



WEB TECHNOLOGIES

Express JS

Prof. Vinay Joshi and Dr. Sarasvathi V

Department of Computer Science and Engineering.

WEB TECHNOLOGIES

Express JS FILE UPLOAD

Department of Computer Science and Engineering

WEB TECHNOLOGIES

FILE UPLOAD

Express JS

- In Express.js, file upload is slightly difficult because of its asynchronous nature and networking approach.
- It can be done by using middleware to handle multipart/form data.
- There are many middleware that can be used like multer, connect, body-parser etc.
- There are two popular and robust options for multipart form processing: Busboy and Formidable.
- Formidable to be slightly easier, as it has a callback that provides objects containing the fields and the files,
- In Busboy, you must listen for each field and file event.

WEB TECHNOLOGIES

FILE UPLOAD

Express JS

- Simple express middleware for uploading files.
- ***npm i express-fileupload***

```
C:\Users\Aisha Begum\express\myexpress>npm i express -fileupload
npm WARN myexpress@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ express@4.17.1
updated 1 package and audited 180 packages in 7.902s

8 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\Aisha Begum\express\myexpress>
```

WEB TECHNOLOGIES

FILE UPLOAD

Express JS

- When you upload a file, the file will be accessible from `req.files`
- It is similar to `req.body` and is turned on either by `express.bodyParser()` or `express.multipart()` middlewares.
- Express.js (and other modules behind the scene) process the request data (which is usually a form) and give us extensive information in the `req.files.FIELD_NAME` object.

WEB TECHNOLOGIES

FILE UPLOAD

Express JS

To illustrate how req.files works, let's add this route to the req project:

```
app.post('/upload', function(req, res){  
  console.log(req.files.archive);  
  //read req.files.archive.path  
  //process the data  
  //save the data  
  res.end();  
})
```

WEB TECHNOLOGIES

FILE UPLOAD



Express JS

Example:

- You're uploading a file called **car.jpg**
- Your input's name field is **foo**: `<input name="foo" type="file" />`
- In your express server request, you can access your uploaded file from `req.files.foo`:

```
app.post('/upload', function(req, res) {  
  console.log(req.files.foo); // the uploaded file object  
});
```

WEB TECHNOLOGIES

FILE UPLOAD



Express JS

The **req.files.foo** object will contain the following:

req.files.foo.name: "car.jpg"

req.files.foo.mv: A function to move the file elsewhere on your server. Can take a callback or return a promise.

req.files.foo.mimetype: The mimetype of your file

req.files.foo.data: A buffer representation of your file, returns empty buffer in case useTempFiles option was set to true.

WEB TECHNOLOGIES

FILE UPLOAD



Express JS

The **req.files.foo** object will contain the following:

req.files.foo.tempFilePath: A path to the temporary file in case useTempFiles option was set to true.

req.files.foo.truncated: A boolean that represents if the file is over the size limit

req.files.foo.size: Uploaded size in bytes

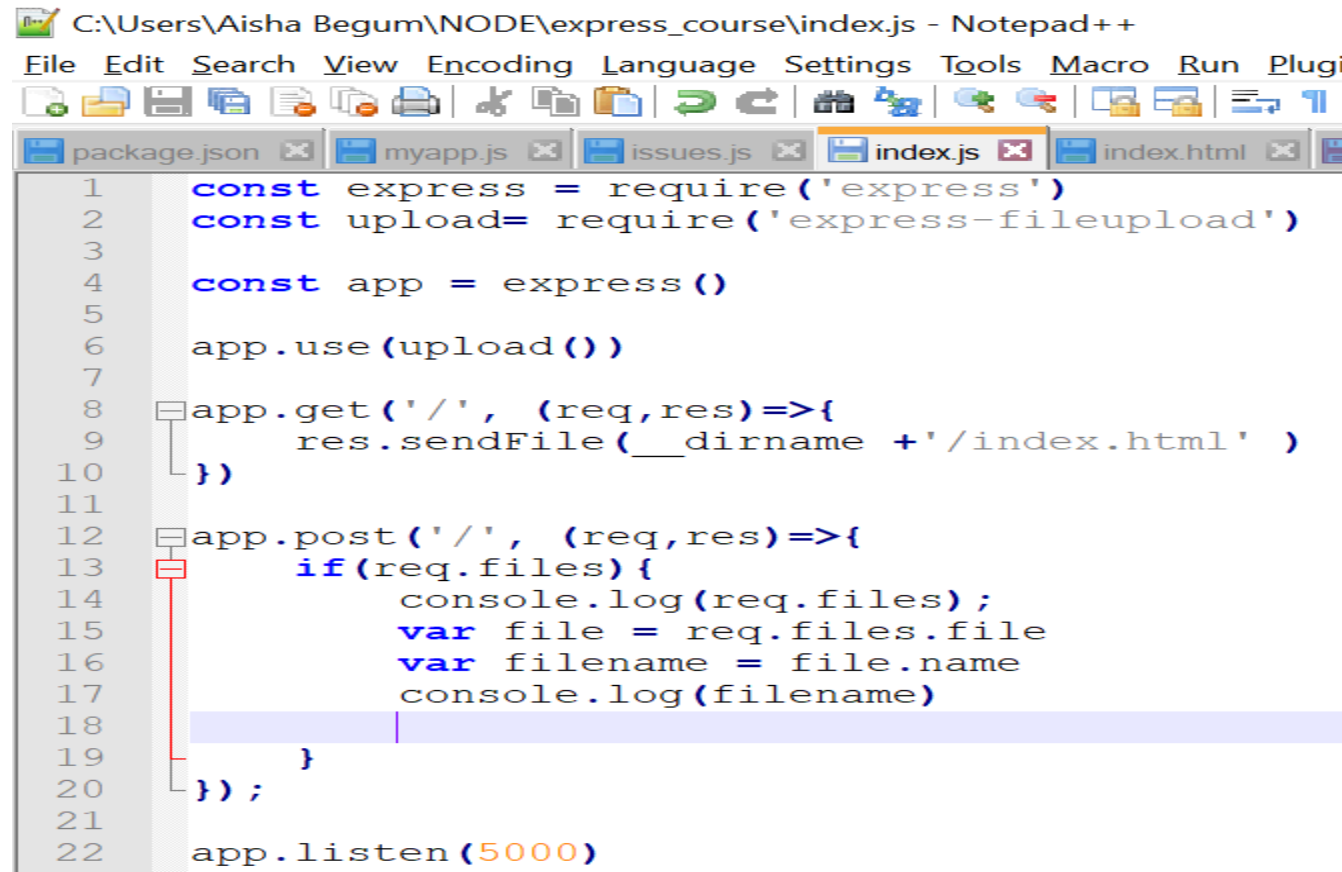
req.files.foo.md5: MD5 checksum of the uploaded file

WEB TECHNOLOGIES

FILE UPLOAD

Express JS

Index.js for printing out the file name and details of uploaded file



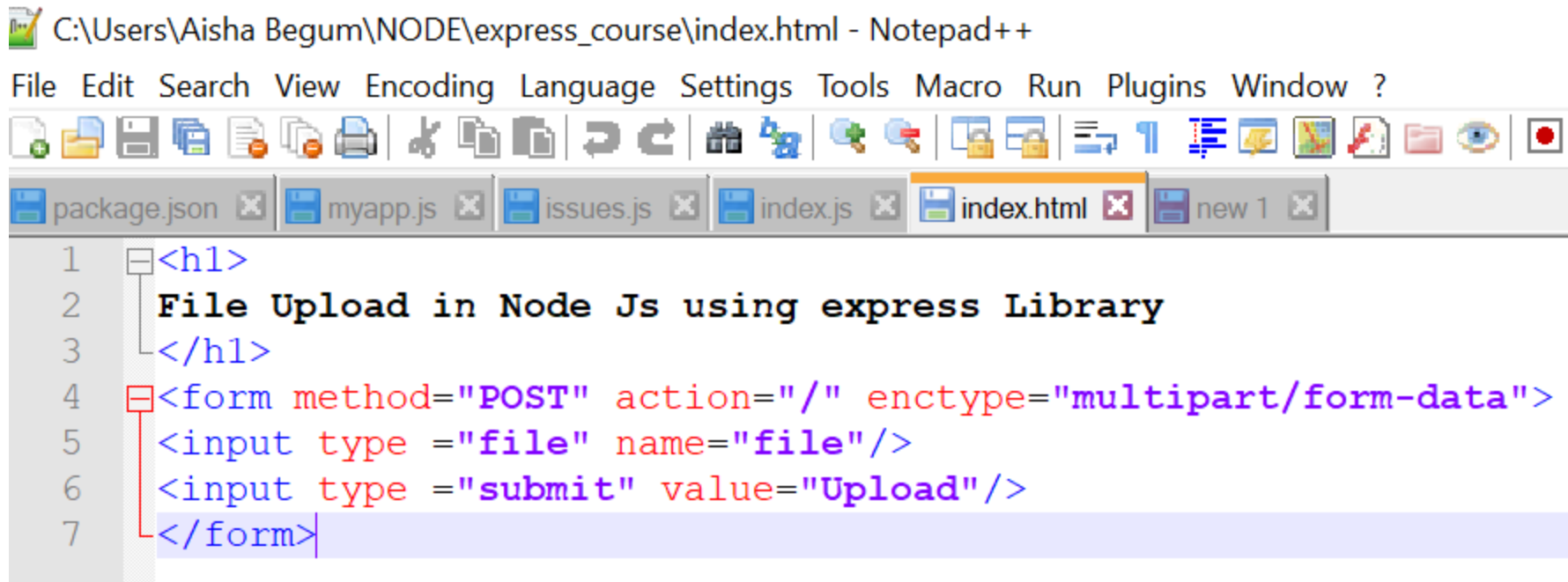
```
C:\Users\Aisha Begum\NODE\express_course\index.js - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plug
package.json x myapp.js x issues.js x index.js x index.html x
1  const express = require('express')
2  const upload= require('express-fileupload')
3
4  const app = express()
5
6  app.use(upload())
7
8  app.get('/', (req,res)=>{
9      res.sendFile(__dirname + '/index.html' )
10 })
11
12 app.post('/', (req,res)=>{
13     if(req.files){
14         console.log(req.files);
15         var file = req.files.file
16         var filename = file.name
17         console.log(filename)
18     }
19 }
20 });
21
22 app.listen(5000)
```

WEB TECHNOLOGIES

FILE UPLOAD

Express JS

Index.html: as simple file upload script shown as below:



C:\Users\Aisha Begum\NODE\express_course\index.html - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

package.json x myapp.js x issues.js x index.js x index.html x new 1 x

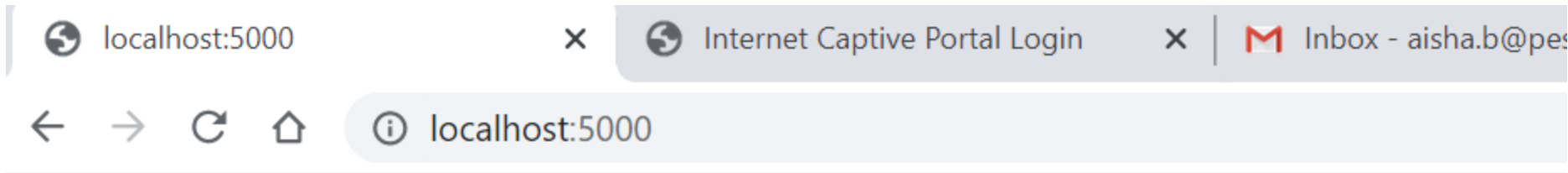
```
1 <h1>
2   File Upload in Node Js using express Library
3 </h1>
4 <form method="POST" action="/" enctype="multipart/form-data">
5   <input type="file" name="file"/>
6   <input type="submit" value="Upload"/>
7 </form>
```

WEB TECHNOLOGIES

FILE UPLOAD

Express JS

In browser when we type the url: `http://localhost:5000`



File Upload in Node Js using express Library

No file chosen

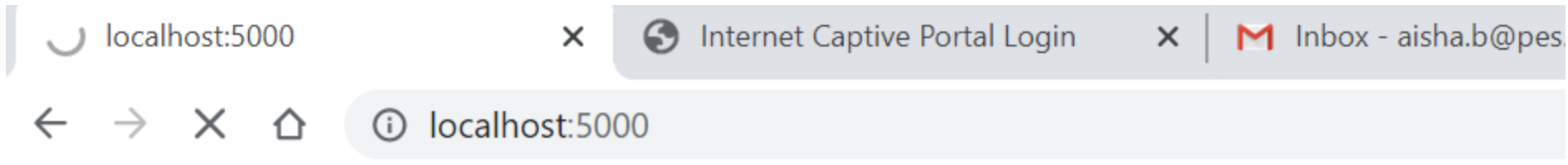
WEB TECHNOLOGIES

FILE UPLOAD

Express JS

In browser when we type the url: <http://localhost:5000> and choose file.

Here I have chosen images.jpg



File Upload in Node Js using express Library

Choose file

images.jpg

Upload

WEB TECHNOLOGIES

FILE UPLOAD



Express JS

Upload a file then you can see the details in the command prompt about the file

```
C:\Users\Aisha Begum\NODE\express_course>node index
{
  file: {
    name: 'images.jpg',
    data: <Buffer ff d8 ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 ff db 00 84 00 09 06 07 13 13 12 15 13 13 13 15 16 15 17 17 1d 1a 18 18 18 18 1b 1d 1d 18 ...
10523 more bytes>,
    size: 10573,
    encoding: '7bit',
    tempFilePath: '',
    truncated: false,
    mimetype: 'image/jpeg',
    md5: '0627265b42b7f89b5db7a4cb9925bd08',
    mv: [Function: mv]
  }
}
images.jpg
```

WEB TECHNOLOGIES






FILE UPLOAD

Express JS

Create a file for upload in the root folder.

The uploaded files will be saved in the created folder

Local Disk (C:) > Users > Aisha Begum > NODE > express_course

	Name	Date modified	Type	Size
★	middleware	06-06-2020 12:14	File folder	
★	node_modules	22-06-2020 15:02	File folder	
★	public	06-06-2020 11:27	File folder	
★	routes	06-06-2020 12:36	File folder	
★	uploads	22-06-2020 15:09	File folder	
	 index	22-06-2020 15:03	Chrome HTML Docu...	1 KB
	 index	23-06-2020 11:44	JavaScript File	1 KB
	 Members	06-06-2020 11:59	JavaScript File	1 KB
	 package.json	22-06-2020 15:02	JSON File	1 KB
	 package-lock.json	22-06-2020 15:02	JSON File	53 KB

WEB TECHNOLOGIES

FILE UPLOAD

Express JS

Using file.mv method we will be storing the file in the created folder

```
package.json  myapp.js  issues.js  index.js  index.html  new 1
const express = require('express')
const upload= require('express-fileupload')

const app = express()

app.use(upload())

app.get('/', (req,res)=>{
  res.sendFile(__dirname + '/index.html' )
})

app.post('/', (req,res)=>{
  if(req.files){
    console.log(req.files);
    var file = req.files.file
    var filename = file.name
    console.log(filename)
    file.mv('./uploads/' + filename,function(err){
      if(err){
        res.send(err)
      }else {
        res.send("File Uploaded")
      }
    })
  }
})

app.listen(5000)
```

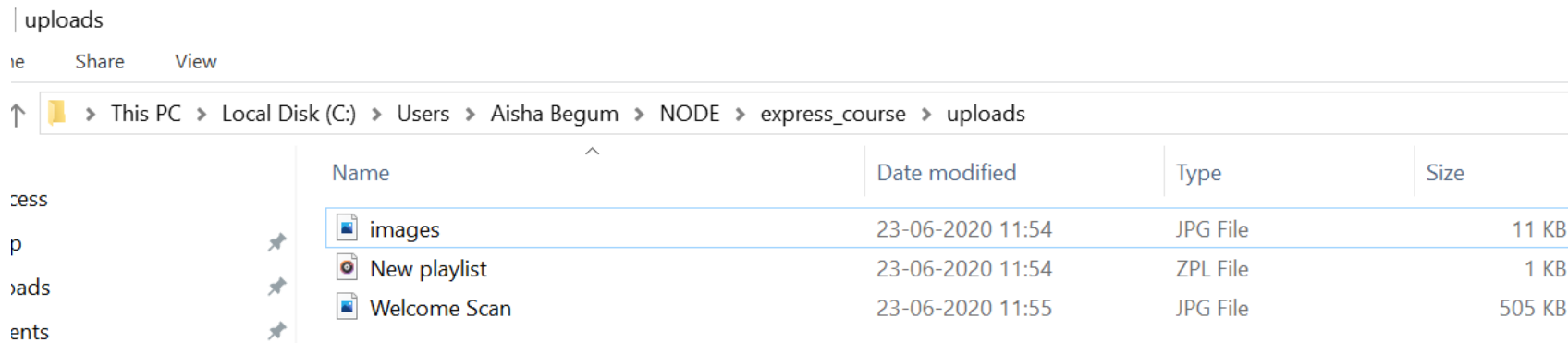
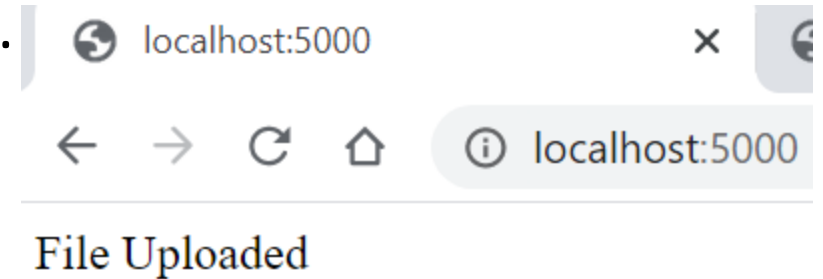

WEB TECHNOLOGIES

FILE UPLOAD



Express JS

In the browser you can see that file has been uploaded and when u check the uploads folder you can see that file has been uploaded.
The file upload can be on any type





THANK YOU

Vinay Joshi and Dr.Sarasvathi V

Department of Computer Science and Engineering

vinayj@pes.edu

sarsvathiv@pes.edu

Acknowledgement

The slides are created from various internet resources with valuable contributions from multiple professors