**Error Handling**

Express js e kivabe error handling kora uchit she bepare details alochona kora hobe ajke. Error thik moto handle na korle production e giye amar application user er kono action er karone crash korte pare. Jehetu node ekta alltime running server shehetu kono error er karone server crash korle sheta valo kono experience na.

Index.js …

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

*const* app = *express*();

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

  res.*send*("Hello World");

});

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

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

});

Ei setup niye kaj shuru korlam. Express e synchronous ebong asynchronous code er khetre dui vabe error handling korte hoy.

First of all synchronous code er ta dekhbo. Synchronous code er jonno express nijei amader jonno error handle kore dey.

Index.js …

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

*const* app = *express*();

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

  throw new *Error*("There has been an error.");

});

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

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

});

Ekhan theke response send na kore error throw kore dilam. Server on kore localhost:3000/ hit korlam. Amader application crash korlo na. user error reponse pacche. Application running ache. Eta possible hoise cz express by default amader error ta handle kore dise. Na korle amader application crash korto ebong user further request korle response peto na. erokom error throw na kore jodi amader application e kono unintentional error hoy nicher moto tahole ki hobe?

Index.js …

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

*const* app = *express*();

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

  res.*send*(a);

});

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

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

});

Ekhane app.get() e ‘a’ name ekta variable user ke response dite chacchi jeta exist kore na. ekhane error hobe. Jehetu eta synchronous code tai express etao handle kore dibe ebong user response pabe.

Ekhon dekhi express kivabe amader error ta handle korlo. jokhoni express amader kono synchronous code theke error pay immediately oi jayga theke ekta default error handling middleware er kache pathay dey ebong shei middleware function tai error ta receive kore, process kore kore, application ke crash hote dey na ebong user ke response dey. Express er ei default error handling middleware tar characteristic holo eta amader application er last middleware hishebe invisibly boshe thake. At least etake ekhon amra evabei vabte pari. By default ei error handling middleware behaviour holo eta client ba user ke error stack ta purata print kore deya. Etokkhon ja postman e dekhlam. Eta only she development environment e kore. Amra jodi amader node application ta production environment e run kori tahole erokom stack user dekhbe na. sheta check korar jonno package.json e nicher code lekhlam

Package.json…

{

  "scripts": {

    "start": "nodemon index.js",

    "production": "NODE\_ENV=production nodemon index.js"

  },

  "dependencies": {

    "body-parser": "^1.20.0",

    "cookie-parser": "^1.4.6",

    "ejs": "^3.1.8",

    "express": "^4.18.1"

  }

}

Then terminale ‘npm install -g win-node-env’ kore package ta install korlam. Then ‘npm run production’ kore postman e localhost:3000/ url e hit korlam. Ekhon ar ager moto postman e bishal error stack ashlo na . just lekha ashlo “Internal server error”. Ebong error status o set kore dise 500 cz eta ekta server side er syntax error. So amra default error handler er behaviour ta dekhlam. What If ami ei default behaviour ta change korte chai? Amra jodi iccha moto error message , error status dite chai, nijeder moto error receive kore nijeder moto customize korte chai then ki korbo? Ektu age je invisible error handling default middleware er kotha bolsilam shetakei explicitly amader application er last middleware hishebe likhe dite hobe. Orthat overwrite korte hobe. Eta ke obosshoi last middleware function hote hobe.

Index.js

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

*const* app = *express*();

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

  res.*send*(a);

});

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

  console.*log*(err);

})

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

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

});

Erpor jokhoni amader application e synchronous kono code server side error hobe tokhon thik jekhane error hoise shekhan theke exit korbe ebong porer shob code skip kore ei function er vitor chole ashbe. Ei function er ‘err’ parameter er moddhe amra error object ta pabo. Apatoto sheta amra console.log() korlam. Postman e localhost:3000/ gele kono response ashce na. jehetu res.send() ba ei type kichu kore client ke kono response dei nai. Just console.log(err) kore boshe achi. Tai client loading pacche. Tai ekhon res.send() kore response pathay dibo.

Index.js …

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

*const* app = *express*();

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

  res.*send*(a);

});

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

  console.*log*(err);

  res.*send*("There was an error");

});

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

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

});

Postman e response er status ta 200 orthat success. Cz ekhon amar error handle function kaj kortese. Amra res.send() use korechi. Kono status bole na diye response send korle express by default status code 200 dhore ney. Ekhane res.status(500).send(“There was an error.”) dile status 500 ashbe. Ektu age je kaj gula express er default error handler nijei kortesilo shetai ekhon amader manually kore dite hobe. Ei ‘err’ object ta ekta common javascript object. Etar ekta property holo ‘message’.

Index.js …

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

*const* app = *express*();

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

  res.*send*(a);

});

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

  if (err.message) {

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

  } else {

    res.*status*(500).*send*("There was an error.");

  }

});

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

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

});

Ekhane ‘err’ object er kono message property thakle 500 status code shoho message ta response akare send korlam. Otherwise amar nijer ekta lekha response hishebe send korlam.

ekhon amar application e programmer er error hok ba onno error hok sheta handled hoye jabe. Application hang or crash korbe na. user kono na kono response pabe.

but client jodi emon kono route e hit kore jeta exist kore na then? Tahole express amader 404 error debe. Ei error ta amader error handling function receive korbe na. karon 404 thik error er porjaye pore na. eta just not found. User er vul. Server ba programmer er vul na. ekarone 404 holei express nije sheta handle kore user ke ekta default response diye dicche. Amra ei jinish ta dhore customize korte chaile tahole amader shob route handling middleware er pore ekta normal middleware likhte hobe eivabe…

index.js ..

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

*const* app = *express*();

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

  res.*send*(a);

});

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

  res.*status*(404).*send*("Rquested URL was not found.");

});

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

  if (err.message) {

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

  } else {

    res.*status*(500).*send*("There was an error.");

  }

});

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

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

});

Jokhoni keu emon url e hit korbe ja amader route e nai tokhon express por por shob route path check korte korte jabe. Tarpor she ei middleware e dhuke jabe.

Then ekhan theke amra chaile user ke response diye dite parbo. Othoba amra etake abar amader porer default error handling middleware er kache next() call kore pathay dite parbo. Ami ekhan thekei response diye dilam user ke. Status diye dilam 404. Now localhost:404/about e hit korle postman e lekha ashtese “request url…..found”. status code 404. Amra jodi response ta ekhane na diye porer middleware e pathay dite chai tahole next(“Requested URL was not found.”) kore amader error handling middleware er kache pathaya dilam. Next function er kono parameter text hishabe deya mane holo error shoho porer middleware e pathano.

Ekhon response streaming er khetre error handling dekhbo. Amra jani, amra nodejs e request stream akare receive kori ebong response o amra stream akare pathate pari.

Index.js …

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

*const* app = *express*();

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

*for*(*let* i=0; i<=10; i++){

    res.*write*('a')

  }

  res.*end*()

});

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

*next*("Rquested URL was not found.");

});

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

  if (err.message) {

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

  } else {

    res.*status*(500).*send*("There was an error.");

  }

});

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

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

});

Amra ekhane loop kore kore ekta kore ‘a’ response e write korchi ebong finally response end korchi. Amra shundor moto response pacchi. Ekhon jodi response e write korte korte kono ekta jaygay error hoy….

Index.js …

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

*const* app = *express*();

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

*for* (*let* i = 0; i <= 10; i++) {

    if (i *===* 5) {

*next*("there was an error");

    } else {

      res.*write*("a");

    }

  }

  res.*end*();

});

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

*next*("Rquested URL was not found.");

});

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

  if (err.message) {

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

  } else {

    res.*status*(500).*send*("There was an error.");

  }

});

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

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

});

Dhori, 5 no iteration e error holo. Amra next() call kore dilamo. Ekhon localhost:3000/ e hit korle console e ekta error dekhtesi. Sheta holo “headers already sent”. Etar karon ki? 5 no iteration e error howar por sheta amader error handling middleware er kache jabe. Ekhon middleware ta handle kore error ta response akare pathanor cheshta korbe kintu sheta korar agei response e write hoye gese 9 no line e. response e jokhoni write hoye jay tar mane client ke stream kora shuru hoye gese. Jotokkhon end na kora hocche totokkhon porjonto stream hote thakbe. tar mane response er shathe je headers gula thake shegulao client ke pathano hoye gese ekbar. Ekta request e client ke response er headers multiple times pathano jay na. abar jehetu ekhane response pathanor cheshta kora hocche ei shomoy abar response er header set korar cheshta kora hocche ashole. Ekaronei express ‘already header sent’ error ditese. Jodi amra ekbar response e write kora shuru kore dei tahole etar mane hocche headers ekbar client ke pathaya disi. Abar jeno ami na pathai. Jokhon amar request ta end hoy nai, request er header pathano start kore disi ebong request end howar agei ami konovabe header send korar try korchi (res.send() call korchi ba res.status().send() call korchi --- egulao headers set kore) tokhoni error ta ashbe. Eta amader error handling middleware handle korte hobe following way te.

Index.js …

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

*const* app = *express*();

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

*for* (*let* i = 0; i <= 10; i++) {

    if (i *===* 5) {

*next*("there was an error");

    } else {

      res.*write*("a");

    }

  }

  res.*end*();

});

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

*next*("Rquested URL was not found.");

});

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

  if (res.headersSent) {

*next*("There was a problem");

  } else {

    if (err.message) {

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

    } else {

      res.*send*("There was an error");

    }

  }

});

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

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

});

Ekhane headersSent property ta use korsi. Headers already pathano hoye gele eta true ashbe. True hole next() call kore express er error handler er kache pathaya dibo. Otherwise custom error message pathabo. Ekhon localhost:3000/ e hit korle dekhte parbo console e error stack nai.

Arekta jinish dekhte pacchi, amader banano custom error handler er vitor thekeo jodi next error call kori tahole sheta express er default invisible error handler er kache jacche. Ei invisible handler ta amader banana custom handler ero niche boshe ache.

Ekhon amra dekhbo kivabe asynchronous code er khetre error handled hobe.

Index.js …

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

*const* app = *express*();

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

  res.*send*("Hello World");

});

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

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

});

Index.js ekdom shurur obosthay niye gelam. Dhorlam, “/” route er handler function ta vitor theke fs module er readFile() ke call kore ebong tate emon ekta file path dilam jeta exist kore na.

Index.js …

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

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

*const* app = *express*();

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

  fs.*readFile*("/file-doesnt-exist", (*err*, *data*) *=>* {

  });

});

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

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

});

Jehetu file ta exist kore na shehetu ekhane error pawar kotha. Error ta amra ei err er moddhe pabo. Ei readFile() function ta holo async function jar kaj holo ei path er file ta read kore data pele ei ‘data’ er moddhe data ta diye dibe. Otherwise error hole ‘err’ er moddhe amra sheta pabo. File system theke data read korbe asynchronous way te. Ei obosthay localhost:3000/ hit korlam. Postman e kono response pacchi na. hang hoye ache. Orthat express ei error ta amader ke default way te handle kore dei nai jeta synchronous code er khetre diyechilo. Ekhon readFile er jaygay readFileSync dile express tar default invisible error handler middleware er maddhome error ta handle ebong client ke ekta response dicche.

Keno express sync er khetre nije handle kore, async er khetre kore na? karon synchronous code er kaj por por hoy. Ekta kaj purapuri complete howar por next kaje jay. But async code er kaj por por hoy na. ekekta code er kaj ekek shomoy shesh hoy. So kono async code er error ashle express jodi sheta handle kore dito tahole amra incosnsistent behaviour petam. Suppose ekhane readFile er por aro kichu code ache. Jodi readFile er CB theke express error diye ber hoye ashto porer line er code execute hoto na.

So jehetu express eta handle kore na tai amader kichu korte hobe. Amra chaile manually express er kache error ta pathaya dite pari handle korar jonno. Niche eta dekhano holo.

Index.js …

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

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

*const* app = *express*();

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

  fs.*readFileSync*("/file-doesnt-exist", (*err*, *data*) *=>* {

    if (err) {

*next*(err);

    } else {

      res.*send*(data);

    }

  });

});

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

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

});

Evabe next er moddhe err pathaya diye call kore. Tahole express er default invisible error handler ba amader banano kono custom error handler sheta catch korte parbe. now postman e localhost:3000/ hit korle ar ager moto hang kortese na. express er default error handler error handle hocche. Ekta response pacchi.

Ekhon ekta custom error handler banabo.

Index.js …

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

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

*const* app = *express*();

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

  fs.*readFileSync*("/file-doesnt-exist", (*err*, *data*) *=>* {

    if (err) {

*next*(err);

    } else {

      res.*send*(data);}});});

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

  console.*log*("I am not called");

*next*();

});

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

  if (res.headersSent) {

*next*("THere was a PROblem");

  } else {

    if (err.message) {

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

    } else {

      res.*send*("THERE was an ERRoR");}}});

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

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

});

Error handled hocche. Arekta jinish important. App.get() er por arekta middleware ache ekhane. readFile theke next(err) call korsi. Er fole process ta direct error handler middleware e chole gese. Holud arrow dara chinnito middleware e jay nai. “I am not called” console eo nai.

Etokkhon amra dekhlam built in asynchronous function er kaj. Ekhon nijeder lekha asynchronous function er error handling dekhbo.

Index.js …

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

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

*const* app = *express*();

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

*setTimeout*(() *=>* {

    try {

      console.*log*(a);

    } catch (err) {

*next*(err);

    }

  }, 2000);

});

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

  console.*log*("I am not called");

*next*();

});

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

  if (res.headersSent) {

*next*("THere was a PROblem");

  } else {

    if (err.message) {

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

    } else {

      res.*send*("THERE was an ERRoR");

    }

  }

});

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

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

});

Ekhane app.get() er “/” route ekta setTimeout chalaisi. Settimeout er moddhe try catch block use kore ‘a’ er value console e dekhte cahcchi. Error hole catch blocke next(err) kore error handler er kache pathaya ditesi.

Ekhon localhost:3000/ hit korle error handled hocche. Postman e lekha ashtese ‘a is not defined.’ Ekhane try catch na korle error dhorte partam na. application crash korto.

Lastly arekta jinish, amra chaile asynchronous code ke komaya synchronous code er default error handling er advantage ta nite pari. Sheta korte hobe middleware chaining kore.

Index.js …

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

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

*const* app = *express*();

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

  fs.*readFile*("/filedoesntexist", "utf-8", (*err*, *data*) *=>* {

    console.*log*(data);

*next*(err);

  });

});

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

  console.*log*("I am not called");

*next*();

});

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

  if (res.headersSent) {

*next*("THere was a PROblem");

  } else {

    if (err.message) {

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

    } else {

      res.*send*("THERE was an ERRoR");

    }

  }

});

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

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

});

Ekhane readFile async function er vitore data ta console.log(data) korte giye error dibe. Jehetu er pore amra next(err) call kore express error handler er kache forward kore disi tai she sheta handle kore dibe. But jodi er porer line e error hoy, suppose next(err) er pore ami console.log(data.property) access korte chacchi ---- tokhon error hobe ebong application crash korbe. ekhane prothom error (file na pawar error) handled hobe thiki. Kintu erpore aro kaja baki ache. Eta async code. Ebong ei error ta amra express ke forward kori nai.

Solution hote pare, amra next(err) er porer line gula porer synchronous middleware function e chain kore dite pari evabe …

Index.js…

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

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

*const* app = *express*();

app.*get*("/", [

  (*req*, *res*, *next*) *=>* {

    fs.*readFile*("/filedoesntexist", "utf-8", (*err*, *data*) *=>* {

      console.*log*(data);

*next*(err);

    });

  },

  (*req*, *res*, *next*) *=>* {

    console.*log*(data.property);

  },

]);

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

  console.*log*("I am not called");

*next*();

});

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

  if (res.headersSent) {

*next*("THere was a PROblem");

  } else {

    if (err.message) {

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

    } else {

      res.*send*("THERE was an ERRoR");

    }

  }

});

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

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

});

Amra jani, amra middleware function evabe array akareo chain korte pari. Ekehtre porer error ta jehetu synchronous code er moddhe ache ei error ta amader manually handle na korleo cholbe. Cz sync code er error express automatically default error handler diye handle kore dey.

Ekhon localhost:3000/ korle application crash kore na. postman e error response ashe. Console e data.property undefined dekhacche.