# Express.js Server Tutorial

* <https://expressjs.com/en/starter/hello-world.html>

## Purpose

* To demonstrate how to install express.js
* To set up a localhost server
* To initiate communication between the client and the server
* Install three.js and doxygen with the express server

## Installation

* Npm install express

## Usage

* Create a js script called app.js:
* const express = require('express')
* const app = express()
* const port = 3000
* app.get('/', (req, res) => {
* res.send("Hello World!")
* })
* app.listen(port, () => {
* console.log(`Example app listening on port ${port}`)
* })
* Run node app.js
* Migrate to [http://localhost/{port}](http://localhost/%7bport%7d)
  + Here {port} is 3000

# Express-Generator Tutorial

* <https://expressjs.com/en/starter/generator.html>
* Can use express-generator to quickly create an application skeleton

## Installation

* Npm install express-generator

## Usage

* Run: express –view=pug myapp
  + Creates an express application skeleton in a fresh directory called myapp
* Migrate to ./myapp
* Npm install
* (While using vscode terminal on windows) Access the server with the following command:
  + $env:DEBUG='myapp:\*'; npm start

## Hierarchy

* .
* ├── app.js
* ├── bin
* │ └── www
* ├── package.json
* ├── public
* │ ├── images
* │ ├── javascripts
* │ └── stylesheets
* │ └── style.css
* ├── routes
* │ ├── index.js
* │ └── users.js
* └── views
* ├── error.pug
* ├── index.pug
* └── layout.pug
* 7 directories, 9 files

## Serving Static Files

* <https://expressjs.com/en/starter/static-files.html>
* Add the following line to app.js
  + app.use(express.static('public'))
  + This makes anything in the ./public folder loadable in the url bar
  + Ex: <http://localhost/3000/test.html> (if test.html exists in ./public)
* Use multiple static assets directories by calling the above line as many times as necessary to all the locations.