



Hlutur

JavaScript & IOT

Leif Terje Fonnes

Nodejs.org

Google's V8 VM with JIT compilation

Async I/O

Lots and lots of libs

Supports quite a few frameworks. We will use
express

Node JS

Single threaded

Non blocking

Background worker threads for I/O processing

Uses callbacks to notify completion

Install node on the pi

- <http://node-arm.herokuapp.com/>

```
wget http://node-arm.herokuapp.com/node_latest_armhf.deb
```

```
sudo dpkg -i node_latest_armhf.deb
```

```
# Check installation
```

```
node -v
```

other platforms

Check nodejs.org

I recommend to have git installed also

Other stuff

express - <http://expressjs.com/> simple web
server framework



Lets make our first app

```
mkdir app1
```

```
cd app1
```

```
npm init
```

```
npm install express --save
```


app1.js

```
var express=require('express');
var app=express();

app.get('/',function(req,res){
    res.send('Hello OSDC 2015');
});

var server=app.listen(3000,function() {
    var host = server.address().address;
    var port = server.address().port;
    console.log("Ready at http://%s:%s",host,port);
});
```

Test

Open a browser and navigate to `http://localhost:3000`

Git

git init .

git add .

git commit -m "Initial version"

NOTE 1

To avoid to big use **.gitignore** files with the following content

node_modules

NOTE 2

git config --global user.name "Your name"

git config --global user.email "example@email.com"

Serving static files

Lets add before “var server”

```
app.use(express.static('public'));
```

Test by adding a image to public

```
# mkdir public  
# cd public/  
# wget https://www.raspberrypi.org/wp-content/uploads/2014/07/rsz_b-.jpg  
...  
# mv rsz_b-.jpg img1.jpg
```

http://ip_adress:3000/img1

index.html

```

```



Adding a generator

```
$ npm install express-generator -g
```


Application 2

```
$ express app2 -e
```

```
create : app2
```

```
.....
```

```
create : app2/bin
```

```
create : app2/bin/www
```

```
$ cd app2 && npm install
```

```
$ DEBUG=app2 npm start
```

Framework setup

```
.
├─ app.js
├─ bin
│   └─ www
├─ package.json
├─ public
│   ├── images
│   ├── javascripts
│   └─ stylesheets
│       └─ style.css
├─ routes
│   ├── index.js
│   └─ users.js
└─ views
    ├── error.jade
    ├── index.jade
    └─ layout.jade
```

7 directories, 9 files

Test it out

How can we add data to the template ?

Where do we modify the template ?

nodemon

<http://nodemon.io/>

Automatic reload of server

```
$ npm install -g nodemon
```

Update package.json

```
"start": "nodemon ./bin/www"
```

Try to make a few changes

Try to make changes to index.js and refresh in browser.

browsersync / LiveReload

<http://www.browsersync.io/>

Tool for automatically reload browser changes

```
$ npm install browser-sync --save-dev
```

Modify bin/www

add in to section:

```
var browserSync = require('browser-sync');
```

and inside the onListening function :

```
browserSync({  
  proxy: 'localhost:' + addr.port,  
  files: ['public/**/*.{js,css}']  
});
```

Try it out

Do some stylesheet changes

Add some text to the templates



MQTT

A practical protocol for the internet of things