

25) DOWNLOADING FILES USING MULTER PACKAGE IN NODE JS (FROM BACKEND).

```
controller > FileUploadController.js > uploadFile
1  const multer = require("multer");
2  const path = require("path");
3  const FileUpload = require("../schema/FileUploadSchema");
4
5  const storage = multer.diskStorage({
6    destination: "./files/",
7    filename: function (req, file, cb) {
8      cb(null, file.originalname);
9    },
10  });
11
12  const upload = multer({
13    storage: storage,
14    limits: {
15      fileSize: 900000,
16    },
17  }).single("file");
18
19  const uploadFile = (req, res) => {
20    // console.log(req);
21
22    upload(req, res, (err) => {
23      if (err) {
24        res.status(500).json({
25          message: "Internal server error!",
26          error: err,
27        });
28      } else {
29        console.log(req.file.userName);
30        console.log(req.file.originalname);
31        console.log(req.file.size);
32        console.log(req.file.path);
33        var p = path.join(__dirname, "../files/" + req.file.originalname);
34        console.log(p);
35
36        if (req.file === undefined) {
37          res.status(400).json({
38            message: "File was never uploaded!",
39          });
40        } else {
41          // res.status(200).json({
42          //   message: "File uploaded successfully",
43          //   file: req.file.originalname,
44          // });
45          const fileUpload = new FileUpload({
46            userName: req.file.userName,
47            name: req.file.originalname,
48            size: req.file.size,
49            url: p,
50            type: req.file.mimetype,
51          });
52
53          fileUpload.save((err, data) => {
54            if (err) {
55              res.status(500).json({
56                message: "Error while uploading the file to the database!",
57                error: err,
58              });
59            } else {
60              res.status(200).json({
61                message: "File uploaded successfully",
62                file: req.file.originalname,
63              });
64            }
65          });
66        }
67      }
68    });
69  };
70
71  module.exports = {
72    uploadFile,
73  };
74
```

26) DOWNLOADING FILES USING MULTER PACKAGE IN REACT JS (FROM FRONTEND).

```
src > components > form > FileUpload.js > ...
1  import axios from 'axios';
2  import React from 'react'
3  import { useForm } from 'react-hook-form'
4
5  export const FileUpload = () => {
6    const {register, handleSubmit}=useForm();
7    const submit = (data)=>{
8      const formData = new FormData();
9      formData.append("file", data.file[0]);
10     formData.append("userName", data.userName);
11     axios.post("http://localhost:2456/files/upload", formData).then((res)=>{
12       console.log(res.data)
13       console.log(data.file[0])
14     }).catch((err)=>{
15       console.log(err)
16     })
17   }
18
19   return (
20     <>
21     <form onSubmit={handleSubmit(submit)}>
22       ENTER NAME:<input type="text" {...register("userName")}/>
23       SELECT FILE:<input type="file" {...register("file")}/><br/>
24       SUBMIT:<input type="submit" value="upload"/><br/>
25     </form>
26   </>
27 )
28 }
29
```

POST http://localhost:2456/ POST http://localhost:2456/ GET http://localhost:2456/c + ... No Environment

http://localhost:2456/files/upload Save

POST http://localhost:2456/files/upload Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	file	login page demo.jpg x			
	Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 205 ms Size: 336 B Save as Example

Pretty Raw Preview Visualize JSON

```
1
2  "message": "File uploaded successfully",
3  "file": "login page demo.jpg"
4
```

27) ENCRYPTING THE USER PASSWORD AND SAVING IT IN DATABASE.

```
utils > js encrypt.js > encrypt
1  const bcrypt = require('bcrypt');
2  const saltRounds = 10;
3  // const secret = "aza1234"
4  ⚡
5  const encrypt = (password)=>{
6    const hash= bcrypt.hashSync(password,saltRounds)
7    return hash
8  }
9
10 const compare = (password, hash)=>{
11   const result = bcrypt.compareSync(password, hash);
12   return result
13 }
14
15 // const promise = compare(secret,"$2b$10$ZgBd.xh6aed2yVfxvV9hk.VQngU1bI3Cl8nYP0ik1qfUUdi7wcqCO");
16
17 // console.log(promise)
18 // promise.then((res)=>{
19 //   console.log(res);
20 // }).catch((err)=>{
21 //   console.log(err)
22 // })
23
24
25 module.exports = {encrypt, compare}
```

http://localhost:2456/users/get

Save



GET http://localhost:2456/users/get

Send

Params Authorization Headers (8) Body • Pre-request Script Tests Settings

Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 38 ms Size: 1.38 KB Save as Example

Pretty Raw Preview Visualize JSON



```
50  "email": "varma@gmail.com",
51  "password": "12345",
52  "age": 12,
53  "role": "64019a4ef2f0192fe4fd26d2",
54  "_v": 0
55  },
56  {
57    "_id": "64229dfb781bf8d1c99815cd",
58    "name": "varma",
59    "email": "varma@gmail.com",
60    "password": "$2b$10$dPQhyIkZTL0ywnb0kLS8eeefuFmngnEt5cDexpK/27ET5xWFB/Yn2",
61    "age": 25,
62    "role": "64019a4ef2f0192fe4fd26d2",
63    "_v": 0
64  }
```