

DocuSign for Django (JWT Authentication method)

This package was developed to utilize e-contract/signatures directly within a Django web application. To be exact, demonstrating DocuSign integration with Django for embedded signing.

Project status

Django-DocuSign is under development. The project is not mature yet, but users can already use it as well as Django developers are able to develop this web application. It means that APIs and implementation may change (improve!), but the app works well right now.

I invite smart programmers to add more features to the app. So, help is welcome! Feel free to report issues, request features, or refactor them.

Next version

I suggest software developers add features like add PDF and DOC as a document to sign. it would be so easy, and there are samples in the DocuSign documentation. (there are not any differences in the process)

Features

- As you see the app has been designed in modular with MVT architecture, which means, it's capable of usage in other Django web applications. For example, you can use just the contract or account app in your Django web applications. Also, you can take advantage of its scalability and quickly develop it. It is worth noticing that, Forget password, change password, log in with users' email, and also simple Django Recaptcha embedded in the account app.

- Users can make a new contract, write their contract, and hit the confirm button. At this stage, the user will be authenticated by the JWT method and just need to grant consent, no need to register to DocuSign or have an account, in other words, the user must give consent to developers for authentication. However, users have to log in to your website.

- The second step is about the second signer, the user fills out the other signatory party data like name, email, and date.

- Now it is time for the user signature, user after signing comes back to your app and sees a success page, and their contract will immediately send to another signatory email.

- _ Eventually both of them have a PDF digital contract with two signatures in their email inboxes.

Django wrapper app for DocuSign embedded functionality (deep dive into the app)

To use the system, you must add your private data as a software developer. The app benefit from JWT authentication, using DocuSign API, so developers should have their own personal data including, Private key, public key, access token, etc.

Achieving all the above data just requires having a look at Docusign documents, but I just mention what you need to look for.

- | | |
|----------------------------|---|
| 1- private.key | (A py file including, private key and public key) |
| 2- ds_client_id | (The app's DocuSign Integration key) |
| 3- authorization_server | (account-d.docusign.com) |
| 4- ds_impersonated_user_id | (The User ID) |
| 5- 'account_id | (Account ID) |

Note: As I mentioned previously, detecting all confidential data is so straightforward. Log in for free on Docusign.com, make a quick app based on your platform(Django in this case), and finally achieve them. In addition, there is multiple videos on the DocuSign website that teach you how you are able to find them.

With the above secret data, we can get the below information.

- 1- Consent URL
- 2- api_account_id
- 3- access_token
- 4- base_path
- 5 - args
- 6 -signer client_id (ds_impersonated_user_id)

It would be Amazing. isn't it?

Installation

1. Run `pip install docusign-esign`
2. Run `pip install django-simple-captcha` based on the below link

<https://django-simple-captcha.readthedocs.io/en/latest/usage.html>

3. Run `python manage.py makemigrations` to create the migrations.
4. Run `python manage.py collectstatic` to transform static files from `asset` to `static_root`.
5. Run `python manage.py migrate` to create the app models.
6. Run `python manage.py createsuperuser` to create an admin username, email, and password to log in. (Note that you have to input the right email address because you will receive the contract in your email inbox, Don't Forget It)
7. Start the development server and visit <http://127.0.0.1:8000>

Thanks

Esfandiar Shiran