**How to Store a plain text document and retrieve a document?**

I am going to use “[**Express Framework**](http://codeforgeek.com/2014/10/top-5-node-packages/)”and middleware called **“Multer”.**

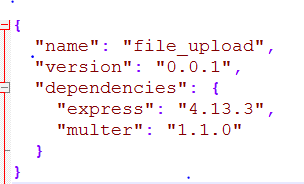
* **Multer**is designed for handling the multipart/form-data, which is primarily used for uploading files and also it will take care of the file handling and HTML for handling form input.
* **Express** is a flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. Following are some of the core features of Express framework :
* Allows to setup middlewares to respond to HTTP Requests.
* Defines a routing table which is used to perform different actions based on HTTP Method and URL.
* Allows to dynamically render HTML Pages based on passing arguments to templates.

In front-end script, we will set the **“target”**parameter of **FORM**to our router. for.eg  **/upload/text**and in **app.get(‘/upload/text’, …)**we will be able to handle our file.

**Creating a project:**

**Step 1:** First we are creating package.json folder in which we are adding dependencies needed to create a project

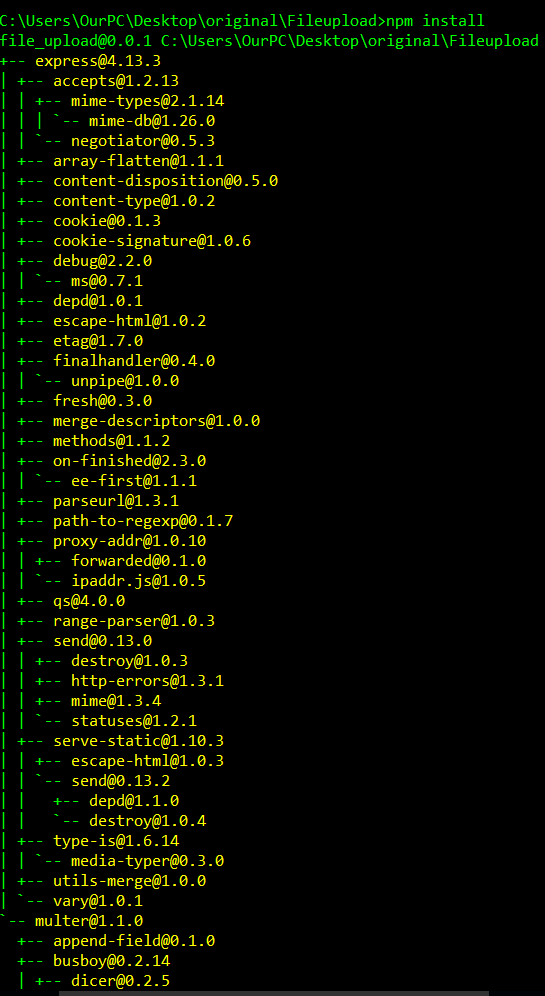
**package.json**

****

**Installing dependency:**

Save the package.json in folder and type **npm install** in command prompt

And a new folder called **‘node\_modules’** is created, which contains the dependencies needed for our project. Please find below results screenshot :



**Step 2:**Following is a **HTML code** inserted in a file intended for display on a World Wide Web [browser](http://searchwindevelopment.techtarget.com/definition/browser) page. The markup tells the Web browser how to display a Web page's words and images for the user.

index.html

<html>

<head>

<title>File upload Node.</title>

</head>

<body>

<form id="uploadForm"

enctype="multipart/form-data"

action="/api/photo"

method="post">

<input type="file" name="userPhoto" />

<input type="submit" value="Upload File" name="submit">

<span id = "status"></span><br>

<a href="/downloads">Download File</a>

</form>

</body>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>

<script src="http://cdnjs.cloudflare.com/ajax/libs/jquery.form/3.51/jquery.form.min.js"></script>

<script>

$(document).ready(function() {

$('#uploadForm').submit(function() {

$("#status").empty().text("File is uploading...");

$(this).ajaxSubmit({

error: function(xhr) {

status('Error: ' + xhr.status);

},

success: function(response) {

console.log(response)

$("#status").empty().text(response);

}

});

return false;

});

});

</script>

</html>

Save the above code in a file named index.html in a Fileupload folder.

**Step 3:** Following is a **Express** app which starts a server and listens on port 3000 for connection. This app allows to upload a file and we can dowland the file for requests to the homepage. For every other path, it will respond with a 404 Not Found**. Multer**will take care of the file handling and HTML for handling form input.

**Server.js**

var express = require("express");

var multer =require('multer');

var app = express();

const testFolder = './uploads';

const fs = require('fs');

var storage = multer.diskStorage({

destination: function (req, file, callback) {

callback(null, './uploads');

},

filename: function (req, file, callback) {

callback(null, file.originalname);

}

});

var upload = multer({ storage : storage}).single('userPhoto');

app.get('/',function(req,res){

res.sendFile(\_\_dirname + "/index.html");

});

app.get('/imgDownload',function(req,res){

var id = req.query.fileName;

console.log("File Name ::"+id);

fs.readFile(testFolder +"/"+ id, function (err, content) {

if (err) {

res.writeHead(400, {'Content-type':'text/html'})

console.log(err);

res.end("No such file");

} else {

//specify Content will be an attachment

res.setHeader('Content-disposition', 'attachment; filename='+id);

res.end(content);

}

});

});

app.get('/downloads',function(req,res){

res.writeHead(200, { 'Content-Type': 'text/html' });

var html = '<!DOCTYPE html><html><head><title>My Title</title></head><body>';

html += '<ul>';

fs.readdir(testFolder, (err, files) => {

files.forEach(file => {

html += '<li><a href="/imgDownload?fileName=' + file + '">' +file + '</a></li>';

});

html += '</ul>';

html += '</body></html>';

res.end(html, 'utf-8');

})

});

app.post('/api/photo',function(req,res){

upload(req,res,function(err) {

if(err) {

return res.end("Error uploading file.");

}

res.end("File is uploaded");

});

});

app.listen(3000,function(){

console.log("Working on port 3000");

});

Save the above code in a file named Server.js

**How to test this project:**

1.Download the zip file from Github location https://github.com/arrDivya/File-Upload\_Download and extract it

2. Open the command prompt in your computer and copy the location of your Fileupload directory. Please find below for reference(In my case it is a in folder called ‘Fileupload’ on my Desktop )



2. As we saved Server.js in a Fileupload folder run the code by entering “node <source code filename>.js “ in my case it is “node Server.js” .

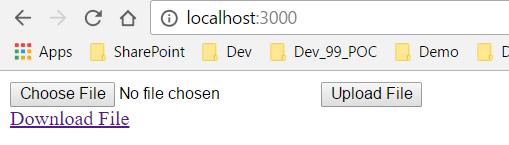
Then you will get “Working on port 3000” .The following is the output



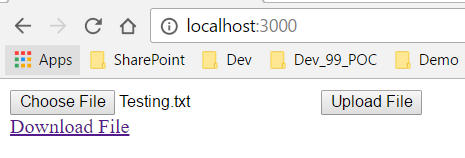
4.Then open any browser and type **localhost:3000**. From the buttons on the window click on choose a file, then click on upload a file. Now the file is uploaded

5.We can download the uploaded file files by clicking on ‘Download File’

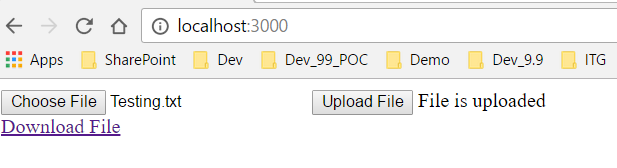
Below are the test results:



**Choosing a file:**



**Uploading a file:**



**Downloading a file:**

