

Bootstrap D3 Web Demos

Stewart Noyce

`d3.oakland()`

March 10, 2015

Goal

Quickly build and deploy
a website that
privately shares a
D3 visualization demo.

Requirements

Feature	Requirements
Framework	Use an existing, stable, consistent, familiar framework
Database	Use a flexible database with an application-defined schema
Workflow	Allow customization, but minimize necessary changes
Registration	Admin registers user and assigns to demo account.
Web Pages	Present HTML5 and JavaScript web pages from templates
Free	No cost to build and deploy the demo

Technology Stack



D3.js



HTML5



Node.js



MongoDB

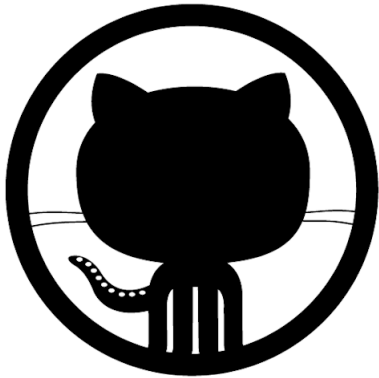
Extensible Framework



Drywall – a website and user system for Node.js

On the Server	On the Client	Development
Express	Bootstrap	Grunt
Jade	Backbone.js	
Mongoose	jQuery	
Passport	Underscore.js	
Async	Font-Awesome	
EmailJs	Moment.js	

Build and Deploy

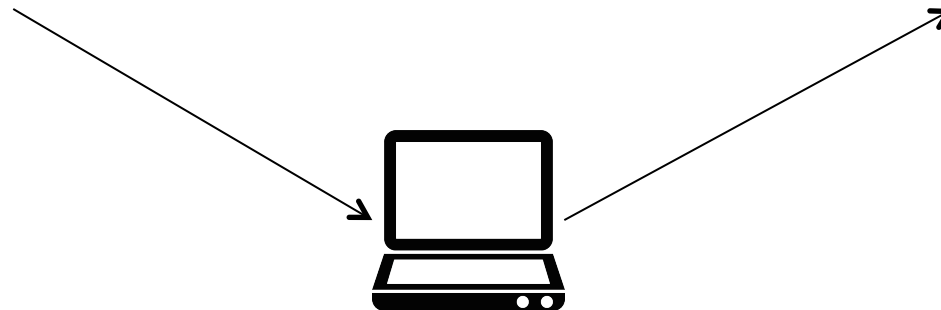


<https://github.com/jedireza/drywall>

<https://github.com/stewartnoyce/mend-d3>



<https://demo.herokuapp.com>



~/Demospace

Agenda

- Get ready
- Build demo repository
- Test drive the system
- Extend your D3 demo

Get Ready

- Load Node
 - <https://nodejs.org/download/>
- Latest NPM Version
 - “npm install npm -g”
- Grunt
 - “npm install grunt cli -g”
- Heroku
 - Create Heroku account
 - Install Heroku toolbelt (<https://toolbelt.heroku.com/>)

Build Demo Repository

```
$ git clone https://github.com/jedireza/drywall.git demo  
$ git clone https://github.com/StewartNoyce/mend-d3.git mend-d3  
$ cd mend-d3  
$ source ./install.sh
```


MEND-D3 Demo Files

File	Purpose
routes.js	Add demo routes to get page and lookupData
models.js	Add Demo schema to the set of models
Demo.js	Add a schema to read (and write) the demodata collection
layouts/	Modify default and account jade files for demo workflow
views/	Add account/demo .js and .jade files for demo page
views/index.jade	Change home page for admin user registration
public/views	Add account/demo .js and .css files for d3 test / starter page
demodata.csv	Import this data to test the installation


Deploy Demo Repository

```
$ cd ../demo
$ rm .gitignore
$ heroku login
$ heroku create
$ heroku addons:add mongolab
$ npm install
$ npm install grunt-cli -g
$ mv config.example.js config.js
$ vi config.js
$ grunt build
$ git add . -A
$ git commit -am "init"
$ git push heroku master
$ heroku open
```

Switch to admin for global
install as necessary



Change systemEmail,
from.address, and smtp
server credentials



But We're Not Done

- MONGODB_URI
 - 'heroku config'
 - mongodb://<dbuser>:<dbpassword>@<host:port>/<app>
- Client Editing
 - 'mongo <host>:<app> -u <dbuser> -p <dbpassword>'
- Import Data
 - 'mongoimport -h <host> -d <app> -c demodata -u <dbuser> -p <dbpassword>' --file demodata.csv --type csv --headerline

Set Up Root User as Admin

- `db.admingroups.insert({ _id: 'root', name: 'Root' })`
- `db.admins.insert({ name: {first: 'Root', last: 'Admin', full: 'Root Admin'}, groups: ['root'] })`
- `var rootAdmin = db.admins.findOne()`
- `db.users.save({ username: 'root', isActive: 'yes', email: '<email>', roles: {admin: rootAdmin._id} })`
- `var rootUser = db.users.findOne()`
- `rootAdmin.user = { id: rootUser._id, name: rootUser.username }`
- `db.admins.save(rootAdmin)`

Extend the Demo

- Rename Your App
 - “heroku apps:rename <newname>”
- Change the Logos
- Change the Text
- Change the Schema
- Make a D3 Visualization

TruNorth