Challenge Question: Express Generator Dependencies

Express Generator automatically installs several packages as dependencies: **cookie-parser**, **jade, mongoose**, etc.

**Part 1:**

Research these packages. Pick 3 of them (aside from Express) and look up their official online documentation pages. Explain briefly what each package is used for overall, then choose one specific part of the package's API and explain in your own words what it does, and how it is used. Add a code snippet on how to use it.

Pick a part of each API that has not already been described by another student. Include the link to the online documentation you used, along with links to any articles you read to supplement the online documentation. Note: There is one package installed by express-generator that is deprecated. If you choose to use this one for one of your answers, then include information about the deprecation and suggested updates.

**Part 2:**

What are some other packages that could be useful to use with Express? Research this question and provide your answer below with links to articles and sample code.

Answers:

**Part 1:**

* **Response-time**
  + Record HTTP response time.
  + <https://expressjs.com/en/resources/middleware/response-time.html>

var express = require('express')

var responseTime = require('response-time')

var app = express()

app.use(responseTime())

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

res.send('hello, world!')

})

* **Express-session**
  + Establish server-based sessions (development only).
  + <https://expressjs.com/en/resources/middleware/session.html>

var app = express()

var sess = {

secret: 'keyboard cat',

cookie: {}

}

if (app.get('env') === 'production') {

app.set('trust proxy', 1) // trust first proxy

sess.cookie.secure = true // serve secure cookies

}

app.use(session(sess))

* **Connect-timeout**
  + Set a timeout period for HTTP request processing.
  + <https://expressjs.com/en/resources/middleware/timeout.html>

var bodyParser = require('body-parser')

var cookieParser = require('cookie-parser')

var express = require('express')

var timeout = require('connect-timeout')

// example of using this top-level; note the use of haltOnTimedout

// after every middleware; it will stop the request flow on a timeout

var app = express()

app.use(timeout('5s'))

app.use(bodyParser())

app.use(haltOnTimedout)

app.use(cookieParser())

app.use(haltOnTimedout)

// Add your routes here, etc.

function haltOnTimedout (req, res, next) {

if (!req.timedout) next()

}

app.listen(3000)

**Part 2:**

* Errorhandler
  + This is for development-only. It looks at the entire path (fullstack) and looks at internal details of objects that pass through the module. If an error is found it will be sent back to the client.
  + <https://expressjs.com/en/resources/middleware/errorhandler.html>

var connect = require('connect')

var errorhandler = require('errorhandler')

var app = connect()

if (process.env.NODE\_ENV === 'development') {

// only use in development

app.use(errorhandler())

}