WEB TECHNOLOGIES

Express JS

Shruthi L

Department of Computer Science and Engineering.

WEB TECHNOLOGIES

Express JS

Shruthi L

Department of Computer Science and Engineering

ERROR HANDLING

Express JS

- ***Error Handling*** refers to how Express catches and processes errors that occur both synchronously and asynchronously

- The success or failure of any REST API call is typically reflected in the HTTP status code.
  - E.g., 400 Bad Request status code or syntactically incorrect requests.
  - 422 Unprocessable Entity

Express JS

The default error handler

- Express comes with a built-in error handler that takes care of any errors that might be encountered in the app.
- This default error-handling middleware function is added at the end of the middleware function stack.

- If you pass an error to next() and you do not handle it in a custom error handler, it will be handled by the built-in error handler;
  the error will be written to the client with the stack trace.

ERROR HANDLING

Express JS

- When an error is written,

the following information is added to the response:

  - The res.statusCode is set from err.status (or err.statusCode).
  - The res.statusMessage is set according to the status code.
  - The body will be the HTML of the status code message
  - Any headers specified in an err.headers object.

WEB TECHNOLOGIES
ERROR HANDLING

Express JS

- Handling errors in the routes itself.

```
app.use(function(err, req, res, next) {
console.error(err.stack);
 res.status(500).send('Something broke!’);
 });
```

- The error handling middleware allows us to separate our error logic and send responses accordingly.

WEB TECHNOLOGIES
ERROR HANDLING

Express JS

Explicit error creation:

```
    app.get('/', function(req, res){
    if(error_condition){
       //Create an error and pass it to the next function
       var err = new Error("Something went wrong");
       next(err); //first error handling middleware
     }
     else {
       res.send(“everything fine”);
       next();
     }
    });
```

Error handling example

```
var express= require("express")
var app= express();

app.get("/student/:id",function(req,res,next)
{
    if(req.params.id.indexOf("2000")!=-1){
      var err = new Error("Something went Wrong!!!")
        next(err)
    }
    else{

        res.send("Everything is fine with"+req.params.id)
        next();
    }
})
app.use("/student",function(req,res){
        console.log("I am Fine")
})
    app.listen(8000,function(){

        console.log("Server is listening on port 3000")
```

The indexOf() method returns the position of the first occurrence of a value in a string.
The indexOf() method returns -1 if the value is not found.

If id is 2000, it returns 0,
Other than 2000, it returns -1

THANK YOU

Shruthi L

Department of Computer Science Engineering