copper about envelope status whenever an envelope changes. If the envelope is completed, the Copper integration adds a link back to the envelope so that Copper users can view the resulting final signed document. The decision to add a link back to DocuSign instead of attaching a file in Copper was made, in part, because Daniel and his team felt that the DocuSign documents are private and should not be copied over from DocuSign to Copper. Limited storage resources in the Copper application was another reason for this decision. The Copper team is aware that having a link may prevent users who do not have a DocuSign account from viewing the signed documents. The team may consider adding a feature that downloads the completed document back to Copper in a future release of the DocuSign integration.

“Extruding” future enhancements with Copper

The first version of the DocuSign integration for Copper was released in June 2019, and several customers began using it in its first month. The Copper product management team is planning to collect feedback from these customers to evaluate whether the DocuSign integration is meeting (or even exceeding) their expectations and what Copper may want to change in future versions. The Copper marketing team is also planning joint customer events with DocuSign to increase awareness of the integration so that more customers can use it and get the benefits.

One of the early ideas for enhancements in v2 is to enable customers to choose whether to get back the completed document stored inside their Copper CRM record, or keep using the link to DocuSign. That would enable non-DocuSign users to view the final signed contract, as the link back to DocuSign limits the capability to only DocuSign authenticated users of the account.

Other possibilities for enhancements may include building an integration with Copper workflow capabilities that enable documents to be sent based on predefined criteria by the users, without the need to send documents manually using the UI.

You can try the power and flexibility of the DocuSign eSignature API for yourself with a free Developer Sandbox. Just visit the DocuSign Developer Center and see for yourself why Copper chose the DocuSign eSignature API.

About DocuSign

DocuSign® is changing how business gets done by empowering anyone to send, sign and manage documents anytime, anywhere, on any device with trust and confidence. DocuSign and Go to keep life and business moving forward.

For U.S. inquiries: toll free **866.219.4318** | [DocuSign.com](https://www.docuSign.com)

For EMEA inquiries: phone **+44 203 714 4800** | email: emea@docuSign.com | [docuSign.co.uk](https://www.docuSign.co.uk)

For APAC inquiries: phone **+61 2 9392 1998** | email: apac@docuSign.com | [docuSign.com.au](https://www.docuSign.com.au)

Copyright © 2003-2019 DocuSign, Inc. All rights reserved. DocuSign, the DocuSign logo, “The Global Standard for Digital Transaction Management”,

“Close it in the Cloud”, SecureFields, Stick-eTabs, PowerForms, “The fastest way to get a signature”, The No-Paper logo, Smart Envelopes, SmartNav, “DocuSign It!”, “The World Works Better with DocuSign” and ForceFields are trademarks or registered trademarks of DocuSign, Inc. in the United States and/or other countries. All other trademarks and registered trademarks are the property of their respective holders.

