**Error Handling (fail gracefully)**

**Types of errors:**

* Technical/Network
  + MongoDB server is down - Show error page to user
* Expected
  + File can’t be read or DB operation fails - Inform user possibly retry
* Bugs/Logical
  + User object used when it doesn’t exists - Fix during development

**Working with Errors**

**Error is thrown**

Synchronous Code - try/catch

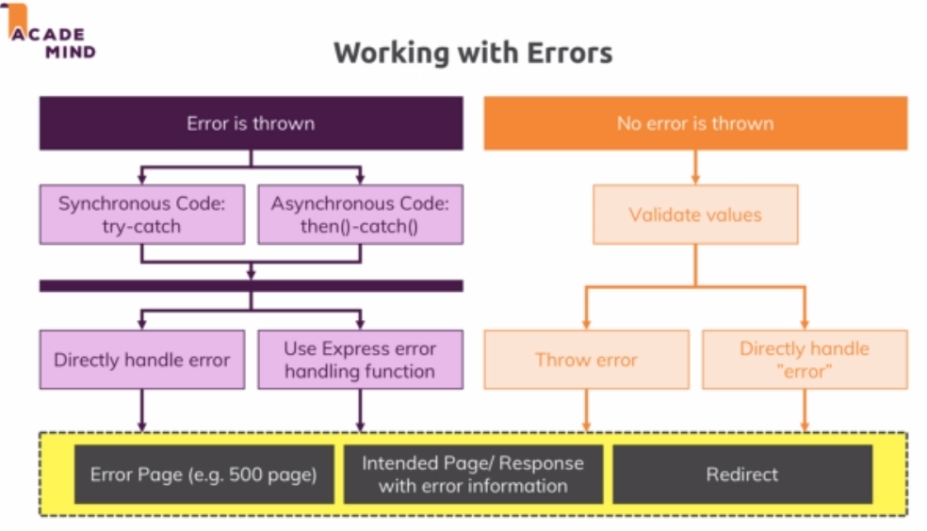
Asynchronous Code - then/catch

> Directly handle error/Use Express error handling function

**No error is thrown**

Validate values

> Throw error/Directly handle error



**Using the express Error Handling Middleware**

All the places where we have catch(synchronous - try-catch, Asynchronous-then,catch) we can add the below code inside catch block:

.catch(err => {

const error = new Error(err);

    error.httpStatusCode = 500;

    return next(error);

});

Here we see a special type of next accepting the argument. That is basically an error handling middleware which is called right away skipping all middlewares when we encountered a next with argument.

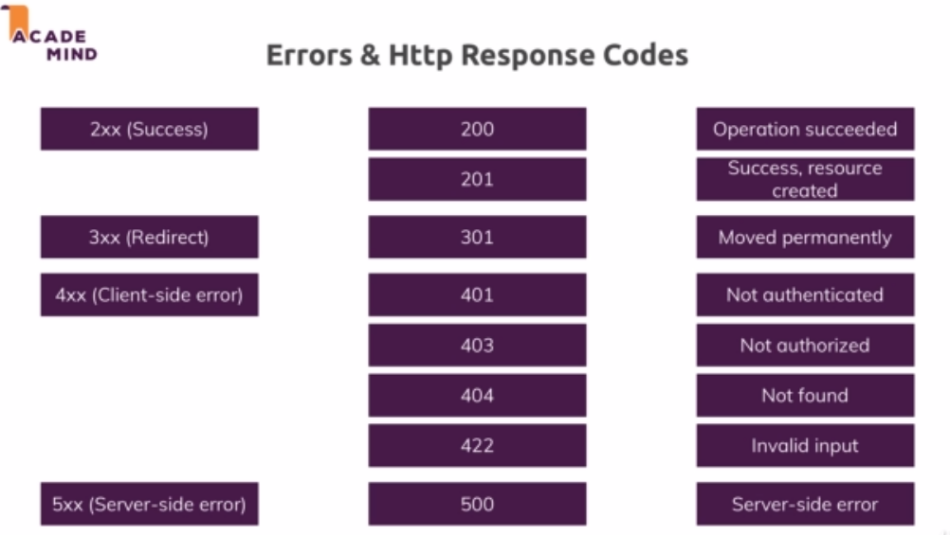
**Using the Error Handling Middleware Correctly**

Inside the promise then/catch or inside of callbacks we cannot throw an error. So we have to use below code:

*.catch(err = {*

*next(new Error(err))*

*})*



<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

**Useful resources:**

Error Handling in Express.js -

Official Docs: [https://expressjs.com/en/guide/error-handling.html](https://expressjs.com/en/guide/error-handling.html" \t "https://www.udemy.com/course/nodejs-the-complete-guide/learn/lecture/_blank)