# The Secrets of the Fullstack Ninja

## Part A - The Client Side

Author:
Oded Sagir

# Prerequisites

- ✓ node +npm
- ✓ git + github account
- ✓ Ruby + Sass + compass
- ✓ Ide / Text editor

- ✓ Core JavaScript + HTML + CSS

# Purpose

- ✓ Adopt modern web development work-flow and stack, in practice those used in LÉ U2.
- ✓ Set a fundamental understanding of BackBone & Marionette

## The Means

Build a SPA Blog App

# Web Application DevOp

| Version Control | Package Manager | Build Tool | Web Package Manager | Server |
|---|---|---|---|---|
| ↓ | ↓ | ↓ | ↓ | ↓ |
| Git | NPM | Grunt | Bower | Node |

# Create a github repository



git npm server grunt bower
create clone push

# Clone Locally

```
→ workspace  git clone git@github.com:hamecoded/myBlog.git
Cloning into 'myBlog'...
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (5/5), done.
Checking connectivity... done.
→ workspace  cd myBlog
→ myBlog git:(master)
```

▼ myBlog
  .gitignore
  LICENSE
  README.md

git npm server grunt bower
create clone push

# commit & push

```
.gitignore — myBlog                    ×

1   node_modules
2   bower_components
3   .sass-cache
4   .DS_Store
5   .idea
6
7   *.log
8
9   build/
10  dist/
11
```

*modify gitignore*

```
→ myBlog git:(master) git add .
→ myBlog git:(master) ✗ git commit -m "modify gitignore"
[master 7f3abd2] modify gitignore
 1 file changed, 11 insertions(+), 2 deletions(-)
→ myBlog git:(master) git push origin master
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 390 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:hamecoded/myBlog.git
   15228c0..7f3abd2  master -> master
→ myBlog git:(master)
```

git npm server grunt bower
create clone push

# Initialize NPM Project

- ▼ myBlog
  - .gitignore
  - LICENSE
  - **package.json**
  - README.md

git **npm** server grunt
**init**

```
→ myBlog git:(master) npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sane defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (myBlog)
version: (0.0.0)
description: A Blog App    ⇐
entry point: (index.js)
test command:
git repository: (https://github.com/hamecoded/myBlog.git)
keywords: blog spa
author: Oded Sagir
license: (ISC) MIT
About to write to /Users/odeds/workspace/myBlog/package.json:

{
  "name": "myBlog",
  "version": "0.0.0",
  "description": "A Blog App",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/hamecoded/myBlog.git"
  },
  "keywords": [
    "blog",
    "spa"
  ],
  "author": "Oded Sagir",
  "license": "MIT",
  "bugs": {
    "url": "https://github.com/hamecoded/myBlog/issues"
  },
  "homepage": "https://github.com/hamecoded/myBlog"
}

Is this ok? (yes) yes ✔
```

file content

# installing our first node module

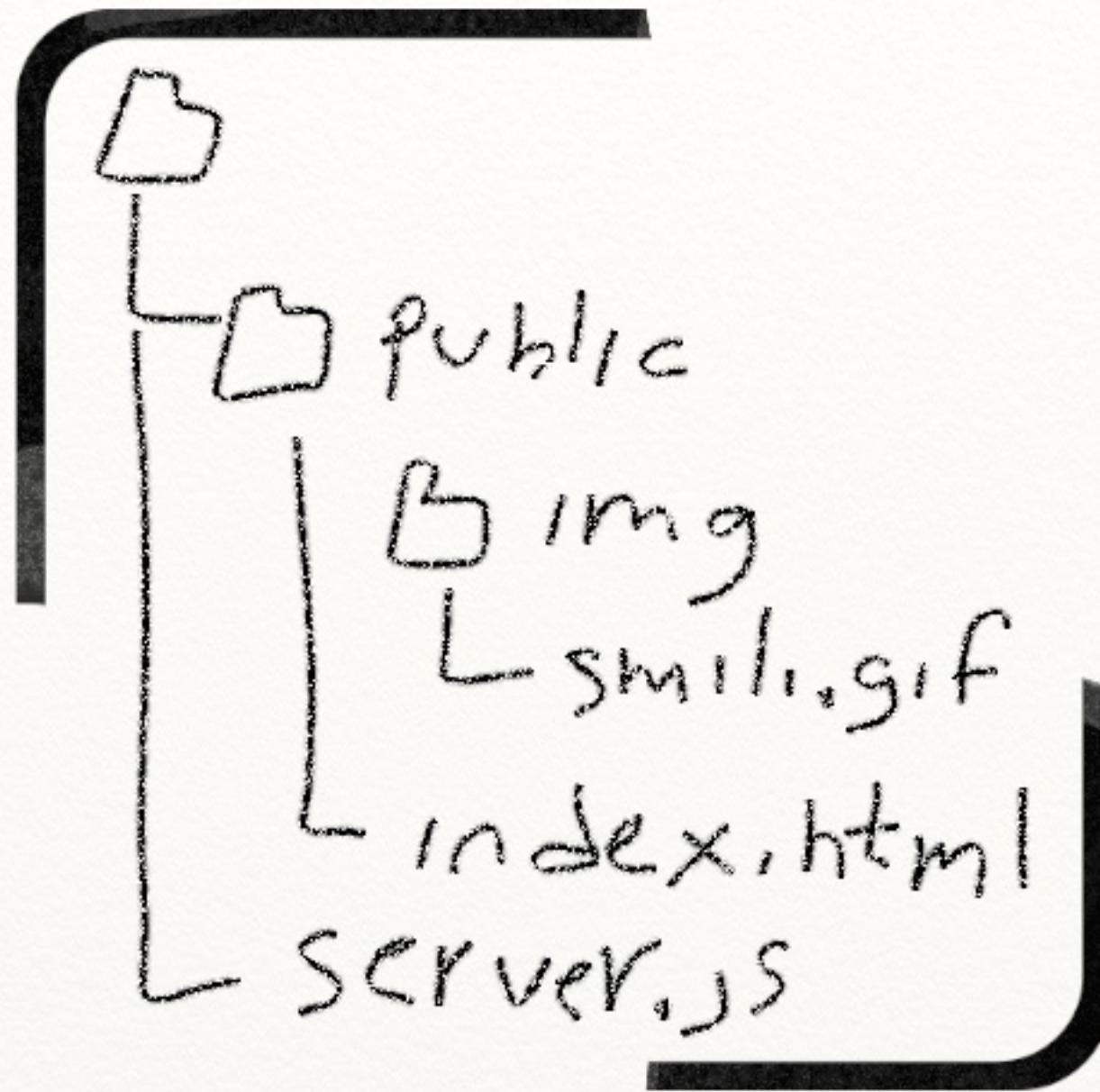→ npm install express --save

will add express
under node_modules

module
name

save to
package.json

git **npm** server grunt bower
init **install**

# Server fs Structure



```
📁
├── 📁 public
│   ├── 📁 img
│   │   └── smili.gif
│   └── index.html
└── server.js
```

git npm `server` grunt bower
`fs` SPA gun

# Not just a server but a SPA server

```javascript
/**
 * An express server to serve a SPA app
 * we use express to simplify handling of static files
 * whilst directing all paths to main index.html
 * for handling browser refresh
 * a more complicate solution using standard modules can be found here:
 * https://coderwall.com/p/b2kmkg
 * an interesting discussion:
 * http://stackoverflow.com/questions/7268033/basic-static-file-server-in-nodejs
 */
var express = require('express'),
        app = express();

app.use(express.static(__dirname + '/public'));

app.get('*', function (req, res) {
    res.sendfile(__dirname + '/public/index.html');
});

var server = app.listen(process.env.PORT || 8088, function() {
    console.log('Listening on port %d', server.address().port);
});
```
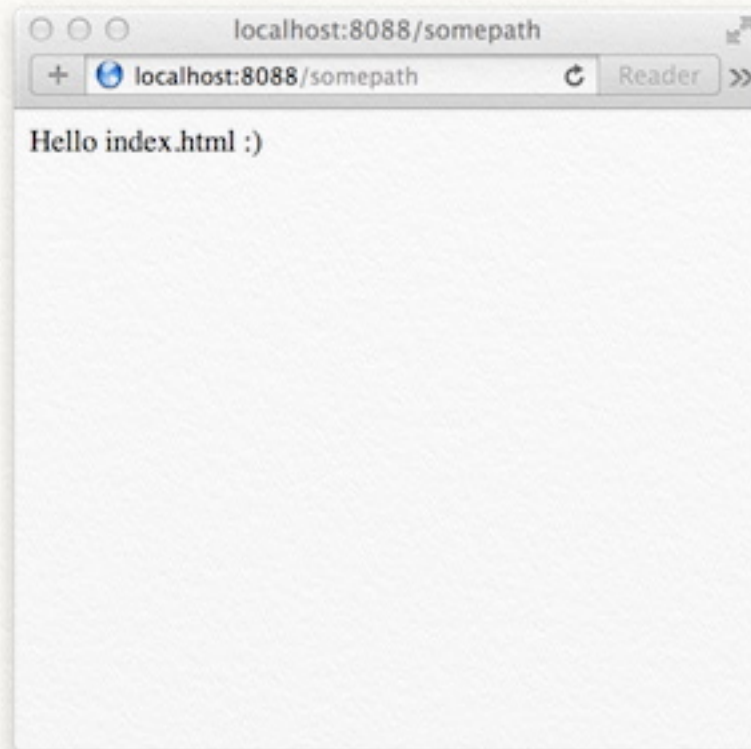
will handle assets

a route not found will respond with index.html

git npm server grunt bower
fs SPA run

# Test & Run your Server

```
→  myBlog git:(master) ✗ node server.js
Listening on port 8088
```

**Window 1 — localhost:8088/somepath**

localhost:8088/somepath    Reader  »

Hello index.html :)

**Window 2 — smili.gif 245×165 pixels**

localhost:8088/img/smili.gif    Reader  »

git  npm  server  grunt  bower
     fs  SPA  run

It's Training Time

# Web Application Infra Soup

Dependency | Core | MV* | UI | Templating | Testings

Require

jQuery
Underscore

BackBone
Marionette

Twitter
Bootstrap

Mustache

qUnit