

DocFynd Injection Service – Upload Service

About

This document explains how to upload files to a custom drive using the DocFyndUpload injection service.

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 upload files to your custom drives.

Sample use cases and implementations

- DocFynd provides integration with Box, Google Drive, Dropbox, and AWS. So that you can directly upload files to any drives from DocFynd.
- Other than inbuilt integration from DocFynd you can also upload files to your own custom drives by using our injection service.

Sample Apex Class

```
1 global class DocFyndUploadService implements callable{
2     global object call(String action,Map<String , object>args){
3         docfynd.InjectionServiceRes injectionServiceRes = new docfynd.InjectionServiceRes();
4         List<Map<String,Object>> folderList = new List<Map<String,Object>>();
5         switch on action{
6             when 'getFolders'{
7                 System.debug('args'+args);
8                 Map<String, Object> objectMap = new map<String, Object>();
9                 objectMap.put('label', 'FolderName');
10                objectMap.put('value', 'FolderId');
11                folderList.add(objectMap);
12
13                injectionServiceRes.response = folderList;
14                injectionServiceRes.isActionExists = true;
15                return injectionServiceRes;
16            }
17            when 'uploadDocument'{
18                System.debug('args'+args);
19                String fileId = 'testFile123';
20                injectionServiceRes.response = fileId;
21                injectionServiceRes.isActionExists = true;
22                return injectionServiceRes;
23            }
24        }
25    }
26 }
```

```

24         when 'getDocumentUrl'{
25             System.debug('args'+args);
26             String fileUrl = 'https://google.com';
27             injectionServiceRes.response = fileUrl;
28             injectionServiceRes.isActionExists = true;
29             return injectionServiceRes;
30         }
31         when else{
32             injectionServiceRes.isActionExists = false;
33             return injectionServiceRes;
34         }
35     }
36 }
37
38 }

```

Details of the apex class

- While executing its default functionality, DocFynd looks for the Apex class named '*DocFyndUploadService*' and the action named 'getFolders' and 'uploadDocument' and 'getDocumentUrl'.
- Apex class and the action name needs to be exactly as mentioned and will not work if altered.
- Each time any of DocFynd interfaces tries to load the upload document, this class will be instantiated.

Action – getFolders

- This action is to get the folder list from the respective drive.
- The getFolders action should return the list of folders from your external drive.
- It should return as follows. For e.g., [{label: 'Folder Name' , value: 'Folder Id'}]

✓ The folders should be listed in the combo-box successfully.

Action – uploadDocument

- This action is to upload files to the respective drive.
- In uploadDocument action, the file name, file size, base64 will be passed as arguments.
- You can get the file body from base 64 and upload the file to your respective drive and return the fileId.

✓ The file should be uploaded successfully.

Action – getDocumentUrl

- This action is to view the file after upload.
- In this action, the file id will be passed as arguments.
- You can get the file Id and return us the Url of the file which needs to be viewed.

✓ The file should be viewed successfully.