

# Devtools: Nodemon

# The Problem

- Right now, if we want to test changes, we have to re-run our scripts
- This isn't that bad for simple terminal scripts, but it's a pain for Express
- What would be nice is if every time we changed our script, it reloaded

# Nodemon

- Nodemon does exactly that, it watches your files and if you make changes, reruns your script
- Instead of running our scripts with node, we'll run them with nodemon to enable this
- We can get nodemon by doing what we normally do for modules, `npm install` it

# Using Nodemon

- Even though we've installed nodemon, we can't just type nodemon in to our terminal to run it because our terminal isn't aware of the node\_modules folder
- That's where npm comes to help, our package.json has a scripts object where we can add handy shortcuts
- The scripts we put in here *are* aware of node\_modules, so this can use nodemon

# Setting up package.json

- Inside the "scripts" key, we'll add a new one called "start" that nodemon's our app.js
- There's a "test" script in there by default from npm init, don't worry about it

```
{  
  "scripts": {  
    "start": "nodemon app.js",  
    "test": "echo \"Error: no test specified\" && exit 1"  
  }  
}
```

# Running package.json Scripts

- Running those scripts is as simple as `npm run [script]`, where the name is the key
- `start` and `test` are special, in that you can also run them by doing `npm start` and `npm run` respectively. Any other script requires `npm run` before it.
- Try running `start`, you should get the following output:

# Nodemon Output

```
[nodemon] 1.11.0
```

```
[nodemon] to restart at any time, enter `rs`
```

```
[nodemon] watching: *.*
```

```
[nodemon] starting `node app.js`
```

```
(Your custom listen console.log)
```

- And if you make a change, you should see:

```
[nodemon] restarting due to changes...
```

```
[nodemon] starting `node app.js`
```

```
(Your custom listen console.log again)
```