**File Upload with Multer**

ajke amra dekhbo kivabe express js e file upload korte hoy. Server side e jokhon kono file upload er request ashe tokhon she file property validate kore handle kora ebong proper file ta upload hocche kina sheta ensure kora onek important ebong jekeno application er jonno sheta ekta integral part.

‘Express js’ folder e index.html name ekta file create kore nicher code likhlam. Erpor live server e run korlam.

Index.html…

<!DOCTYPE *html*>

<html *lang*="en">

  <head>

    <meta *charset*="UTF-8" />

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

    <title>Express File Upload with Multer</title>

  </head>

  <body>

    <form *action*="http://localhost:3000/" *method*="post">

      <input *type*="file" *name*="avatar" />

      <input *type*="submit" *value*="Submit" />

    </form>

  </body>

</html>

Eta amar front end er part. Ekhon kono client ekhane file upload kore submit korle amader server e request jabe ebong server e kono ekta route e sheta ke receive kore process korbo. Etai amar file upload er simple procedure. Etar jonno ekta simple express server run kore rakhsi.

Index.js …

*const* express = *require*("express");

*const* app = *express*();

*//apllication route*

app.*get*("/", (*req*, *res*) *=>* {

  res.*send*("This is file upload tutorial");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Amra jani jokhon kono file upload hocche tar mane ekta form process hocche. So sheta amra ‘get’ method e accept korte parbo na. ‘post’ e accept korte hobe.

Ei je client file ta upload korbe shei file ta form data hishebe pathate hoy. Shetar jonno amader index.html er ‘form’ tag er moddhe ‘enctype= `multipart/form-data` ’ dite hobe. Ei multipart data ke process korar jonno ki rokom middleware dorkar ta ekhono amra shikhi nai. Amra express.json diye json data nite shikhechilam. Ekhon ei multipart data ke parse korte express amader ‘multer’ name ekta package dise. ‘npm install multer’ kore eta install kore nilam.

Index.js e app.get() shoraya app.post() kore nilam. Karon ami post method e file ta receive korbo. Echara multer package ta ke require kore niye ashlam.

Index.js …

*const* express = *require*("express");

*const* multer = *require*("multer");

*const* app = *express*();

*//apllication route*

app.*post*("/", (*req*, *res*) *=>* {

  res.*send*("This is file upload tutorial");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Multer package tar 1st kotha holo, ami je file upload korbo shei upload korar jonno ekta environment take toiri kore dite hobe. Ei je multer ami paisi, eta function akare call kore dile ami multer er jonno dorkari options shekhane dite pari. Er jonno ‘upload’ name ekta variable niye multer() call kore dicchi. Multer() parameter hishebe ekta object ney. Tar moddhe amra bivinno options ullekh kore dite pari. Most basic option hocche am ikon folder e file ta upload korte chai. So tar jonno ‘express js’ folder e ‘uploads’ name ekta folder create korlam. Upload korar po rei folder e amar shomosto file jabe. Parameter object e ‘dest’ name ekta property ache. Ei destination e bole dite hobe ‘uploads’ folder er path diye dite hobe.

Ei je upload variable ta amra nisi eta basically ekta middleware amader return korbe.

amra jokhon kono route e hit kori tokhon oi route er jonno je CB dei shetao ekta middleware. Ekta route er jonno amra oshonkho middleware function boshate pari. To ei middleware e jawar age amra multer er ekta middleware use korbo. Multer onekgula middleware provide kore. Er moddhe ekta hoche upload.single() method. Amra eta use korbo. Ei method ba middleware ke ekta basic parameter dite hobe. Eta holo file upload er input fielde er ‘name’ which is ‘avatar’ in our case. So jokhoni post method e “/” route e hit hobe tokhon express prothome upload.single(‘avatar’) method er moddho diye jabe. Ei middleware er kaj hocche multipart data ke parse kora ebong she khujbe ‘avatar’ name je file ta ache sheta ke upload korte hobe.

Index.html…

<!DOCTYPE *html*>

<html *lang*="en">

  <head>

    <meta *charset*="UTF-8" />

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

    <title>Express File Upload with Multer</title>

  </head>

  <body>

    <form

*action*="http://localhost:3000/"

*method*="post"

*enctype*="multipart/form-data"

    >

      <input *type*="file" *name*="avatar" />

      <input *type*="submit" *value*="Submit" />

    </form>

  </body>

</html>

Index.html theke ‘go live’ e giye live hote hobe.

Index.js…

*const* express = *require*("express");

*const* multer = *require*("multer");

*const* app = *express*();

*const* UPLOADS\_FOLDER = "./uploads/";

*// prepare the final multer upload object*

*var* upload = *multer*({

  dest: UPLOADS\_FOLDER,

});

*//apllication route*

app.*post*("/", upload.*single*("avatar"), (*req*, *res*) *=>* {

  res.*send*("This is file upload tutorial");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ekhon live index.html er route orthat <http://127.0.0.1:5500/index.html> ---ekhane giye ekta browse option e click kore ekta file select kore submit chap dilam. Shathe shathe localhost:3000 route e chole gelam. Lekha ashlo “This is file upload tutorial”. ‘uploads’ folder check korlam. Dekhlam file ta ekta hijibiji akare chole ashche.

Amra simple ekta file upload korlam. But amra ekhon multiple files upload korte chai. Er jonno index.html e ‘avatar’ namok input field e ‘multiple’ namok attribute boshaya dilam. So amar frontend er kaj holo. Ekhon index.js e kaj korbo. Amra ekhane upload.single() use korechilam single file handle korar jonno. Multiple files handle korte upload.array() use korbo. Ei method er moddhe argument hishebe dite hobe input field er “name” ebong koyta file upload korte chai.

Index.html…

<!DOCTYPE *html*>

<html *lang*="en">

  <head>

    <meta *charset*="UTF-8" />

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

    <title>Express File Upload with Multer</title>

  </head>

  <body>

    <form

*action*="http://localhost:3000/"

*method*="post"

*enctype*="multipart/form-data"

    >

      <input *type*="file" *name*="avatar" *multiple* />

      <input *type*="submit" *value*="Submit" />

    </form>

  </body>

</html>

Index.js…

*const* express = *require*("express");

*const* multer = *require*("multer");

*const* app = *express*();

*const* UPLOADS\_FOLDER = "./uploads/";

*// prepare the final multer upload object*

*var* upload = *multer*({

  dest: UPLOADS\_FOLDER,

});

*//apllication route*

app.*post*("/", upload.*array*("avatar", 3), (*req*, *res*) *=>* {

  res.*send*("This is file upload tutorial");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ekhon live index.html e giye 3 ta porjonto file select kore submit kore thikmoto upload korte partesi. ‘uploads’ folder e dekhao jaitese valo moto. But jokhoni 4 ta file select kore upload korte gesi tokhoni bishal error diye dise.

Index.html e amar type file er input field ekta. Ekhon type file er arekta input field rakhte chai. Er jonno ‘gallery’ name diye arekta input field nilam. Index.js e multiple input fields handle korte amader upload.fields() middleware use korte hobe. Ei function e argument hishebe ekta array of objects dite pari. Ekekta object e thakbe input field er name ebong maximum number of uploads.

Index.html…

<!DOCTYPE *html*>

<html *lang*="en">

  <head>

    <meta *charset*="UTF-8" />

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

    <title>Express File Upload with Multer</title>

  </head>

  <body>

    <form

*action*="http://localhost:3000/"

*method*="post"

*enctype*="multipart/form-data"

    >

      <input *type*="file" *name*="avatar" *multiple* /> <br /><br />

      <input *type*="file" *name*="gallery" *multiple* />

      <input *type*="submit" *value*="Submit" />

    </form>

  </body>

</html>

Index.js …

*const* express = *require*("express");

*const* multer = *require*("multer");

*const* app = *express*();

*const* UPLOADS\_FOLDER = "./uploads/";

*// prepare the final multer upload object*

*var* upload = *multer*({

  dest: UPLOADS\_FOLDER,

});

*//apllication route*

app.*post*(

  "/",

  upload.*fields*([

    { name: "avatar", maxCount: 1 },

    { name: "gallery", maxCount: 2 },

  ]),

  (*req*, *res*) *=>* {

    res.*send*("This is file upload tutorial...Yahoo");

  }

);

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ekhono ager moto kaj kortese. Ebong ekhono max limit er cheye beshi file upload korle bishal error show kortese.

Ekhon jodi emon case hoy je amra kono file upload korte chai na kintu amra form data handle korte chai tahole ki korbo? Form data handle korar jonnow multer use hoy. Eta bojhar jonno index.html e khanikta change anlam.

Index.html …

<!DOCTYPE *html*>

<html *lang*="en">

  <head>

    <meta *charset*="UTF-8" />

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

    <title>Express File Upload with Multer</title>

  </head>

  <body>

    <form

*action*="http://localhost:3000/"

*method*="post"

*enctype*="multipart/form-data"

    >

      <input *type*="text" *name*="name" /> <br /><br />

*<!-- <input type="file" name="gallery" multiple /> -->*

      <input *type*="submit" *value*="Submit" />

    </form>

  </body>

</html>

Ekhetre index.js e app.post() e middleware hishabe upload.none() use korbo. Ekhtre ar file upload korte hobe na. but amra onnanno input data gula niye handle korte parbo. Shekhetre req.body er moddhe form datagula thakbe.

Ekhon abar single file input field er setup e chole gelam. Server side orthat index.js e ekhon porjonoto amra bole dei nai kon dhoroner file upload korte parbe. kono restriction dei nai. But amader server side e validation korte hobe. Ta nahole user kono malicious file upload kore dite pare. Ami jotoi client side e validate kori na keno shetar daam nai. So ekhon amra validation dekhbo.

First e file er size er ekta restriction dibo. Er jonno index.js e multer() function er object argument e ‘limits’ name ekta property diye dibo. Eta ekta object jar fileSize namok property er moddhe Byte e size ta bole dibo.

Index.html …

<!DOCTYPE *html*>

<html *lang*="en">

  <head>

    <meta *charset*="UTF-8" />

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

    <title>Express File Upload with Multer</title>

  </head>

  <body>

    <form

*action*="http://localhost:3000/"

*method*="post"

*enctype*="multipart/form-data"

    >

      <input *type*="file" *name*="avatar" />

      <input *type*="submit" *value*="Submit" />

    </form>

  </body>

</html>

Index.js …

*const* express = *require*("express");

*const* multer = *require*("multer");

*const* app = *express*();

*const* UPLOADS\_FOLDER = "./uploads/";

*// prepare the final multer upload object*

*var* upload = *multer*({

  dest: UPLOADS\_FOLDER,

  limits: {

    fileSize: 1000000,

  },

});

*//apllication route*

app.*post*("/", upload.*single*("avatar"), (*req*, *res*) *=>* {

  res.*send*("This is file upload tutorial...Yahoo");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ekhon 1MB er niche kono file upload kore submit korle kono problem hoy na. but 1MB er upore gelei bishal error ashe.

Ekhon user kon type er file upload korbe shetar upor restriction dibo. Suppose, shudhu image type file upload korte dite chai like jpg, png etc. tahole etar jonno ki korbo? Er jonno ‘limits’ namok property er pore fileFilter namok ekta property dibo. Etar value hishebe dibo ekta function.

Index.js …

*const* express = *require*("express");

*const* multer = *require*("multer");

*const* app = *express*();

*const* UPLOADS\_FOLDER = "./uploads/";

*// prepare the final multer upload object*

*var* upload = *multer*({

  dest: UPLOADS\_FOLDER,

  limits: {

    fileSize: 1000000,

  },

*fileFilter*: (*req*, *file*, *cb*) *=>* {

    if (

      file.mimetype *===* "image/png" ||

      file.mimetype *===* "image/jpeg" ||

      file.mimetype *===* "image/jpg"

    ) {

*cb*(null, true);

    } else {

*cb*(new *Error*("Only jpg, jpeg and png format is allowed"));

    }

  },

});

*//apllication route*

app.*post*("/", upload.*single*("avatar"), (*req*, *res*) *=>* {

  res.*send*("This is file upload tutorial...Yahoo");

});

app.*use*((*err*, *req*, *res*, *next*) *=>* {

  if (err) {

    if (err instanceof multer.MulterError) {

      res.*status*(500).*send*("Upload ErRor");

    } else {

      res.*status*(500).*send*(err.message);

    }

  } else {

    res.*send*("Success");

  }

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ei function ta receive korbe req, file ebong cb. Req hocche request object orthat client er kach theke je request ta asche sheta. ‘file’ er moddhe client er upload kora file er detail information thakbe. cb hocche ekta callback. Ami shob kaj kore ei callback er maddhome ami express ke Janay dibo ami ki client er file ta allow korbo naki korbo na. ekhon ekta file upload kore console.log(file) kore dekhbo ei ‘file’ er moddhe ki ache. Ei ‘file’ ta holo ekta object jar moddhe fieldname, originalname, mimetype etc property ache. So fileFilter er function er moddhe if-else diye file.mimetype check kore dekhbo file ta kon format er. Jodi true hoy cb(null,true) dibo. Ei callback ta error back pattern follow kore. Prothome null diye error ta null kore dilam. Cb er 2nd parameter e true or false dite hoy. True dile bujhbe file ta allowed. File upload hobe. Porer middleware e proceed korbe. ar false dile okhanei stop kore jabe.

Onno dike if-else jodi false return kore tahole cb(new Error("Only jpg, jpeg and png format is allowed") kore dilam. Ekhane chaile cb(null, false) o kore dite partam. Eta niye ektu pore bolbo. Jehetu ekhane cb er moddhe ekta error diye disi shehetu ektake amar handle kore show korte hobe. Tai etar jonno ekdom niche ekta custom error handling middleware rakhlam. Ei error handling middleware e ami multer specific error dhorte chacchi. Tai if(err instanceof multer.MulterError) kore check korlam amar ‘err’ object ta multer er theke ashce kina.

Ekhon server on kore ekta docs ba pdf file upload korle dekha jabe “Only jpg, jpeg and png format is allowed” error ta show kortese. Eta multer specific error na. ei error ta ami nije throw korechi. Multer specific error holo multer er configuration related error like frontend er input field er name ebong backend er upload.single() er argument er value jodi same na hoy. Kingba file upload korar shomoy jodi thik mimetype 1MB er cheye boro size er kono file upload kori. Tokhon "Upload ErRor" show korbe.

Ami jodi upore cb(new Error("Only jpg, jpeg and png format is allowed") na kore cb(null,false) kore ditam tahole express theke kono error show korto na. silently just file ta upload hoto na. client mone korto file upload hoye geche. Jodio sheta hoy nai server end e.

Etokkhon amra single file er khetre limitation, restriction, error handling dekhlam.

Index.html…

<!DOCTYPE *html*>

<html *lang*="en">

  <head>

    <meta *charset*="UTF-8" />

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

    <title>Express File Upload with Multer</title>

  </head>

  <body>

    <form

*action*="http://localhost:3000/"

*method*="post"

*enctype*="multipart/form-data"

    >

      <input *type*="file" *name*="avatar" />

      <br /><br />

      <input *type*="file" *name*="doc" />

      <input *type*="submit" *value*="Submit" />

    </form>

  </body>

</html>

Ekhon index.html e arekta file type input field nilam ‘doc’ name. ami ‘avatar’ field er jonno image/jpeg allow korte chaitesi ar ‘doc’ er jonno ami chaitesi pdf format allow korte.

Index.js e ‘upload’ er moddhe je fileFilter e je function ta rakhsi ta prottek field (avatar kingba doc) er jonno applicable. Ekhane app.post() upload.fields() boshalam. Etar moddhe ekta array of objects dilam. Object gula ekekta input field ke represent kore ebong fieldgula theke maximum koyta file upload korte parbo sheta bole deya ache. Abar amar duita field er jonno dui rokom mimetype check korte hobe. Jehetu fileFilter er function ta duibar call hobe duita field er jonno so ami eitar moddhei checking ta korte hobe.

Index.js …

*const* express = *require*("express");

*const* multer = *require*("multer");

*const* app = *express*();

*const* UPLOADS\_FOLDER = "./uploads/";

*// prepare the final multer upload object*

*var* upload = *multer*({

  dest: UPLOADS\_FOLDER,

  limits: {

    fileSize: 1000000,

  },

*fileFilter*: (*req*, *file*, *cb*) *=>* {

    if (file.fieldname *===* "avatar") {

      if (

        file.mimetype *===* "image/png" ||

        file.mimetype *===* "image/jpeg" ||

        file.mimetype *===* "image/jpg"

      ) {

*cb*(null, true);

      } else {

*cb*(new *Error*("Only jpg, jpeg and png format is allowed"));

      }

    } else if (file.fieldname *===* "doc") {

      if (file.mimetype *===* "application/pdf") {

*cb*(null, true);

      } else {

*cb*(new *Error*("Only pdf file is allowed"));

      }

    }

  },

});

*//apllication route*

app.*post*(

  "/",

  upload.*fields*([

    { name: "avatar", maxCount: 1 },

    { name: "doc", maxCount: 1 },

  ]),

  (*req*, *res*) *=>* {

    res.*send*("This is file upload tutorial...Yahoo");

  }

);

app.*use*((*err*, *req*, *res*, *next*) *=>* {

  if (err) {

    if (err instanceof multer.MulterError) {

      res.*status*(500).*send*("Upload ErRor");

    } else {

      res.*status*(500).*send*(err.message);

    }

  } else {

    res.*send*("Success");

  }

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Shob thik moto kaj kortese. Ekhane lokkho korar bishoy Prothom field er khetre thik moto thik format er file select korleo jodi 2nd field er khetre vul format er file select kori (vice versa) tahole duita file er konotai upload hobe na.

‘uploads’ folder e upload howa filegular format er thik nai. Extension na thakar karone filefula corrupted dekhacche. User ba client je name file gula upload korbe shei original name er shathe ekta unique number add kore ekekta unique name file gula ‘uploads’ folder e rakhbo. Orthat file gula jekhane store hocche shei storage er upor control chacchi. Multer er diskStorage method use kore ei kaj ta korbo.

Ei method ta ekta option receive kore object akare.

Index.js …

*const* express = *require*("express");

*const* multer = *require*("multer");

*const* path = *require*("path");

*const* app = *express*();

*const* UPLOADS\_FOLDER = "./uploads/";

*// define the storage*

*const* storage = multer.*diskStorage*({

*destination*: (*req*, *file*, *cb*) *=>* {

*cb*(null, UPLOADS\_FOLDER);

  },

*filename*: (*req*, *file*, *cb*) *=>* {

*const* fileExt = path.*extname*(file.originalname);

*const* fileName =

      file.originalname

        .*replace*(fileExt, "")

        .*toLowerCase*()

        .*split*(" ")

        .*join*("-") +

      "-" +

      Date.*now*();

*cb*(null, fileName + fileExt);

  },

});

*// prepare the final multer upload object*

*const* upload = *multer*({

  storage: storage,

  limits: {

    fileSize: 1000000,

  },

*fileFilter*: (*req*, *file*, *cb*) *=>* {

    if (file.fieldname *===* "avatar") {

      if (

        file.mimetype *===* "image/png" ||

        file.mimetype *===* "image/jpeg" ||

        file.mimetype *===* "image/jpg"

      ) {

*cb*(null, true);

      } else {

*cb*(new *Error*("Only jpg, jpeg and png format is allowed"));

      }

    } else if (file.fieldname *===* "doc") {

      if (file.mimetype *===* "application/pdf") {

*cb*(null, true);

      } else {

*cb*(new *Error*("Only pdf file is allowed"));

      }

    }

  },

});

*//apllication route*

app.*post*(

  "/",

  upload.*fields*([

    { name: "avatar", maxCount: 1 },

    { name: "doc", maxCount: 1 },

  ]),

  (*req*, *res*) *=>* {

console.*log*(req.files);

    res.*send*("This is file upload tutorial...Yahoo");

  }

);

app.*use*((*err*, *req*, *res*, *next*) *=>* {

  if (err) {

    if (err instanceof multer.MulterError) {

      res.*status*(500).*send*("Upload ErRor");

    } else {

      res.*status*(500).*send*(err.message);

    }

  } else {

    res.*send*("Success");

  }

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ei object e first e ‘destination’ name ekta property dibo. Ei ‘destination’ ta ami upload variable er ‘dest: UPLOADS\_FOLDER’ --- evabe dibo na. destination er upor aro control amar dorkar. Ekhane ami ‘uploads’ folder ta shurutei create kore nisi. Emon jodi hoto amar folder ta create kora nai. Tahole shekehtre somehow kichu programme likhe folder ta automatically nije nije create howar bebostha kortam. Tai ‘destination’ er upor amar ekta function control lagbe. Tai ei property value hocche ekta function ja req, file, cb parameter akare receive kore. Jehetu ami already manually ‘uploads’ folder create kore rakhsi ebong UPLOADS\_FOLDER er moddhe er path tao save kore rakhsi tai ami automatic folder creation er jhamelay gelam na. ei ‘destination’ er function theke callback deyar shomoy bole dite hobe amar folder ta ki. Ekhane just cb(null, UPLOADS\_FOLDER) kore dilam. Ekhano null bolte error null bujhacchi. Ekhane ullekkho cb er 2nd parameter e agerbarer moto true-false deya jabe na. ekhane folder er path ta expected.

Diskstorage er option object er 2nd property holo ‘filename’. Etar valueo hocche req, file, cb parameter wala ekta function. Ekhane ‘file’ parameter ta use kore file.originalname access korbo. Etate extension shoho file er name ta ache. Client er file er name jodi hoy ‘leo messi.jpg’ tahole amar target holo file ta ke ‘leo-messi-145798635.jpg’ --- ei akare ‘uploads’ folder e save kora. Etar programme ekhon likhbo.

First e extension part alada korbo. Nodejs er path module ta use kore fileExt variable er moddhe extension part tuku rakhlam.

Erpor file er namer 1st part tuku banabo. Then callback er moddhe error null dilam ebong 2nd parameter file name er 1st part + extension part kore pura file er name ta dilam.

Erpor ‘upload’ variable er ‘dest’ object ta shoraya ‘stoarage’ name ekta property diye dibo. Ebong etar value hobe amader banano ‘storage’ variable ta.

File upload howar por process ta app.post er middleware e jay. Shekhane amader browser e lekha ashtese “ This is file upload tutorial...Yahoo ” . amara chaile ei middleware console.log(req.files) kore upload howa files er info dekhte pari. Single file hole req.file.