**Express.js**

0=global; 1… = steps;

app=response=request

1. Node Install✔1
2. NPM Install✔️2
3. NPM Init Projects✔️3
4. Express Installing✔️4
5. Nodemon Installing As Dev Dependencies✔️5
6. Project Structure Setup✔️6
7. Project Run Setup✔️7
8. Server Create✔️8
9. Route or Sub Route Setup✔️9
10. Sub Route Setup✔️90
11. Request & Response Handle✔️91
12. Request data accept ✔️9.2
13. Response data Sent ✔️93
14. Application Accept Data Type Setup(app.use())✔️10
15. Error Middleware Setup(app.use())✔️11
16. Not Found Middleware Setup(app.use())✔️12
17. Application View Engine Setup(app.set())✔️13
18. Application Usefull Method and Properties✔️0
19. Middleware Concept ✔️0.1

* 6
* controller
* views
* middleware
* router
* 9
* .all(“\*”)
* .get()
* .post()
* .put()
* .delete()
* .param()
* .route()
* 9.0
* express.Router()
* 9.2
* req.method
* req.body
* 9.3
* res.app = reference to the app instance
* res.headersSent = boolean if the app sent HTTP Headers for the response
* res.locals = local variables scoped to the response
* res.cookie = sets cookie name to value
* res.clearCookie = clears cookie by name
* res.end() = ends the response process
* res.send() = sends the HTTP response
* res.json() = sends json response
* res.status() = sets the HTTP status for the Response
* res.sendStatus() = sets the reponse HTTP status code
* res.render() = renders a view
* res.format() = performs content-negotiation on the Accept HTTP header on request object
* res.location() = sets the Response location HTTP Header
* res.redirect() = redirects to the URL derived from the specified path
* res.get() = returns the HTTP Response header
* res.set() = sets response HTTP header to a value
* 10
* app.use(express.json()) = json request accept
* app.use(express.raw()) = raw data accept but buffer/convert use toString()
* app.use(express.text()) = text data accept
* app.use(express.urlencoded()) = form data accept
* app.use(express.static(\_\_dirname+ “/public”)) = this folder file will be accessibel after domain
* app.use(express.router())
* 0
* app.locals.[any] = available entire application
* app.mountpath = accept sub app use route path
* app.param(‘id’,(req,res,next,[id])); = like middleware
* app.use(); = application usefull
* app.get() = get method
* app.post() = post method
* app.put() = put method
* app.all() = all method
* app.delete() = delete method
* app.listen() = server run
* app.route(). = multiple route request
* app.set(“view engine”, “ejs”) = view engine setup
* 0.1
* application/global level middleware
* router level middleware
* error handling middleware
* built-in middleware
* third-party middleware
* configurable middleware