

DocFynd Injection Service – Document Validation Service

About

This document explains how to configure a document validation on document upload.

Introduction

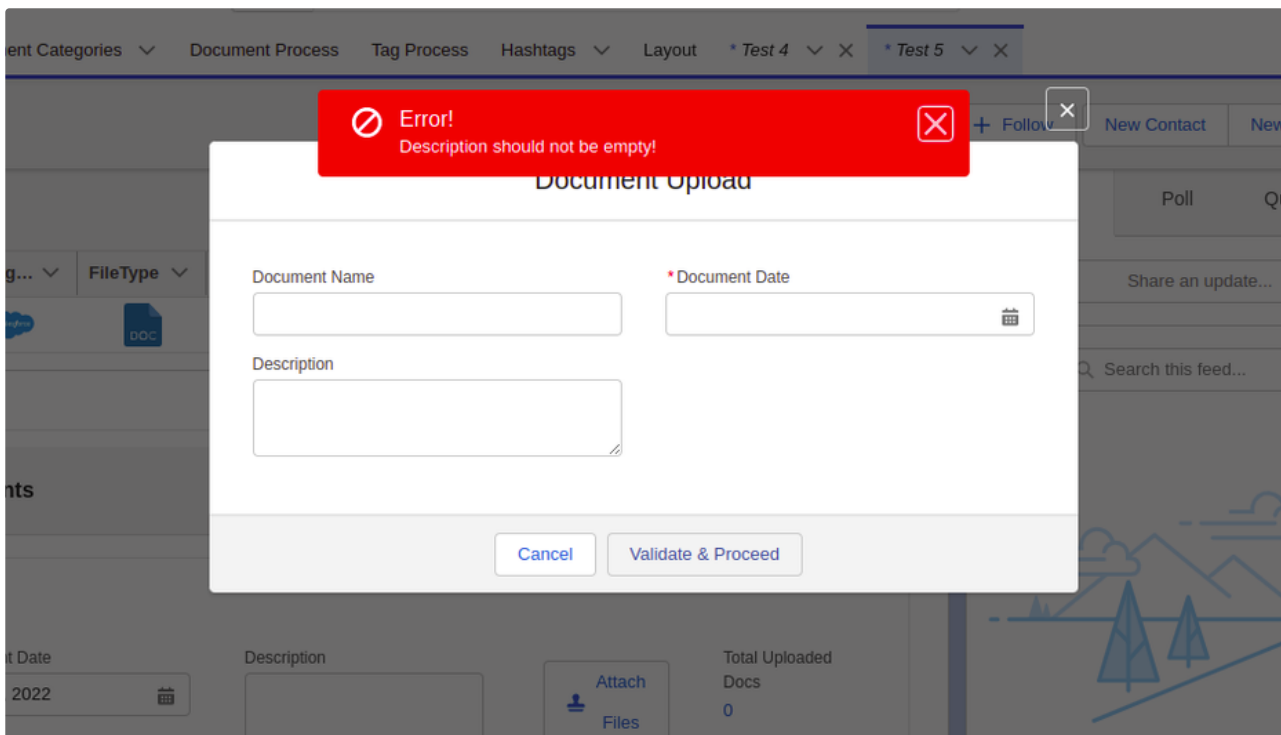
- DocFynd injection service serves the custom need of businesses. It provides an easy way for salesforce developers to change the behavior of the DocFynd application by adding custom logic via apex code. They can ask the developer/salesforce admin to write a custom code that will enable custom behavior.
- DocFynd uses a dependency injection service to enable this customization.

How does it work!

- We have used the salesforce callable interface which looks for some specific class names while executing the default functionality of our package.
- We have added callable interfaces to throw errors by writing custom logic for document validation. For e.g., You can write validation logic in the injection service class and throw errors.

Sample use case and implementation

Whenever the validate & proceed button is clicked with the document description blank, it should throw an error.



Steps to achieve this use case

1. Create an apex class named "DocFyndDocValidationService".
2. Copy and paste the below code into the apex class and save it.

```
1 global class DocFyndDocValidationService implements callable {
```

```

2  global object call(String action,Map<String , Object>args){
3      switch on action{
4          when 'validation'{
5              try{
6                  docfynd__Document__c doc = (docfynd__Document__c)args.get('document');
7                  // You can write your custom logic
8                  if(doc.docfynd__Description__c == null){
9                      throw docfyndException('Description should not be empty!');
10                 }
11             } catch(Exception e){
12                 throw docfyndException(e.getMessage());
13             }
14         }
15     }
16     return null;
17 }
18
19 private static AuraHandledException docfyndException(String message) {
20     AuraHandledException e = new AuraHandledException(message);
21     e.setMessage(message);
22     return e;
23 }
24
25 }

```

✔ Custom validation service implemented successfully