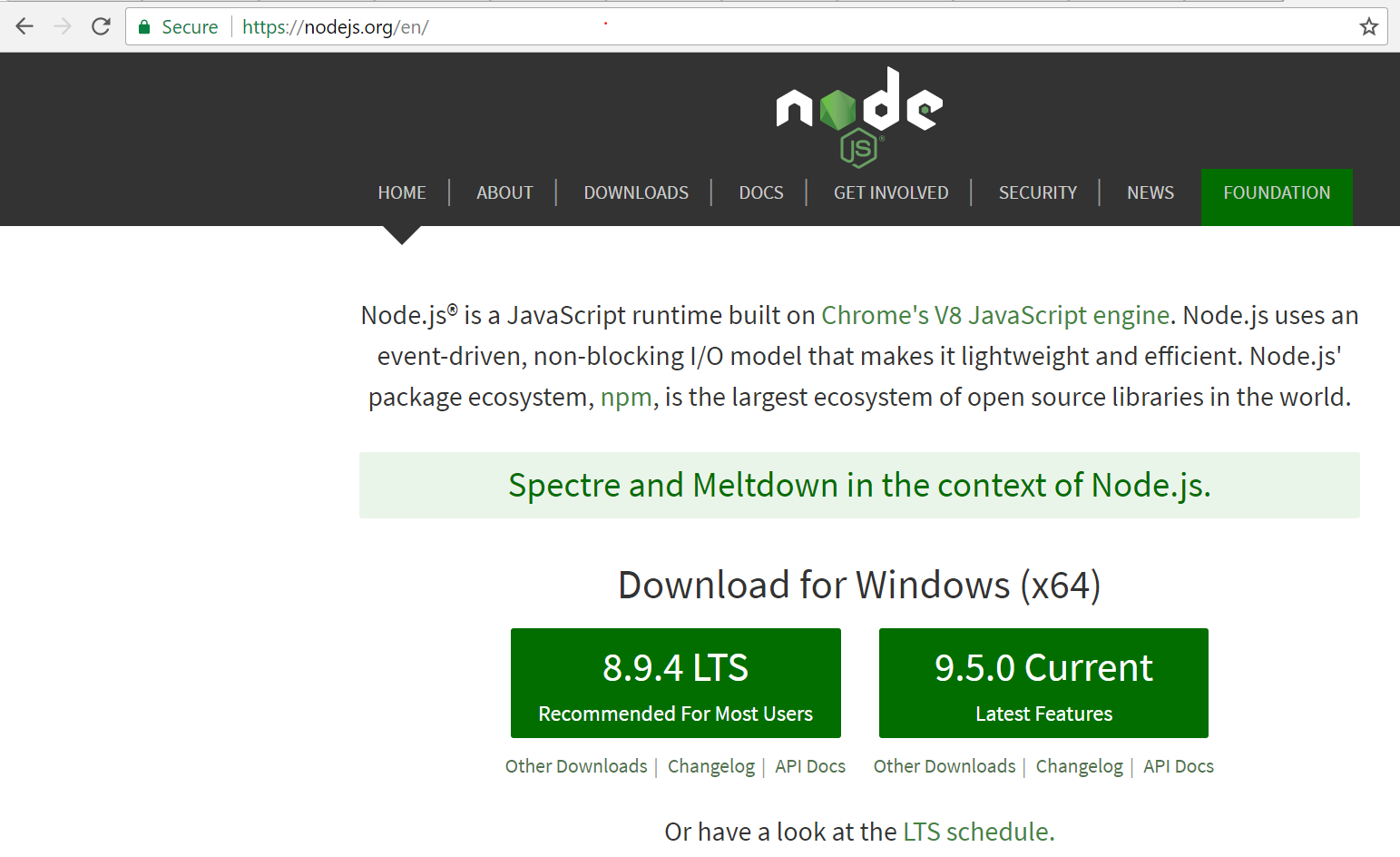
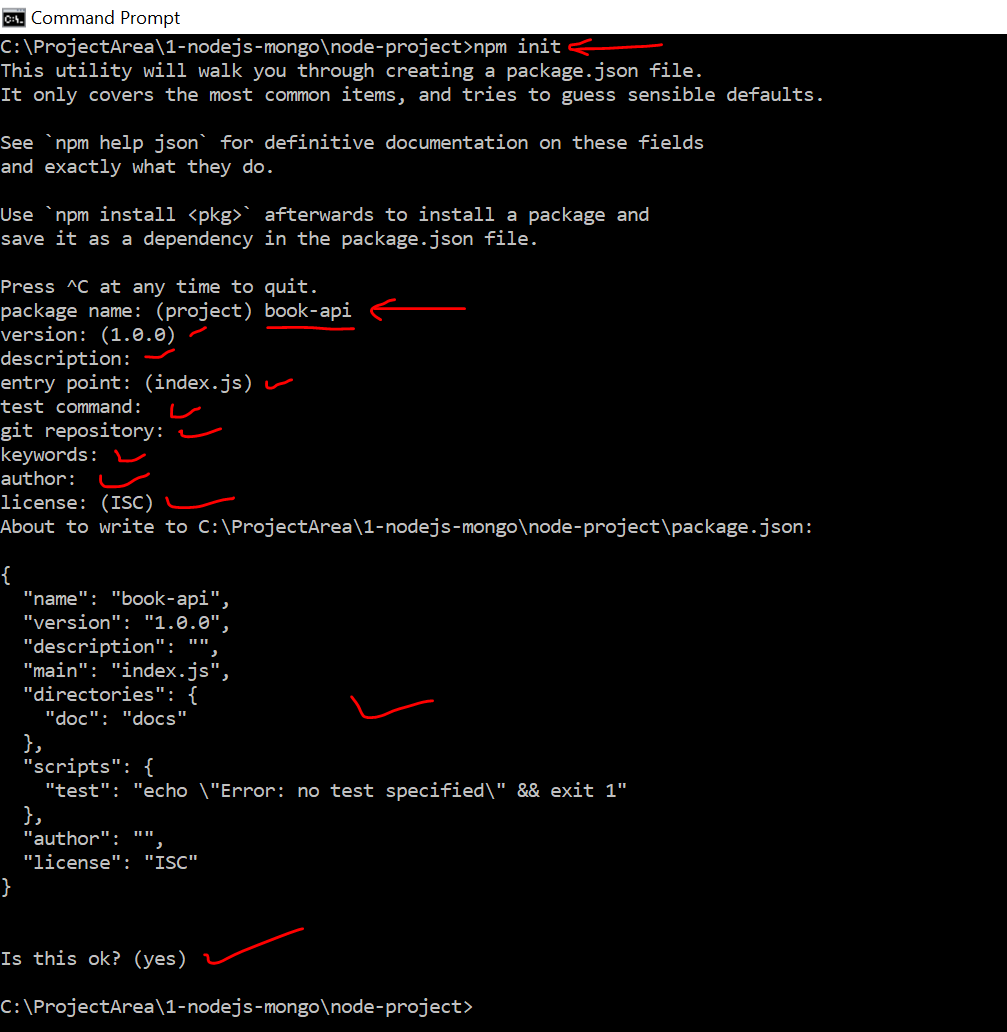
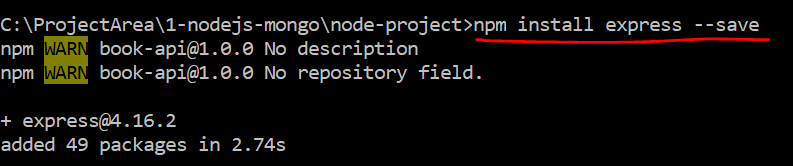
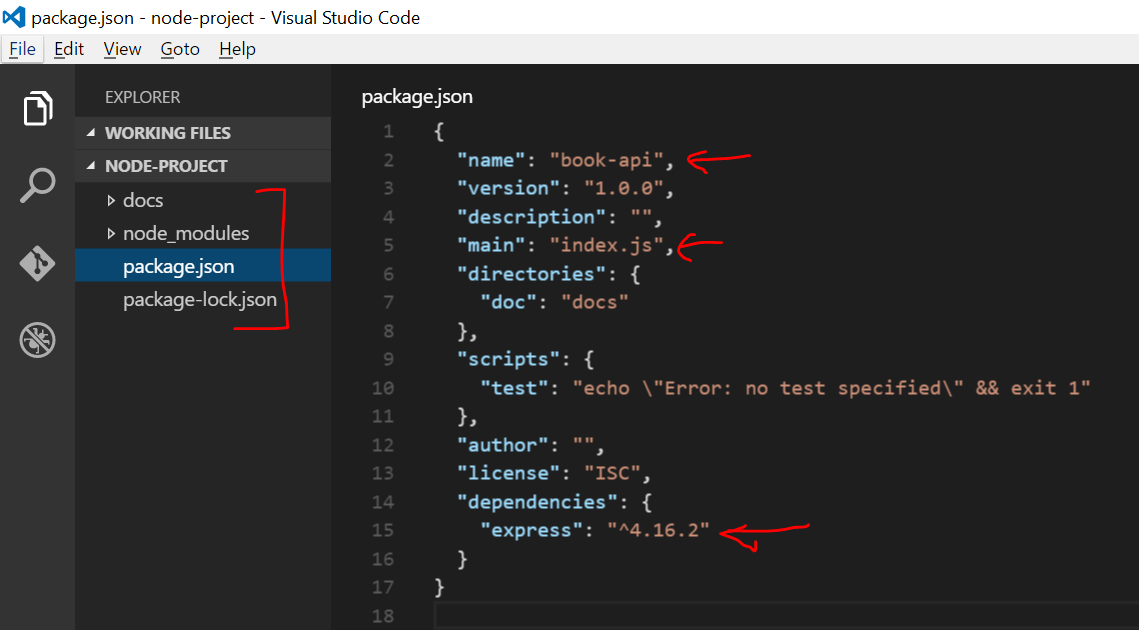
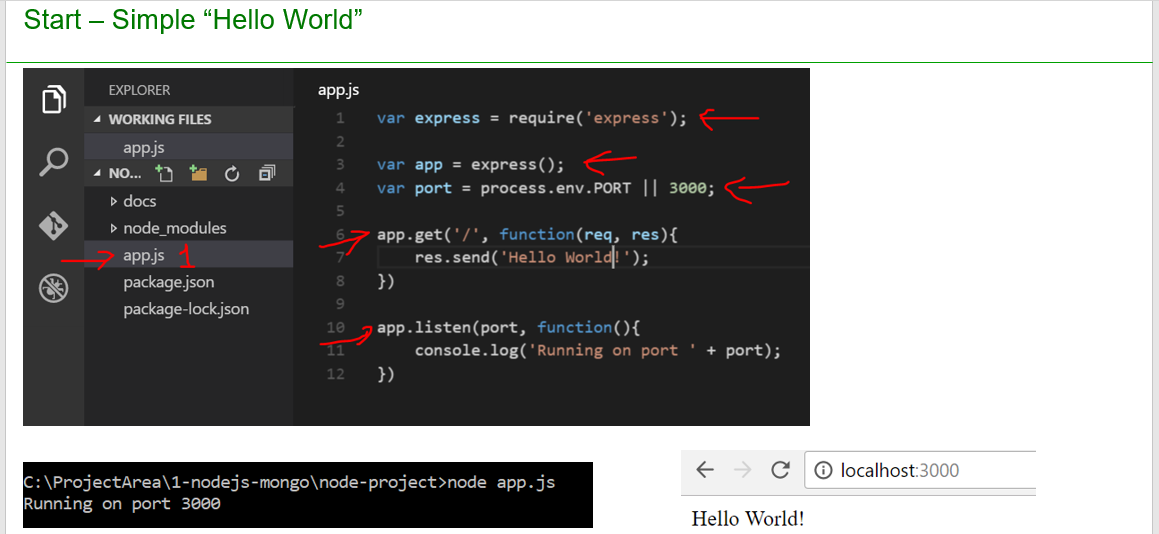
Installation

1. STEP1 – Install node from nodejs.org
   1. 
2. **STEP3 – npm init from any project folder**
   1. 
3. **STEP3 – Install Express (Notice package.json after the command)**
   1. 
4. **STEP4 – Open the code is editor and notice..** 
   1. 
5. Simple ‘Hello World’
   1. 
   2. Open app.js and past the below code

var express = require('express')

var app = new express()

var port = 3000

app.get('/', function(req,res){

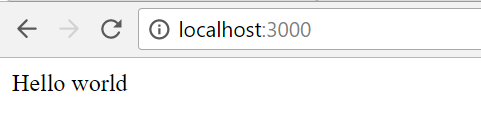
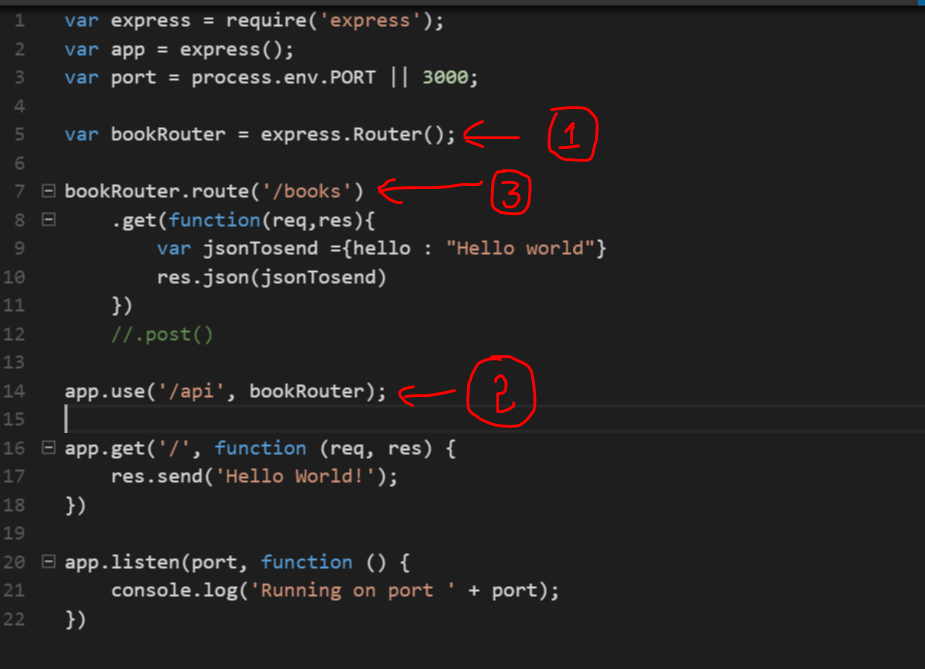
res.send('Hello world')

})

app.listen(port,function(){

console.log('Running on port' + port)

})

1. Go to project folder and run below command
   * 1. Node app.js
2. Type localhost:3000 is chrome(browser) to get the message
   1. 
3. GET API using router
   1. 
   2. Paste the below code in app.js and run http://localhost:3000/api/books

var express = require('express');

var app = express();

var port = process.env.PORT || 3000;

var bookRouter = express.Router();

bookRouter.route('/books')

.get(function(req,res){

var jsonTosend ={hello : "Hello world"}

res.json(jsonTosend)

})

//.post()

app.use('/api', bookRouter);

app.get('/', function (req, res) {

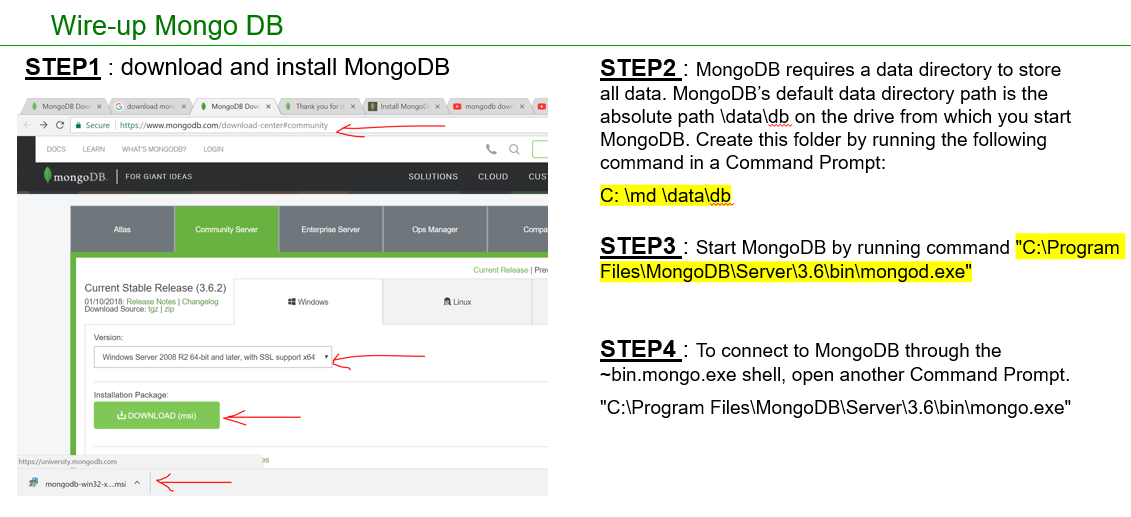
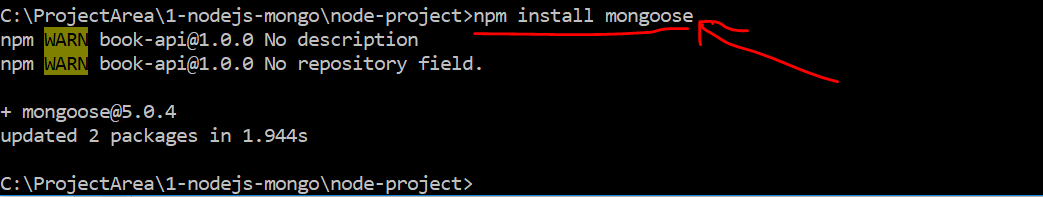
res.send('Hello World!');

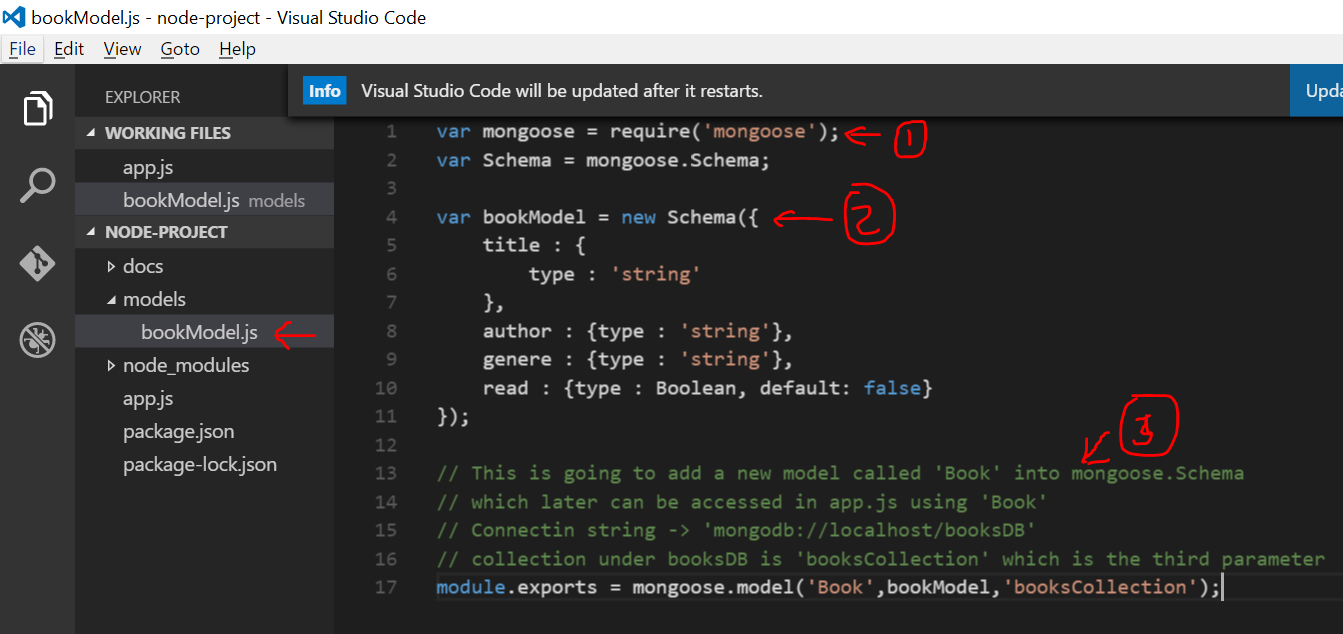
})

app.listen(port, function () {

console.log('Running on port ' + port);

})

1. Wire up MongoDB
   1. 
2. GET API using Books model and mongoose
   1. Install mongoose using npm install mongoose
   2. 



* 1. Add a file bookModel.js and paste below code

var mongoose = require('mongoose');

var Schema = mongoose.Schema;

var bookModel = new Schema({

title : {

type : 'string'

},

author : {type : 'string'},

genere : {type : 'string'},

read : {type : Boolean, default: false}

});

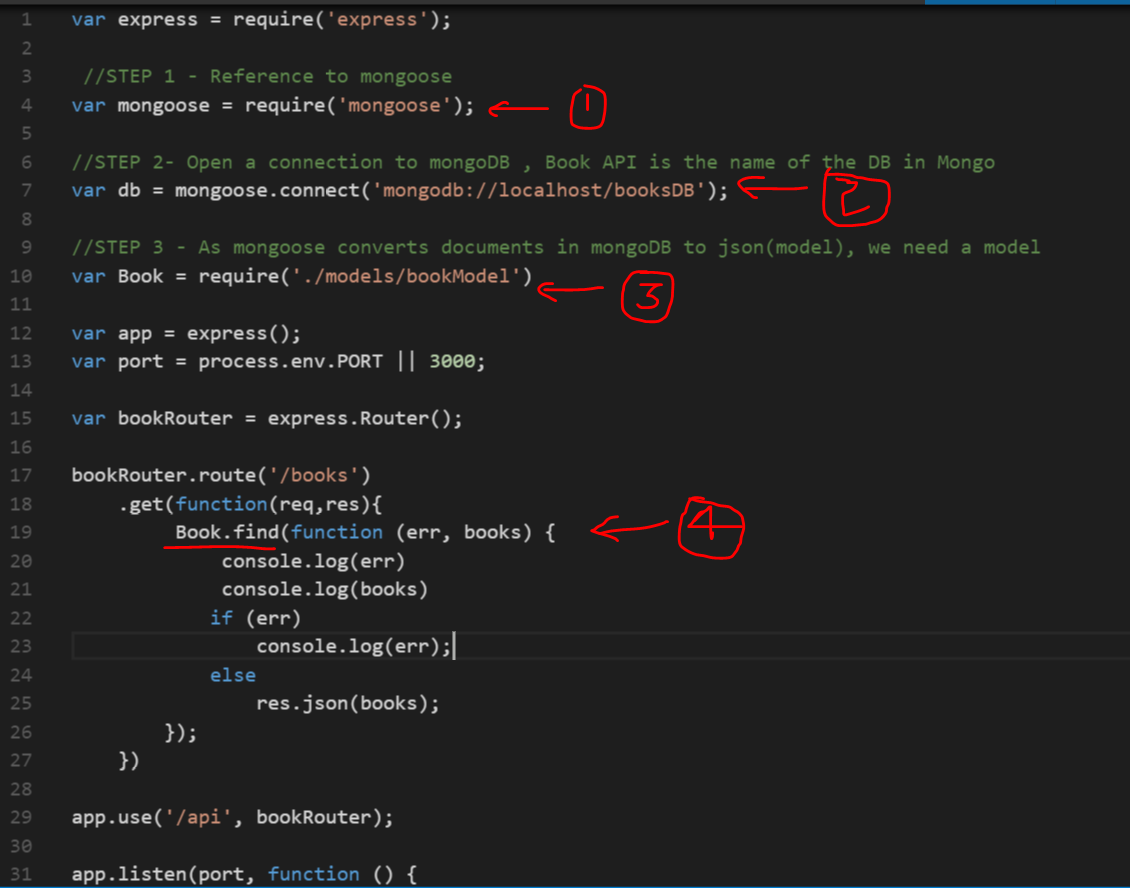
// This is going to add a new model called 'Book' into mongoose.Schema

// which later can be accessed in app.js using 'Book'

// Connectin string -> 'mongodb://localhost/booksDB'

// collection under booksDB is 'booksCollection' which is the third parameter

module.exports = mongoose.model('Book',bookModel,'booksCollection');

* 1. Update app.js with below code
  2. 

var express = require('express');

//STEP 1 - Reference to mongoose

var mongoose = require('mongoose');

//STEP 2- Open a connection to mongoDB , Book API is the name of the DB in Mongo

var db = mongoose.connect('mongodb://localhost/booksDB');

//STEP 3 - As mongoose converts documents in mongoDB to json(model), we need a model

var Book = require('./models/bookModel')

var app = express();

var port = process.env.PORT || 3000;

var bookRouter = express.Router();

bookRouter.route('/books')

.get(function(req,res){

Book.find(function (err, books) {

console.log(err)

console.log(books)

if (err)

console.log(err);

else

res.json(books);

});

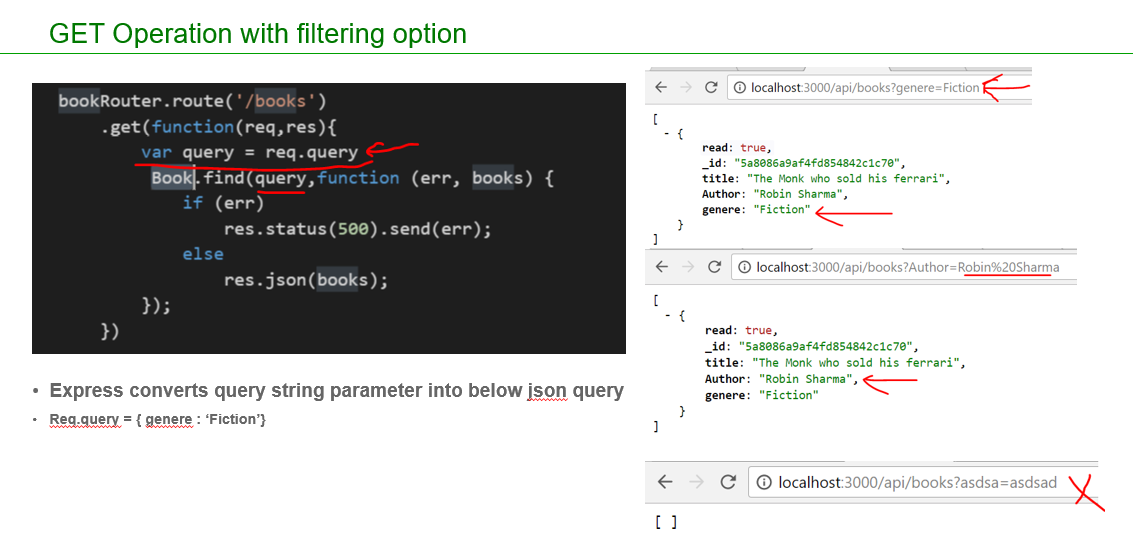
})

app.use('/api', bookRouter);

app.listen(port, function () {

console.log('Running on port ' + port);

})

1. GET API with filtering
   1. 
   2. Place the below code appropriately in app.js ( do not overwrite entire app.js)

bookRouter.route('/books')

.get(function(req,res){

var query = req.query

Book.find(query,function (err, books) {

if (err)

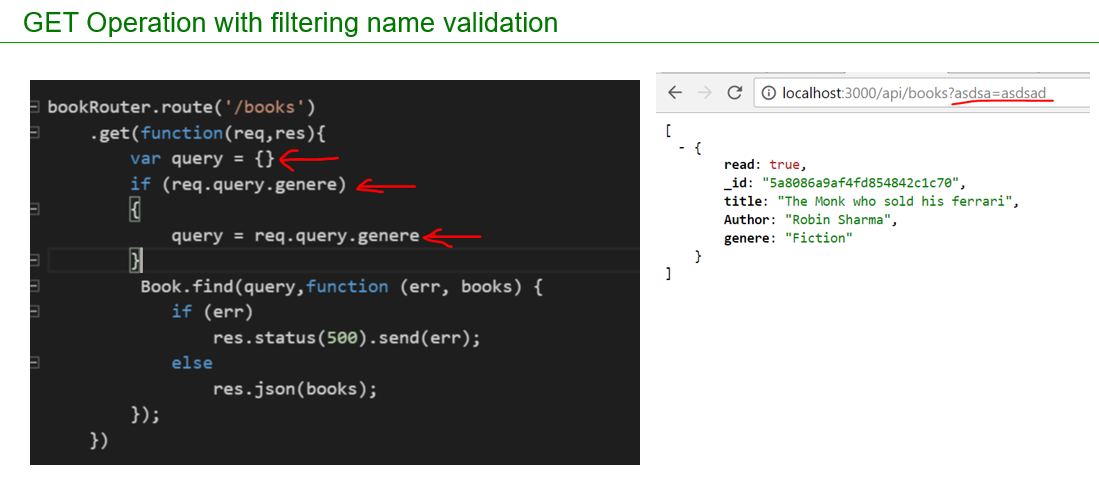
res.status(500).send(err);

else

res.json(books);

});

})

1. GET API with filter name check
   1. 
   2. Place the below code appropriately in app.js ( do not overwrite entire app.js)

bookRouter.route('/books')

.get(function(req,res){

var query = {}

if (req.query.genere)

{

query = req.query.genere

}

Book.find(query,function (err, books) {

if (err)

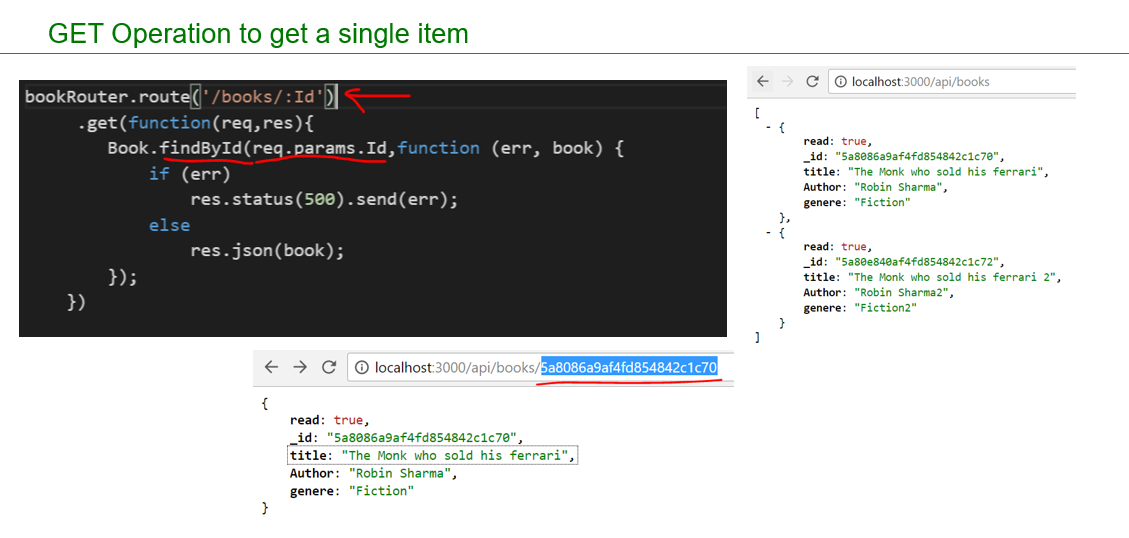
res.status(500).send(err);

else

res.json(books);

});

})

1. GET API to get a single item
   1. 
   2. Place the below code appropriately in app.js ( do not overwrite entire app.js)

bookRouter.route('/books/:Id')

.get(function(req,res){

Book.findById(req.params.Id,function (err, book) {

if (err)

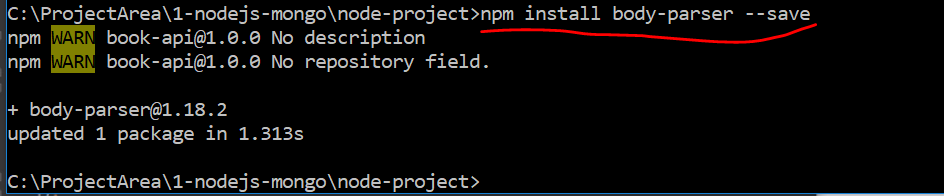
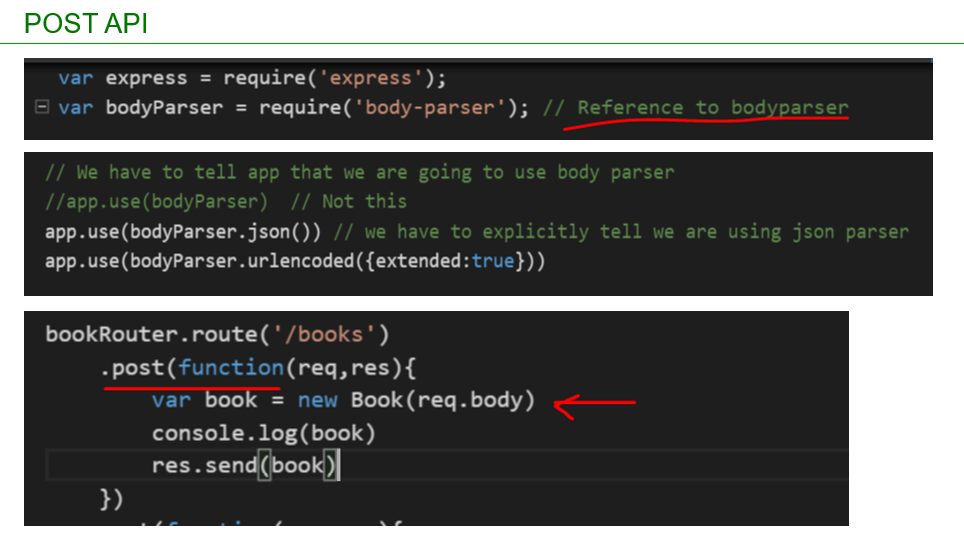
res.status(500).send(err);

else

res.json(book);

});

})

1. Posting data
   1. Install body parser package which is required to read the posted data and convert them to json
   2. 
   3. POST API code changes in app.js
   4. 
   5. Place the below code appropriately in app.js ( do not overwrite entire app.js)

var bodyParser = require('body-parser'); // Reference to bodyparser

// We have to tell app that we are going to use body parser

//app.use(bodyParser) // Not this

app.use(bodyParser.json()) // we have to explicitly tell we are using json parser

app.use(bodyParser.urlencoded({extended:true}))

bookRouter.route('/books')

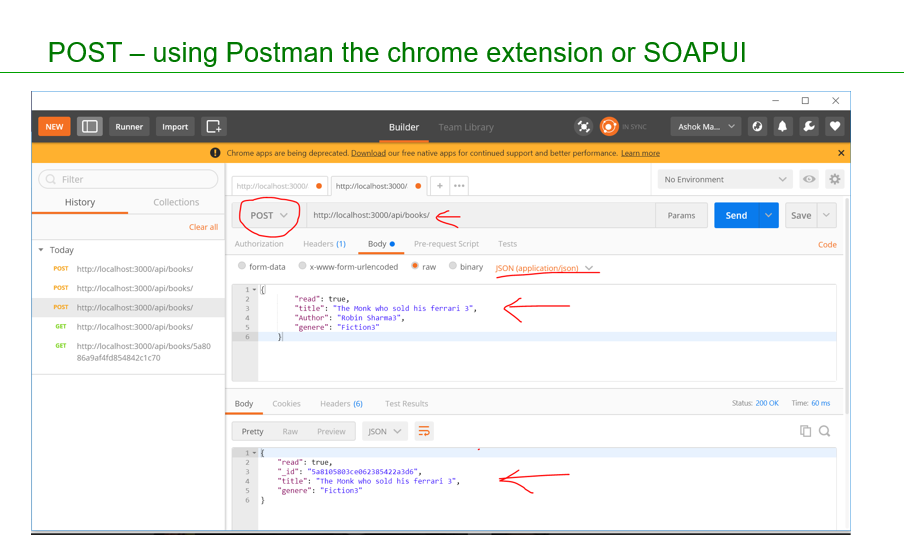
.post(function(req,res){

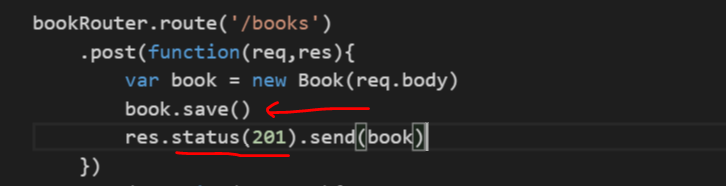
var book = new Book(req.body)

console.log(book)

res.send(book)

})

* 1. Insert below json into DB
     1. {
     2. "title": "The Monk who sold his ferrari",
     3. "author": "Robin Sharma",
     4. "genre": "Fiction",
     5. "read": false
     6. }
  2. Execute the code using postman or soapui
  3. 

1. Posting data – Save to Mongodb
   1. 
   2. Place the below code appropriately in app.js ( do not overwrite entire app.js)

bookRouter.route('/books')

.post(function(req,res){

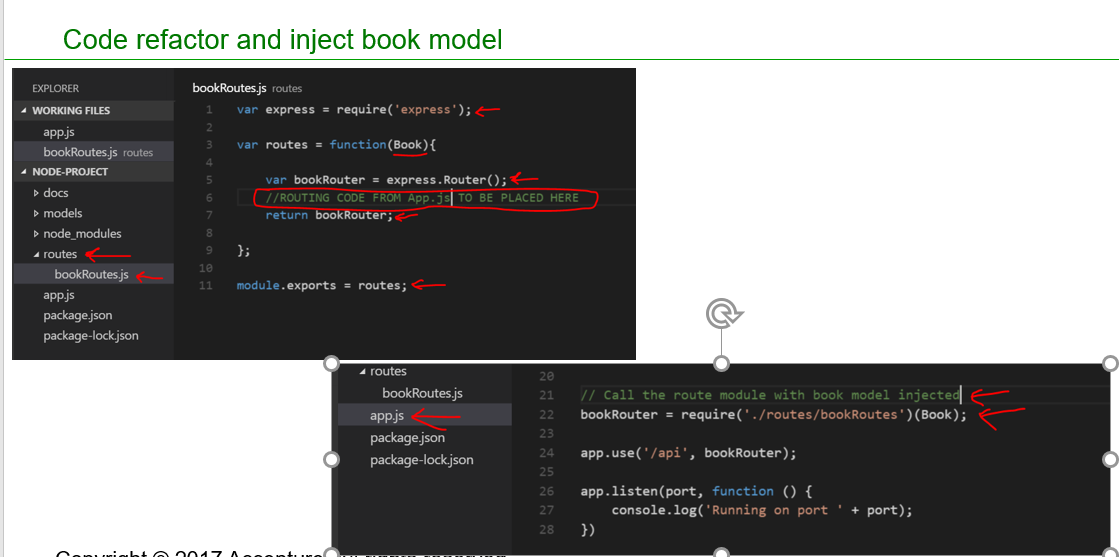
var book = new Book(req.body)

book.save()

res.status(201).send(book)

})

1. Posting data – Injecting model with code cleanup
   1. As our code in app.js is growing we need move some code(routing API code) to a separate file and inject books into it
   2. Create a file as shown below



* 1. Place the below code appropriately in bookRoutes.js

var express = require('express');

var routes = function(){

var bookRouter = express.Router();

bookRouter.route('/books')

.post(function(req,res){

var book = new Book(req.body)

book.save()

res.status(201).send(book)

})

.get(function(req,res){

var query = {}

if (req.query.genere)

{

query = req.query.genere

}

Book.find(query,function (err, books) {

if (err)

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

else

res.json(books);

});

})

bookRouter.route('/books/:Id')

.get(function(req,res){

Book.findById(req.params.Id,function (err, book) {

if (err)

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

else

res.json(book);

});

})

return bookRouter;

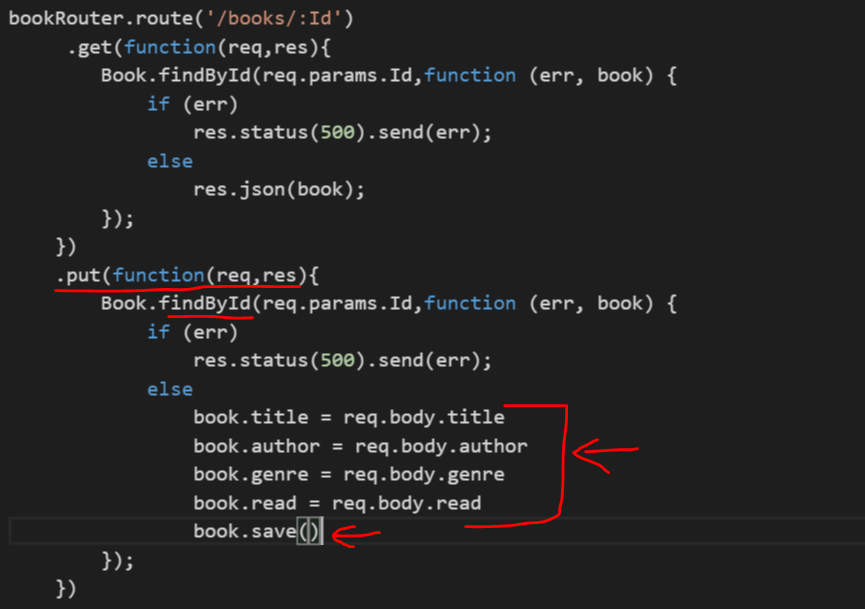
};

module.exports = routes;

1. Place the below code appropriately in app.js ( do not overwrite entire app.js)

// Call the route module with book model injected

bookRouter = require('./routes/bookRoutes')(Book);

1. PUT API
   1. 
   2. Place the below code appropriately in bookRoutes.js

bookRouter.route('/books/:Id')

.get(function(req,res){

Book.findById(req.params.Id,function (err, book) {

if (err)

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

else

res.json(book);

});

})

.put(function(req,res){

Book.findById(req.params.Id,function (err, book) {

if (err)

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

else

book.title = req.body.title

book.author = req.body.author

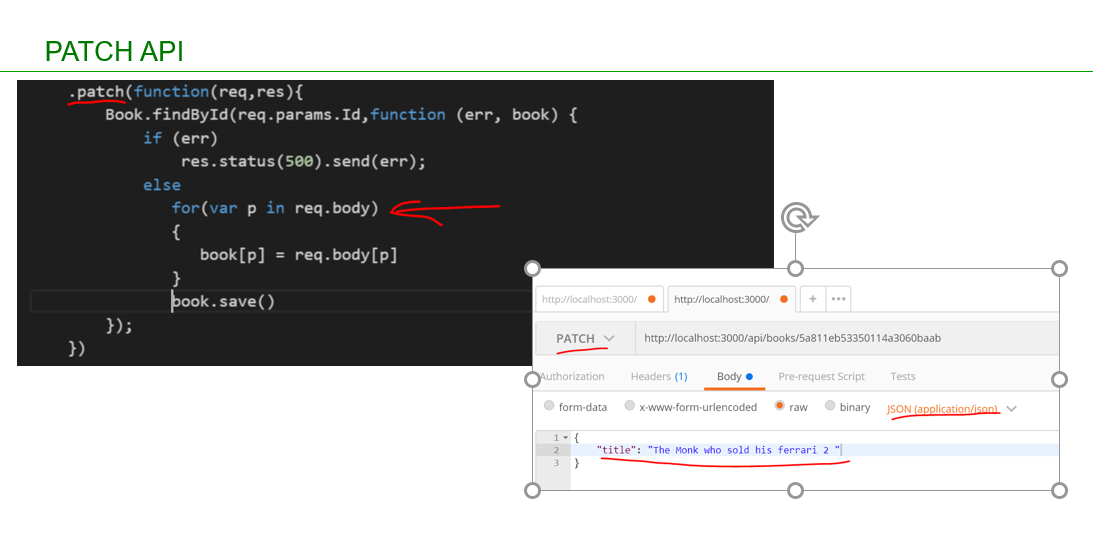
book.genre = req.body.genre

book.read = req.body.read

book.save()

});

})

1. PATCH API
   1. 
   2. Place the below code appropriately in bookRoutes.js

.patch(function(req,res){

Book.findById(req.params.Id,function (err, book) {

if (err)

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

else

for(var p in req.body)

{

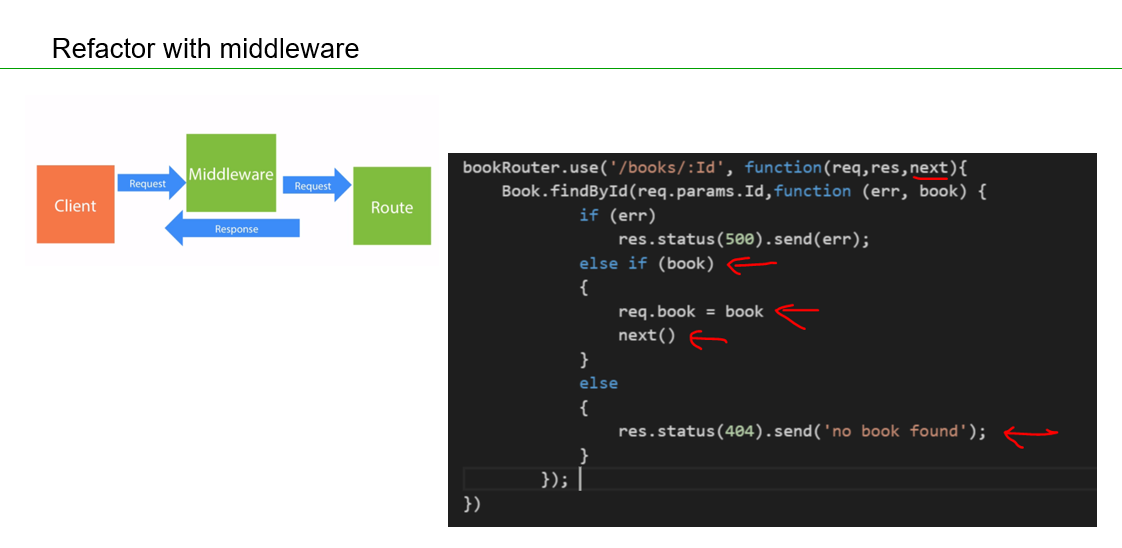
book[p] = req.body[p]

}

book.save()

});

})

1. Refactor with middleware
   1. 
   2. Place the below code appropriately in bookRoutes.js

// Middle Ware

bookRouter.use('/books/:Id', function(req,res,next){

Book.findById(req.params.Id,function (err, book) {

if (err)

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

else if (book)

{

req.book = book

next()

}

else

{

res.status(404).send('no book found');

}

});

})

bookRouter.route('/books/:Id')

.get(function(req,res){

res.json(req.book);

})

.put(function(req,res){

req.book.title = req.body.title

req.book.author = req.body.author

req.book.genre = req.body.genre

req.book.read = req.body.read

req.book.save()

res.json(req.book)

})

.patch(function(req,res){

if(req.body.\_id)

delete req.body.\_id

for(var p in req.body)

{

req.book[p] = req.body[p]

}

req.book.save()

res.json(req.book)

})

1. Remove API

