Node.js Upload File to Server – Example

Node.js Upload File To Server

Node.js Upload File – In this Node.js Tutorial, we shall learn to Upload a File to Node.js Server from a web client. In other words a client could upload a file to Node.js Server.

To Upload File To Node.js Server, following is a step by step guide :

1. Prerequisite modules

We shall use http, fs and formidable modules for this example. **http** : for server acitivities. **node fs** : to save the uploaded file to a location at server. **formidable** : to parse html form data. If above mentioned modules are not installed already, you may install now using NPM. Run the following commands, in Terminal, to install the respective modules :

```
npm installhttp npm installfs npm installformidable
```

2. Prepare a HTML Form

Prepare a HTML page (upload_file.html) with the following form, which includes input tags for file upload and form submission.

HTML Form for file upload

```
<form action="fileupload" method="post" enctype="multipart/form-data">
    <input type="file" name="filetoupload">
    <input type="submit" value="Upload">
</form>
```

3. Create a HTTP Server

Create a HTTP Server that listens at port 8086 (you may change the port) and servers two urls as shown below :

Create HTTP Server

```
http.createServer(function (req, res) {
    if (req.url == '/uploadform') {
        // if request URL contains '/uploadform'
        // fill the response with the HTML file containing upload form
    } else if (req.url == '/fileupload') {
        // if request URL contains '/fileupload'
        // using formiddable module,
        // read the form data (which includes uploaded file)
        // and save the file to a location.
    }
 }).listen(8086);
```

4. File Saving

Using formidable module, parse the form elements and save the file to a location. Once file is uploaded,

you may respond with a message, saying file upload is successful. Initially, files are saved to a temporary location. We may use fs.rename() method, with the new path, to move the file to a desired location.

Using Formidable module to parse form

```
var form = new formidable.IncomingForm();
form.parse(req, function (err, fields, files) {
    // oldpath : temporary folder to which file is saved to
    var oldpath = files.filetoupload.path;
    var newpath = upload_path + files.filetoupload.name;
    // copy the file to a new location
    fs.rename(oldpath, newpath, function (err) {
        if (err) throw err;
        // you may respond with another html page
        res.write('File uploaded and moved!');
        res.end();
    });
 });
```

Example for Node.js Upload File

Following is a complete working Example for Node.js Upload File

This example has two files as shown below :

```
arjun@tutorialkart:~/workspace/nodejs/upload_file$ ls
nodejs-upload-file.js  upload_file.html
```

upload_file.html

```
<!DOCTYPE html>
<html>
<head>
<title>Upload File</title>
<style>
    body{text-align:center;}
    form{display:block;border:1px solid black;padding:20px;}
</style>
</head>
<body>
    <h1>Upload files to Node.js Server</h1>

    <form action="fileupload" method="post" enctype="multipart/form-data">
        <input type="file" name="filetoupload">
        <input type="submit" value="Upload">
    </form>
</body>
</html
```

nodejs-upload-file.js

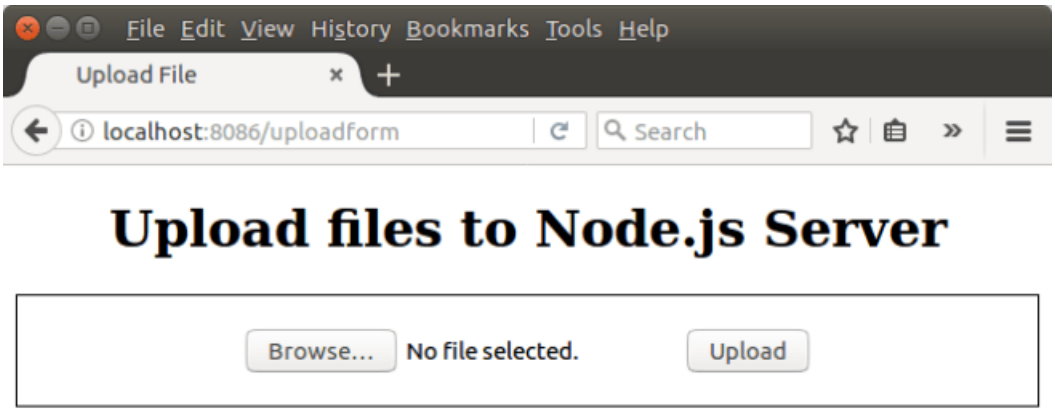
```
var http = require('http');
var fs = require('fs');
var formidable = require('formidable');

// html file containing upload form
var upload_html = fs.readFileSync("upload_file.html");

// replace this with the location to save uploaded files
var upload_path = "/home/arjun/workspace/nodejs/upload_file/";

http.createServer(function (req, res) {
    if (req.url == '/uploadform') {
      res.writeHead(200);
      res.write(upload_html);
      return res.end();
    } else if (req.url == '/fileupload') {
        var form = new formidable.IncomingForm();
        form.parse(req, function (err, fields, files) {
            // oldpath : temporary folder to which file is saved to
            var oldpath = files.filetoupload.path;
            var newpath = upload_path + files.filetoupload.name;
            // copy the file to a new location
            fs.rename(oldpath, newpath, function (err) {
                if (err) throw err;
                // you may respond with another html page
                res.write('File uploaded and moved!');
                res.end();
            });
        });
    }
 }).listen(8086);
```

Run Node.js script file in Terminal with node

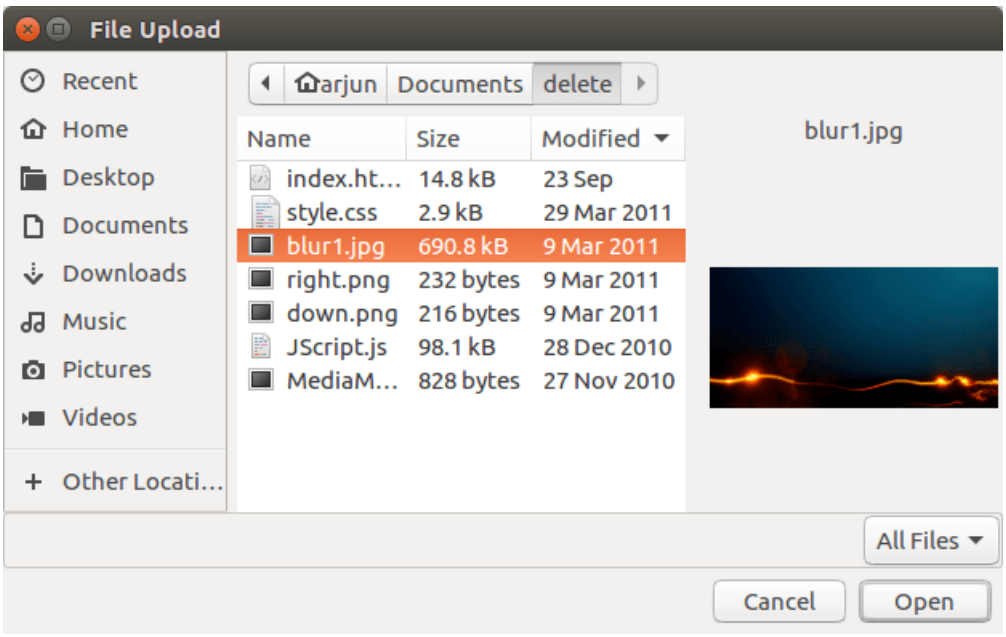
```
Terminal

arjun@tutorialkart:~/workspace/nodejs/upload_file$ node nodejs-upload-file.js
```

The files uploaded are saved next to the node.js file, nodejs-upload-file.js. You may change this location in the node.js script file.

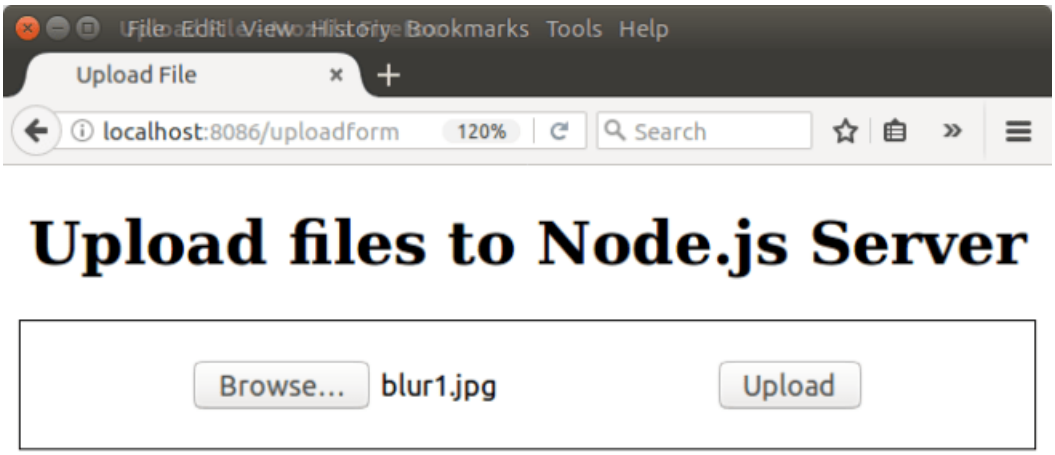
Open a Web browser (HTTP Client) and hit the url**http://localhost:8086/uploadform**



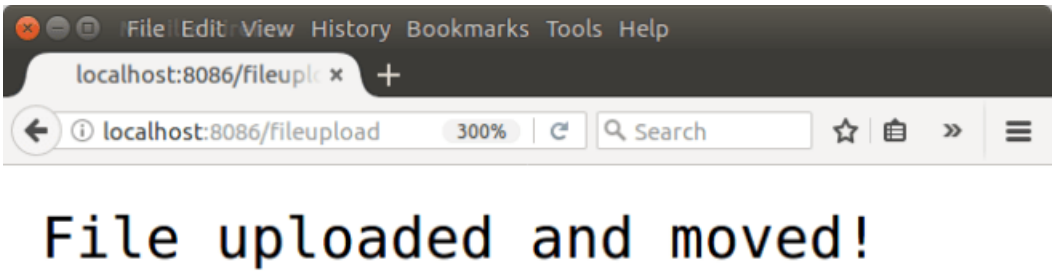
Click on browse.



Select a file and click on Open.



Currently, the file is uploaded to the form. Click on the Upload button for the Node.js to parse the form elements and save the file.



Check the Node.js Server, next to the node.js script file.

```
Terminal
arjun@tutorialkart:~/workspace/nodejs/upload_file$ ls
blur1.jpg  nodejs-upload-file.js  upload_file.html
```

Conclusion :

In this Node.js Tutorial – **Node.js Upload File to Server**, we have learnt to use formidable, fs and http modules for uploading a file to Node.js Server.

- ☐ Node.js - Array to Buffer

Node.js HTTP

- ☐ Node.js - Create HTTP Web Server
- ☐ Node.js - Redirect URL

Node.js MySQL

- ☐ Node.js MySQL
- ☐ Node.js MySQL - Connect to MySQL Database
- ☐ Node.js MySQL - SELECT FROM
- ☐ Node.js MySQL - SELECT WHERE
- ☐ Node.js MySQL - ORDER BY
- ☐ Node.js MySQL - INSERT INTO
- ☐ Node.js MySQL - UPDATE
- ☐ Node.js MySQL - DELETE
- ☐ Node.js MySQL - Result Object

Node.js MongoDB

- ☐ Node.js MongoDB
- ☐ Node.js - Connect to MongoDB
- ☐ Node.js - Create Database in MongoDB
- ☐ Node.js - Drop Database in MongoDB
- ☐ Node.js - Create Collection in MongoDB
- ☐ Node.js - Delete Collection in MongoDB
- ☐ Node.js - Insert Documents to MongoDB Collection
- ☐ MongoError: failed to connect to server

Node.js Mongoose

- ☐ Node.js Mongoose Tutorial
- ☐ Node.js Mongoose - Installation
- ☐ Node.js Mongoose - Connect to MongoDB
- ☐ Node.js Mongoose - Define a Model
- ☐ Node.js Mongoose - Insert Single Document to MongoDB
- ☐ Node.js Mongoose - Insert Multiple Documents to MongoDB

Node.js URL

- ☐ Node.js - Parse URL parameters

Node.js FS (File System)

- ☐ Node FS
- ☐ Node FS - Read a File
- ☐ Node FS - Create a File
- ☐ Node FS - Write to a File
- ☐ Node FS - Append to a File
- ☐ Node FS - Rename a File
- ☐ Node FS - Delete a File
- ☐ Node FS Extra - Copy a Folder

Node.js JSON

- ☐ Node.js Parse JSON
- ☐ Node.js Write JSON Object to File

Node.js Error Handling

- ☐ Node.js Try Catch

Node.js Examples

- ☐ Node.js Examples
- ☐ Node.js - Handle Get Requests
- ☐ Node.js Example - Upload files to Node.js server

Useful Resources

- ☐ Node.js Interview Questions
- ☐ How to Learn Programming