



---

# Requester Specification

Version 1.1

---

June 2016

A Digital India Initiative  
National e-Governance Division.  
Department of Electronics and Information Technology.

## Revision History

Version	Date	Comments
1.0	12/08/2015	Version 1.0 Released.
1.1	16/06/2016	Added support for a smaller Fetch from Locker button.

## Table of Contents

Revision History.....	1
Introduction.....	3
Requester Workflow .....	3
Step 1: User clicks “Fetch from DigiLocker” button on the client application.....	3
Step 2: DigiLocker Requestor Widget pops up and prompts user to login. ....	3
Step 3: User logs in to the account and selects a file to share.....	4
Step 4: DigiLocker shares file with the client application and shows status message.....	4
Steps to Integrate Requester Workflow .....	5
Step 1: Register as Requestor on DigiLocker.....	5
Step 2: Register Requestor Application and Domain on DigiLocker.....	5
Step 3: Integrate DigiLocker Requestor Widget in your web application .....	6
Integrating with Digital Locker.....	6
REQUESTER REGISTRATION .....	6
TRIGGERING DIGILOCKER FROM YOUR WEB PAGE .....	7
HANDLING THE RESPONSE .....	8

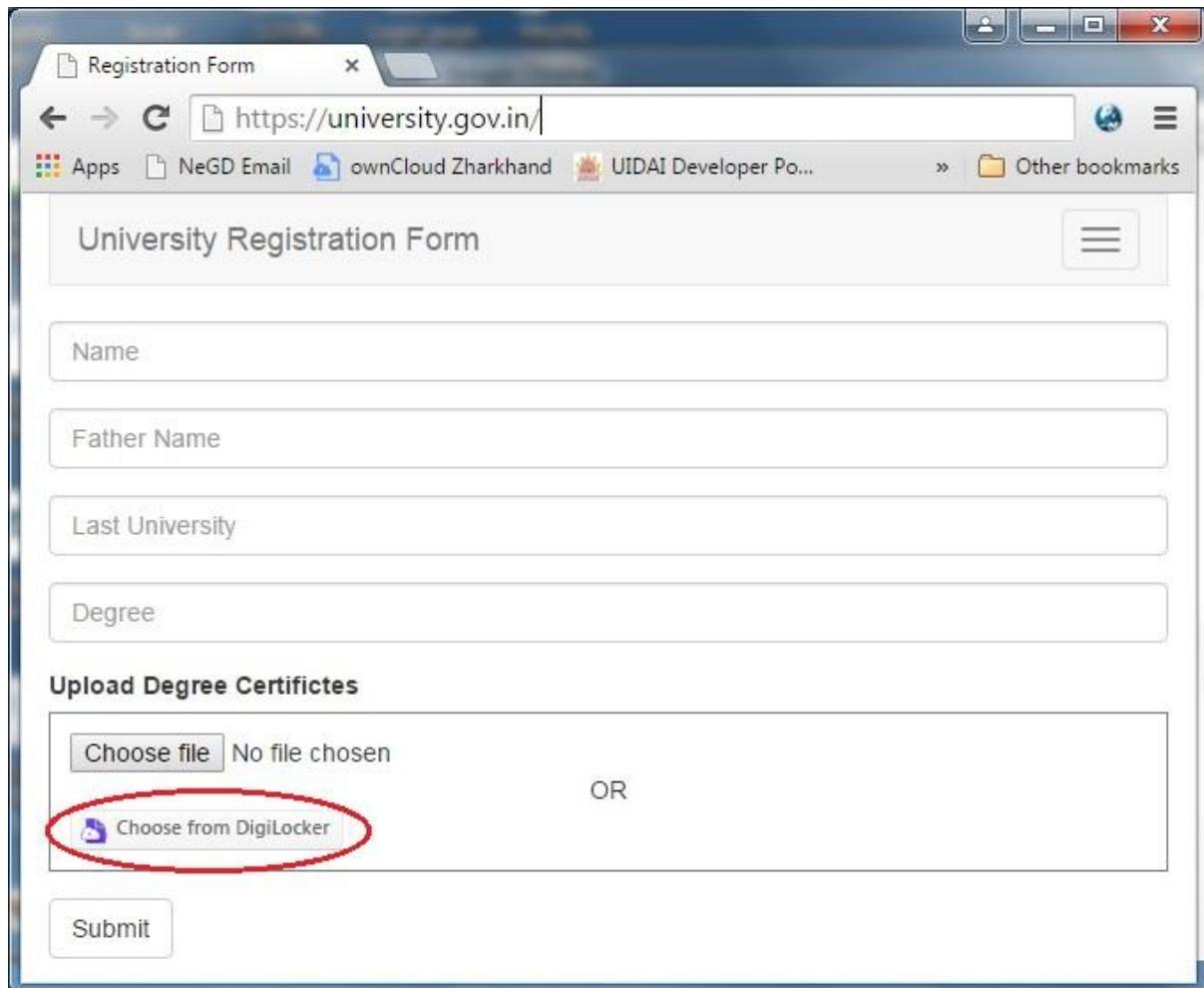
# Digital Locker Requester Specification

## Introduction

This document provides detailed specification to integrate with Digital Locker as a Requester. The steps described in this document will be used by various departments to access users' documents from their Digital Locker. This document assumes that the reader is aware of Digital Locker application functionality.

## Requester Workflow

**Step 1: User clicks “Fetch from DigiLocker” button on the client application.**

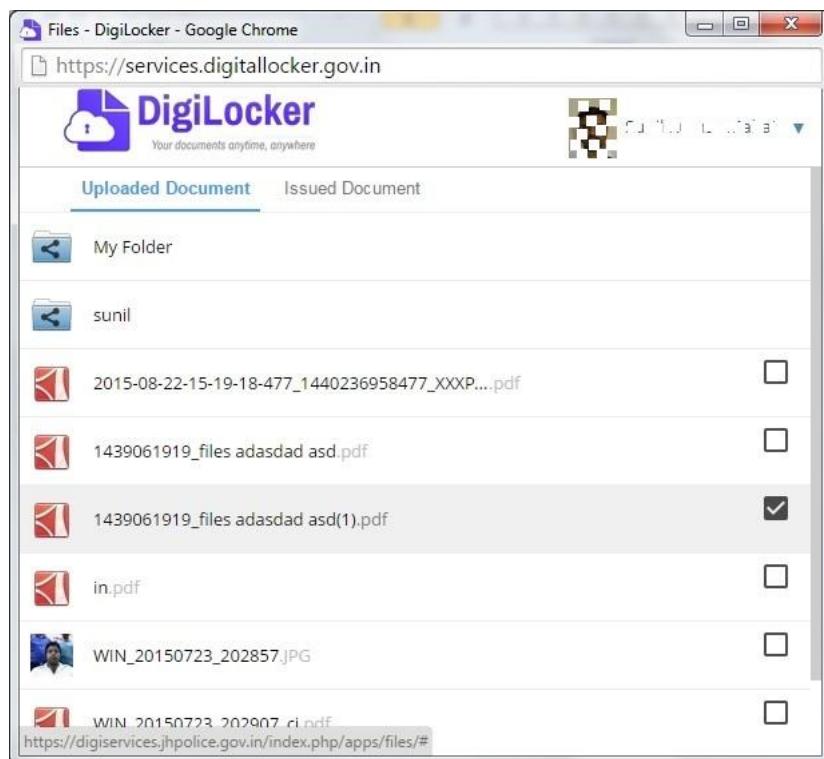


**Step 2: DigiLocker Requestor Widget pops up and prompts user to login.**

## Requester Specification



**Step 3: User logs in to the account and selects a file to share.**



**Step 4: DigiLocker shares file with the client application and shows status message.**

## Requester Specification

Registration Form <https://services.digilocker.gov.in>

University Registration Form

Name

Father Name

Last University

Degree

Upload Degree Certificates

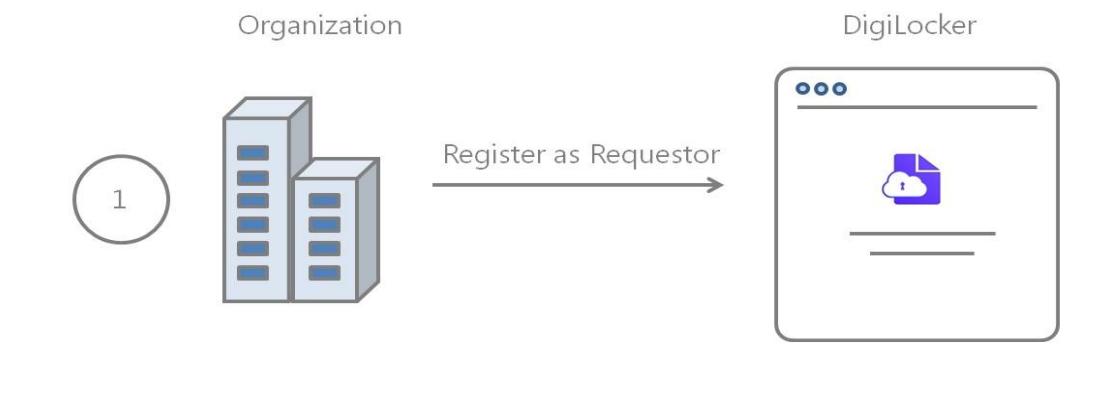
Choose file No file chosen

Choose from DigiLocker  Sharing Successful

Submit

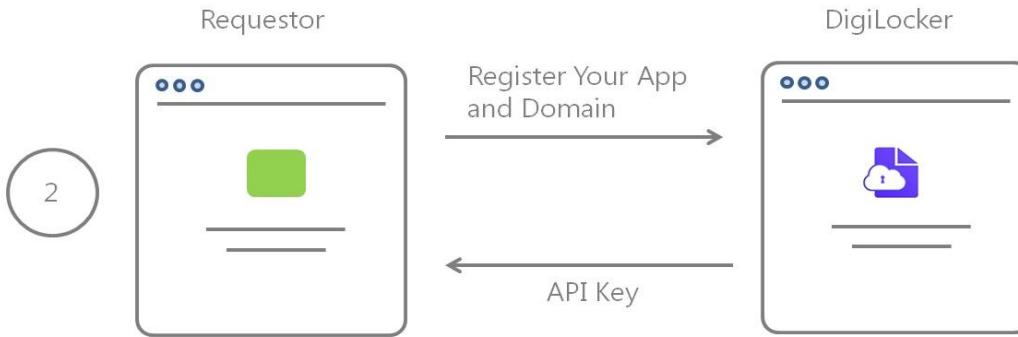
## Steps to Integrate Requester Workflow

### Step 1: Register as Requestor on DigiLocker

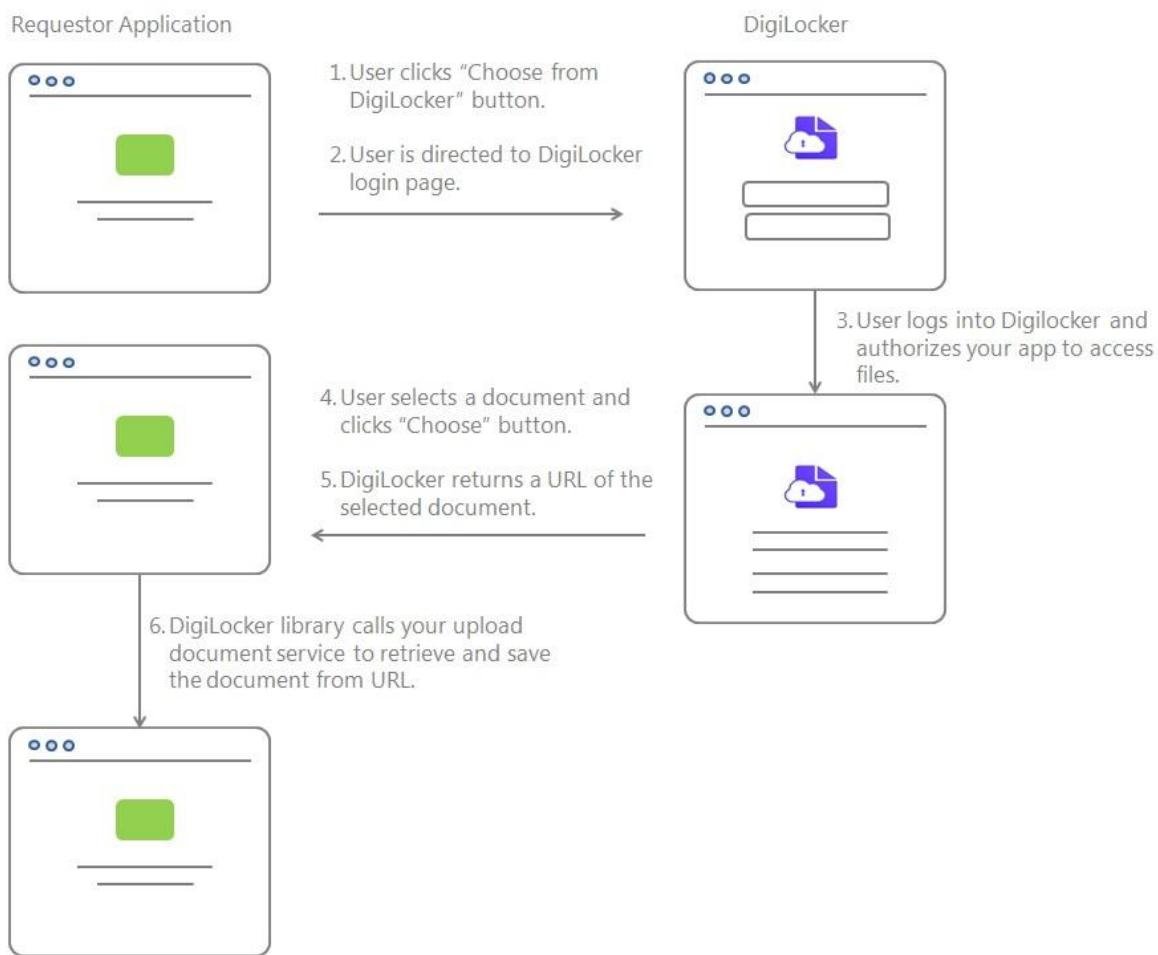


### Step 2: Register Requestor Application and Domain on DigiLocker

## Requester Specification



### Step 3: Integrate DigiLocker Requestor Widget in your web application



## Integrating with Digital Locker

### REQUESTER REGISTRATION

The first step in integrating with Digital Locker application is to register on Digital Locker as a Requester. On successful registration, the Requester application must be registered along with the domain of your application from which the requests to Digital Locker will be initiated. During the registration process you create an application id (app id) for your

## Requester Specification

application. You are provided with an api key which is a secret key. The App Id and the api key are used to authenticate the request from your application.

### TRIGGERING DIGILOCKER FROM YOUR WEB PAGE

DigiLocker provides a Requestor Widget to share files from DigiLocker. This widget can be launched with a button provided by DigiLocker Javascript library. The button launches DigiLocker application login page in a popup. A user can login to his/her DigiLocker account to select a file that the user want to share with the requestor application. This button looks like this -



Once you register as a Requester and register your application with Digital Locker, add following JavaScript snippet to your web page.

```
<script src="jquery.min.js"></script>

<script type="text/javascript"
src="https://services.digitallocker.gov.in/requester/api/1/dl.js"
id="dlshare" data-app-id="YOUR_APP_ID" data-app-hash="YOUR_APP_HASH"
time-stamp="TIMESTAMP" data-upload-url="YOUR_UPLOAD_URL">
</script>
```

Please note that the DigiLocker library uses jQuery. So please download jquery.min.js file from jQuery.com and place it on your web server. Specify the path of this file in the above snippet in the **src** parameter of jQuery script.

Provide app id that was provided during the application registration process in **data-app-id** parameter.

Provide SHA-256 encrypted value of your app id, app key and the timestamp values in this sequence in **data-app-hash** parameter.

Provide the url of the service that uploads the file from the public url of the document provided by Digital Locker in **data-upload-url** parameter. Please referto Handling The Response section below for more details.

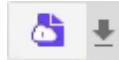
Provide a timestamp value in IST time zone in **time-stamp** parameter. This timestamp value must not be older than 30 minutes.

To add the “Fetch from DigiLocker” button to your page, add following line to your web page where you want to add the button. Provide a unique **id** for every instance of the button on your web page. The value of **id** can be used to identify the type of document being shared by a user.

```
<div class="share_fm_dl" id="attachment_poi"></div>
```

## Requester Specification

A smaller “Fetch from DigiLocker” button is also available if you prefer a smaller size of the button that suits your website. The smaller button looks like this –



To add the smaller button to your page, add following line to your web page where you want to add the button. Please note that you need to use `share_fm_dl_sm` class for this button.

```
<div class="share_fm_dl_sm" id="attachment_poi"></div>
```

If the integration is successful, the DigiLocker Requestor Widget will be launched and you will see the DigiLocker login page. If there is an error in the integration, the widget will show an authentication error and an error code. Following table lists the error codes.

Code	Description
203	Invalid app hash. One of app id, app key or timestamp is incorrect.
204	Invalid app id.
205	The app domain is not registered.
206	The timestamp is older than 30 minutes.

## HANDLING THE RESPONSE

DigiLocker workflow returns a public url of the document selected by the user from his/her account. The client application must retrieve the document from this url and save it at a secure location. The client application must provide an http service to extract and save the file from the url. The javascript library automatically calls this http service once the DigiLocker service returns the url. The service must comply with following specification.

### DESCRIPTION

This service will be provided by the client application over https.

### URL STRUCTURE

```
https://clientapp.gov.in/upload
```

**METHOD** POST

### PARAMETERS

- **user\_ref** (*required*) DigiLocker library passes a unique reference that can be associated with this instance of the client web page. This can be used to associate the file with the user who is sharing the file.

## Requester Specification

- **doc\_id** (*required*) This is the id value of the “Fetch from DigiLocker” button. This can be associated with the type of document the user is uploading such as proof of identity (poi) or proof of address (poa).
- **doc\_url** (*required*) This is the public url of the document shared by the user from DigiLocker.

### FORMAT

Your service should accept the above parameter in JSON format from POST method. The service should extract the url of the shared document from doc\_url parameter and read the document from the URL.

The parameters will be sent in following format -

```
{  
    'user_ref' : YOUR_USER_REF,  
    'doc_id'   : YOUR_DOC_ID,  
    'doc_url'  : DOCUMENT_URL  
}
```

### RETURN

The service should return ‘SUCCESS’ if the file is successfully stored or ‘FAILURE’ in case of failure.