

Web Development with Node.js and MongoDB

Andrew Marcinkevičius

Video 3.4

Scaffolding an App



In this Video, we are going to take a look at...

- Express generator tool
- Middleware
- User authentication

```
$ npm install -g express-generator@4
```

```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a on master [!]  
$ npm install -g express-generator@4  
/Users/ifdattic/.npm-global/bin/express -> /Users/ifdattic/.npm-global/lib/node_modules/express-generator/bin/express  
/Users/ifdattic/.npm-global/lib  
└─┬ express-generator@4.13.1  
  └─┬ commander@2.7.1  
    └─┬ graceful-readlink@1.0.1  
      └─┬ mkdirp@0.5.1  
        └─┬ minimist@0.0.8  
          └─┬ sorted-object@1.0.0
```

```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a on master [!]  
$ █
```

```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a on master [!]
```

```
$ express -h
```

```
Usage: express [options] [dir]
```

```
Options:
```

-h, --help	output usage information
-V, --version	output the version number
-e, --ejs	add ejs engine support (defaults to jade)
--hbs	add handlebars engine support
-H, --hogan	add hogan.js engine support
-c, --css <engine>	add stylesheet <engine> support (less stylus compass sass) (defaults to plain css)
--git	add .gitignore
-f, --force	force on non-empty directory

```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a on master [!]
```

```
$ █
```



```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a on master [!]
```

```
$ express workout --hbs --git
```

```
create : workout
create : workout/package.json
create : workout/app.js
create : workout/.gitignore
create : workout/public
create : workout/public/javascripts
create : workout/routes
create : workout/routes/index.js
create : workout/routes/users.js
create : workout/public/images
create : workout/public/stylesheets
create : workout/public/stylesheets/style.css
create : workout/views
create : workout/views/index.hbs
create : workout/views/layout.hbs
create : workout/views/error.hbs
create : workout/bin
create : workout/bin/www
```

```
install dependencies:
```

```
$ cd workout && npm install
```

```
run the app:
```

```
$ DEBUG=workout:* npm start
```

```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a on master [!]
```

```
$ █
```

```
$ cd workout
```

```
$ npm install
```

[illegible]

```
| |-- range-parser@1.0.3
| | |-- send@0.13.0
| | | |-- destroy@1.0.3
| | | | |-- mime@1.3.4
| | |-- serve-static@1.10.1
| | | |-- escape-html@1.0.3
| | | | |-- send@0.13.1
| | | | | |-- depd@1.1.0
| | | | | | |-- destroy@1.0.4
| |-- utils-merge@1.0.0
| | |-- vary@1.0.1
|-- hbs@3.1.1
| |-- handlebars@3.0.0
| | |-- optimist@0.6.1
| | | |-- minimist@0.0.10
| | | | |-- wordwrap@0.0.3
| | |-- source-map@0.1.43
| | | |-- amdefine@1.0.0
| | | |-- uglify-js@2.3.6
| | | | |-- async@0.2.10
| | | | | |-- optimist@0.3.7
| | |-- walk@2.2.1
| | | |-- forEachAsync@2.2.1
| | | | |-- sequence@2.2.1
|-- morgan@1.6.1
| |-- basic-auth@1.0.3
| | |-- on-headers@1.0.1
|-- serve-favicon@2.3.0
```

ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?!]

\$ █


```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?] |
```

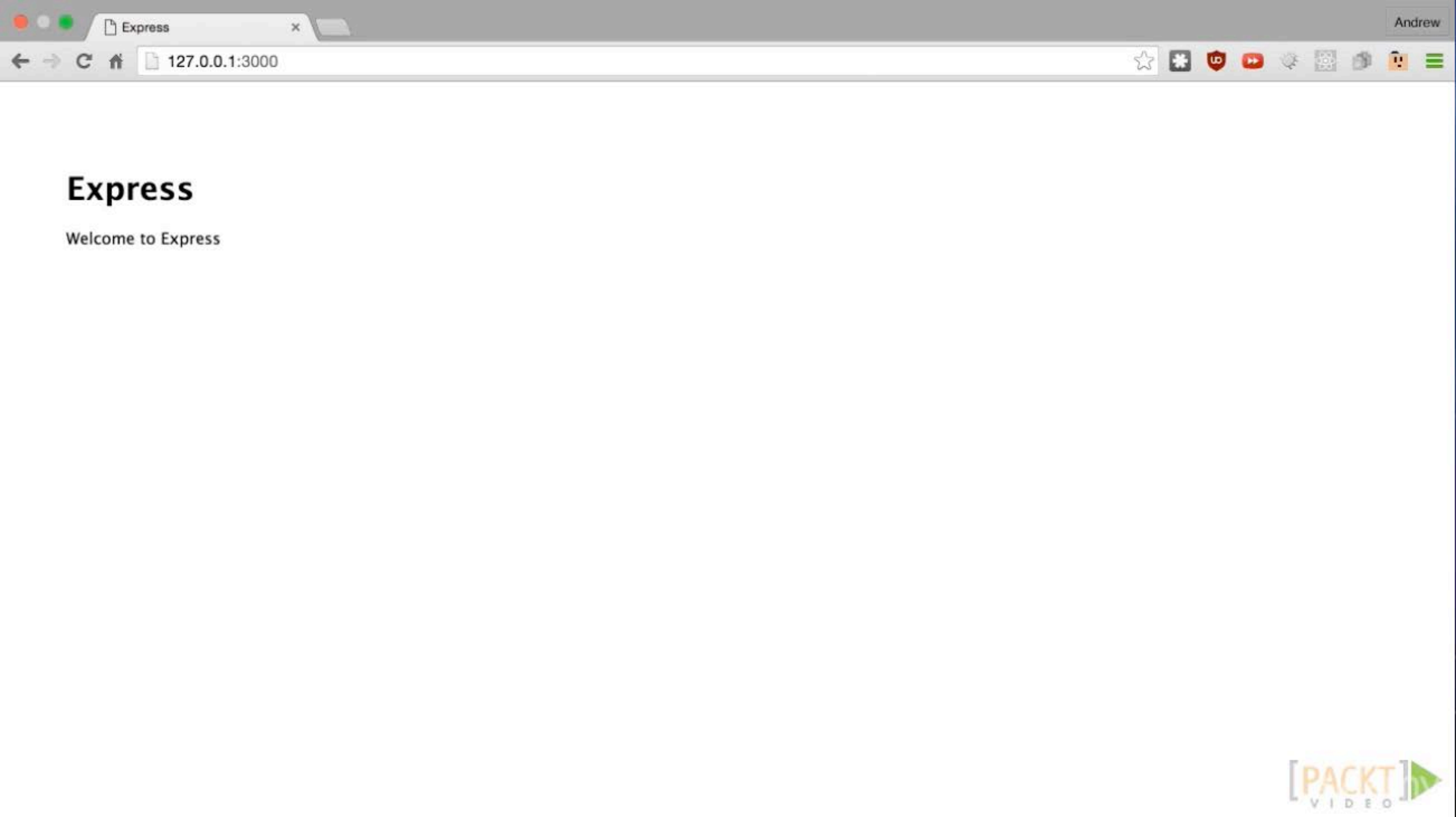
```
$ DEBUG=workout:* npm start
```

```
> workout@0.0.0 start /Users/ifdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout
```

```
> node ./bin/www
```

```
workout:server Listening on port 3000 +0ms
```

```
█
```



Express

Welcome to Express

FOLDERS

- ▼ code
 - ▶ 1
 - ▶ 2
 - ▼ 3
 - ▶ 3.1
 - ▶ 3.2
 - ▶ 3.3
 - ▼ 3.4
 - ▼ a
 - ▼ workout
 - ▶ bin
 - ▶ node_modules
 - ▶ public
 - ▶ routes
 - ▶ views
 - .gitignore
 - app.js
 - package.json
 - start.js
 - ▶ workout
 - ▶ workout-modified
 - ▶ 3.5
 - ▶ 4
 - ▶ 5
 - ▶ 6
 - ▶ 7
 - ▶ 8
 - export-environment-variables.sh
- ▶ scripts

```
start.js x
1 require('./bin/www');
2
```

FOLDERS

- ▼ code
 - ▶ 1
 - ▶ 2
 - ▼ 3
 - ▶ 3.1
 - ▶ 3.2
 - ▶ 3.3
 - ▼ 3.4
 - ▼ a
 - ▼ workout
 - ▶ bin
 - ▶ node_modules
 - ▶ public
 - ▶ routes
 - ▶ views
 - 📄 .gitignore
 - 📄 app.js
 - 📄 package.json
 - 📄 start.js
 - ▶ workout
 - ▶ workout-modified
 - ▶ 3.5
 - ▶ 4
 - ▶ 5
 - ▶ 6
 - ▶ 7
 - ▶ 8
 - 📄 export-environment-variables.sh
 - ▶ scripts

```
package.json
1  {
2    "name": "workout",
3    "version": "0.0.0",
4    "private": true,
5    "scripts": {
6      "start": "nodemon start.js"
7    },
8    "dependencies": {
9      "body-parser": "~1.13.2",
10     "cookie-parser": "~1.3.5",
11     "debug": "~2.2.0",
12     "express": "~4.13.1",
13     "hbs": "~3.1.0",
14     "morgan": "~1.6.1",
15     "serve-favicon": "~2.3.0"
16   }
17 }
```



```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?] |
```

```
$ DEBUG=workout:* npm start
```

```
> workout@0.0.0 start /Users/ifdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout
```

```
> nodemon start.js
```

```
[nodemon] 1.8.1
```

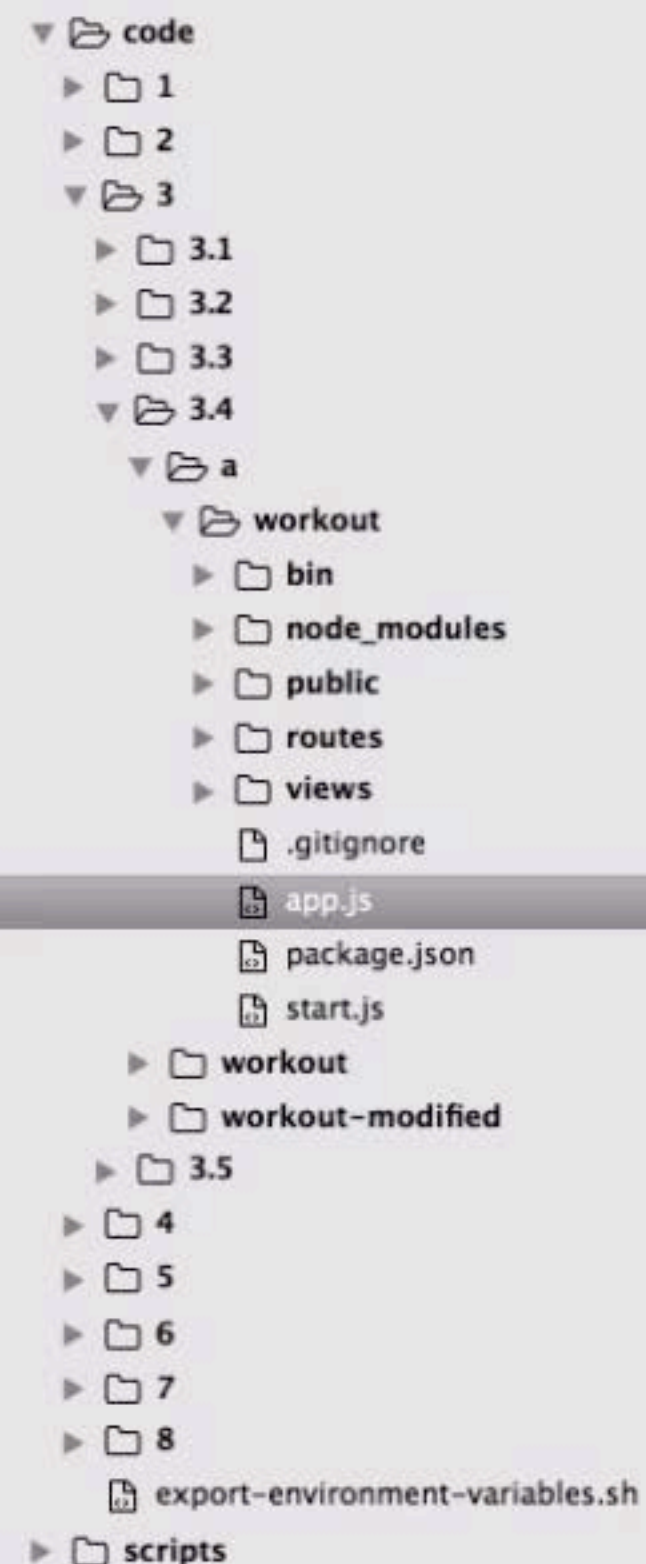
```
[nodemon] to restart at any time, enter `rs`
```

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

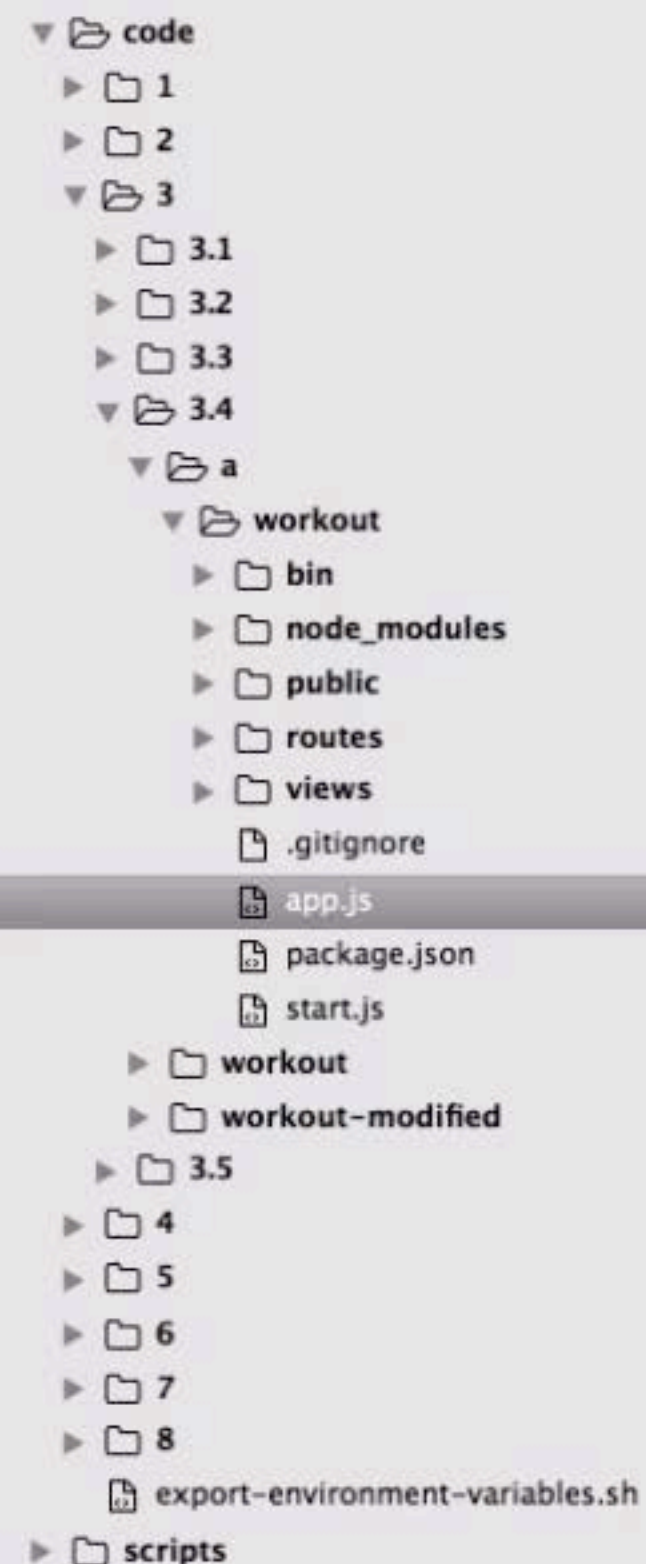
```
[nodemon] starting `node start.js`
```

```
workout:server Listening on port 3000 +0ms
```

```
█
```



```
app.js
1 var express = require('express');
2 var path = require('path');
3 var favicon = require('serve-favicon');
4 var logger = require('morgan');
5 var cookieParser = require('cookie-parser');
6 var bodyParser = require('body-parser');
7 console.log('')
8 var routes = require('./routes/index');
9 var users = require('./routes/users');
10
11 var app = express();
12
13 // view engine setup
14 app.set('views', path.join(__dirname, 'views'));
15 app.set('view engine', 'hbs');
16
17 // uncomment after placing your favicon in /public
18 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
19 app.use(logger('dev'));
20 app.use(bodyParser.json());
21 app.use(bodyParser.urlencoded({ extended: false }));
22 app.use(cookieParser());
23 app.use(express.static(path.join(__dirname, 'public')));
24
25 app.use('/', routes);
26 app.use('/users', users);
27
28 // catch 404 and forward to error handler
29 app.use(function(req, res, next) {
30   var err = new Error('Not Found');
```

```
app.js
1 var express = require('express');
2 var path = require('path');
3 var favicon = require('serve-favicon');
4 var logger = require('morgan');
5 var cookieParser = require('cookie-parser');
6 var bodyParser = require('body-parser');
7 console.log('Hi');
8 var routes = require('./routes/index');
9 var users = require('./routes/users');
10
11 var app = express();
12
13 // view engine setup
14 app.set('views', path.join(__dirname, 'views'));
15 app.set('view engine', 'hbs');
16
17 // uncomment after placing your favicon in /public
18 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
19 app.use(logger('dev'));
20 app.use(bodyParser.json());
21 app.use(bodyParser.urlencoded({ extended: false }));
22 app.use(cookieParser());
23 app.use(express.static(path.join(__dirname, 'public')));
24
25 app.use('/', routes);
26 app.use('/users', users);
27
28 // catch 404 and forward to error handler
29 app.use(function(req, res, next) {
30   var err = new Error('Not Found');
```

```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?] |
```

```
$ DEBUG=workout:* npm start
```

```
> workout@0.0.0 start /Users/ifdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout
```

```
> nodemon start.js
```

```
[nodemon] 1.8.1
```

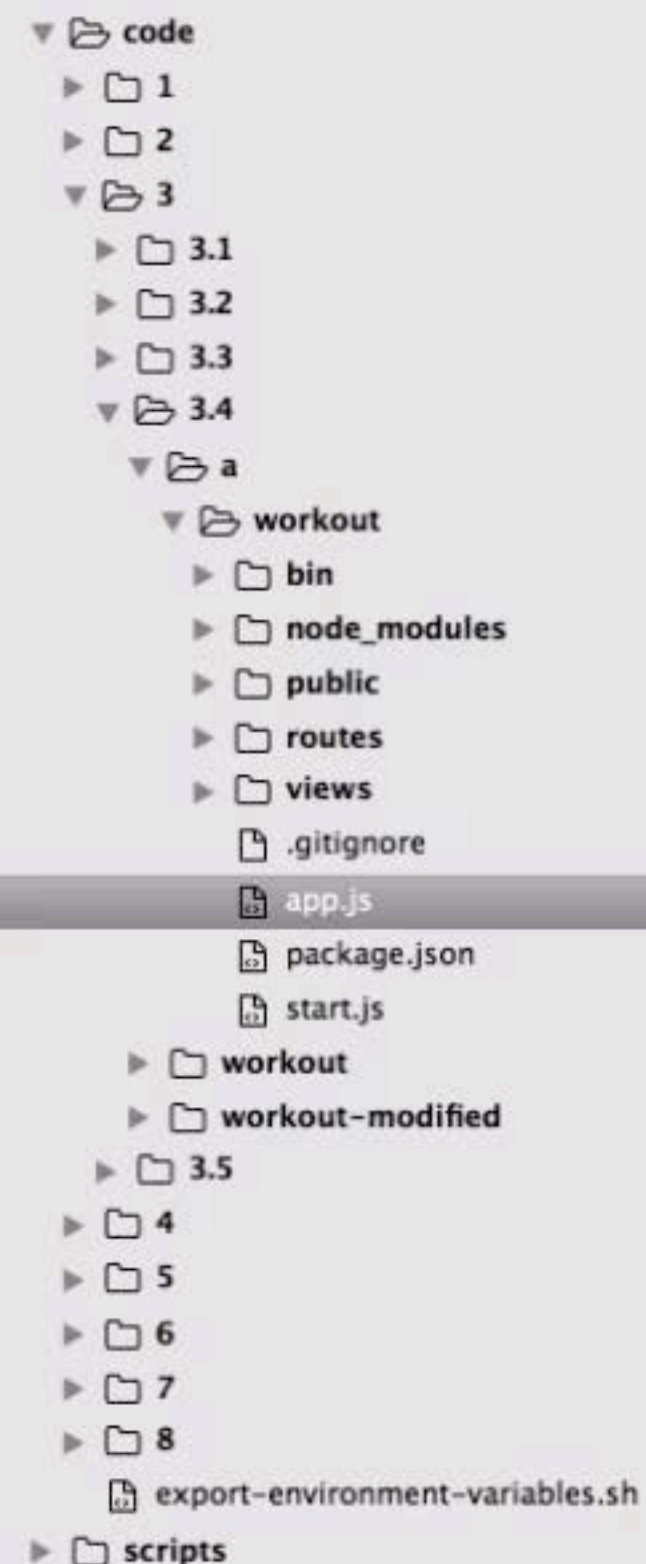
```
[nodemon] to restart at any time, enter `rs`
```

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

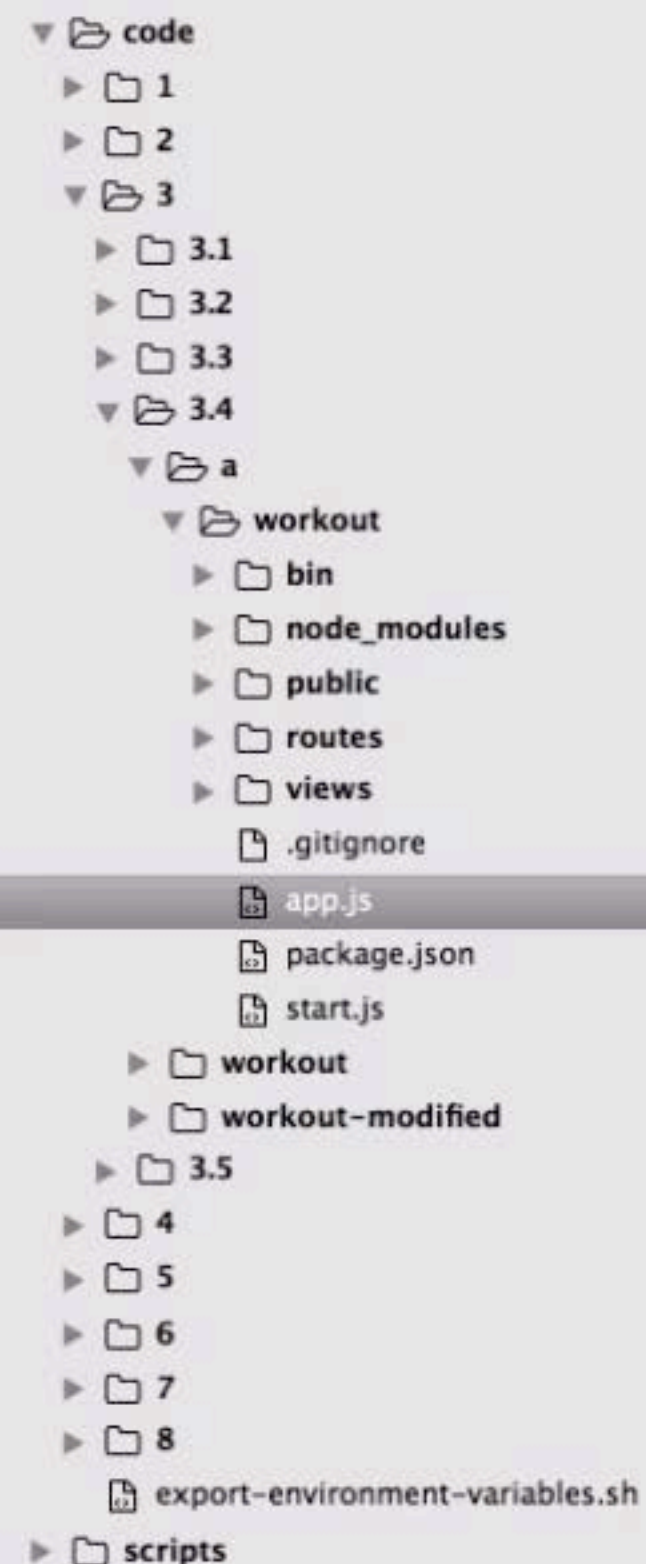
```
[nodemon] starting `node start.js`
```

```
workout:server Listening on port 3000 +0ms
```

```
█
```

```
app.js
1 var express = require('express');
2 var path = require('path');
3 var favicon = require('serve-favicon');
4 var logger = require('morgan');
5 var cookieParser = require('cookie-parser');
6 var bodyParser = require('body-parser');
7 console.log('')
8 var routes = require('./routes/index');
9 var users = require('./routes/users');
10
11 var app = express();
12
13 // view engine setup
14 app.set('views', path.join(__dirname, 'views'));
15 app.set('view engine', 'hbs');
16
17 // uncomment after placing your favicon in /public
18 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
19 app.use(logger('dev'));
20 app.use(bodyParser.json());
21 app.use(bodyParser.urlencoded({ extended: false }));
22 app.use(cookieParser());
23 app.use(express.static(path.join(__dirname, 'public')));
24
25 app.use('/', routes);
26 app.use('/users', users);
27
28 // catch 404 and forward to error handler
29 app.use(function(req, res, next) {
30   var err = new Error('Not Found');
```

```
app.js
1 var express = require('express');
2 var path = require('path');
3 var favicon = require('serve-favicon');
4 var logger = require('morgan');
5 var cookieParser = require('cookie-parser');
6 var bodyParser = require('body-parser');
7 console.log('Hi');
8 var routes = require('./routes/index');
9 var users = require('./routes/users');
10
11 var app = express();
12
13 // view engine setup
14 app.set('views', path.join(__dirname, 'views'));
15 app.set('view engine', 'hbs');
16
17 // uncomment after placing your favicon in /public
18 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
19 app.use(logger('dev'));
20 app.use(bodyParser.json());
21 app.use(bodyParser.urlencoded({ extended: false }));
22 app.use(cookieParser());
23 app.use(express.static(path.join(__dirname, 'public')));
24
25 app.use('/', routes);
26 app.use('/users', users);
27
28 // catch 404 and forward to error handler
29 app.use(function(req, res, next) {
30   var err = new Error('Not Found');
```

```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?]
$ DEBUG=workout:* npm start
```

```
> workout@0.0.0 start /Users/ifdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout
```

```
> nodemon start.js
```

```
[nodemon] 1.8.1
```

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

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

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

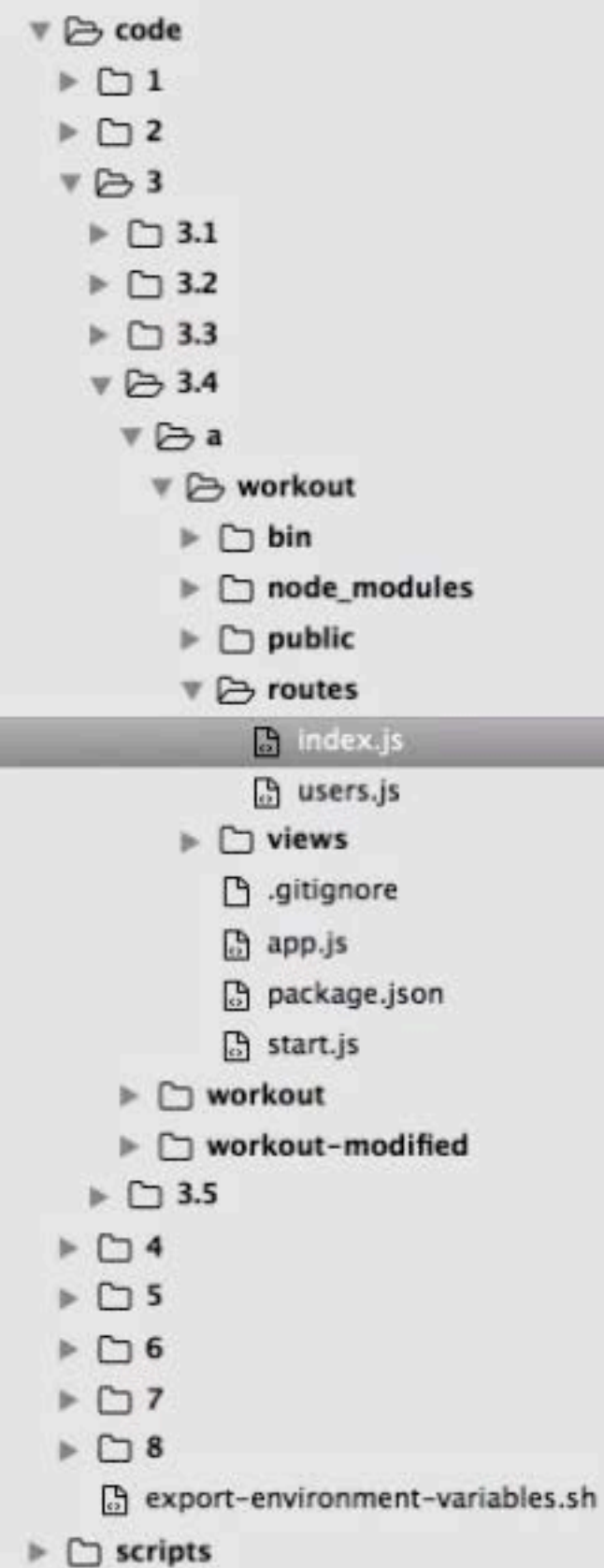
```
workout:server Listening on port 3000 +0ms
```

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

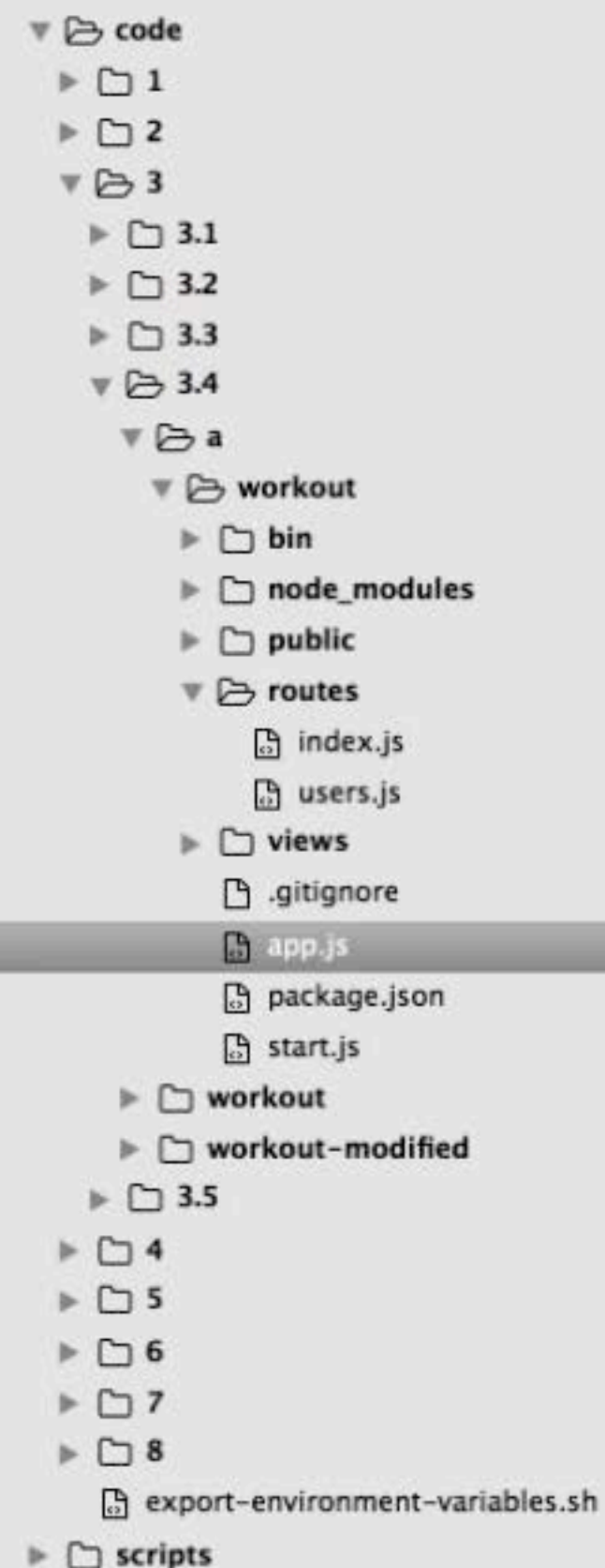
```
[nodemon] starting `node start.js`
```

```
Hi
```

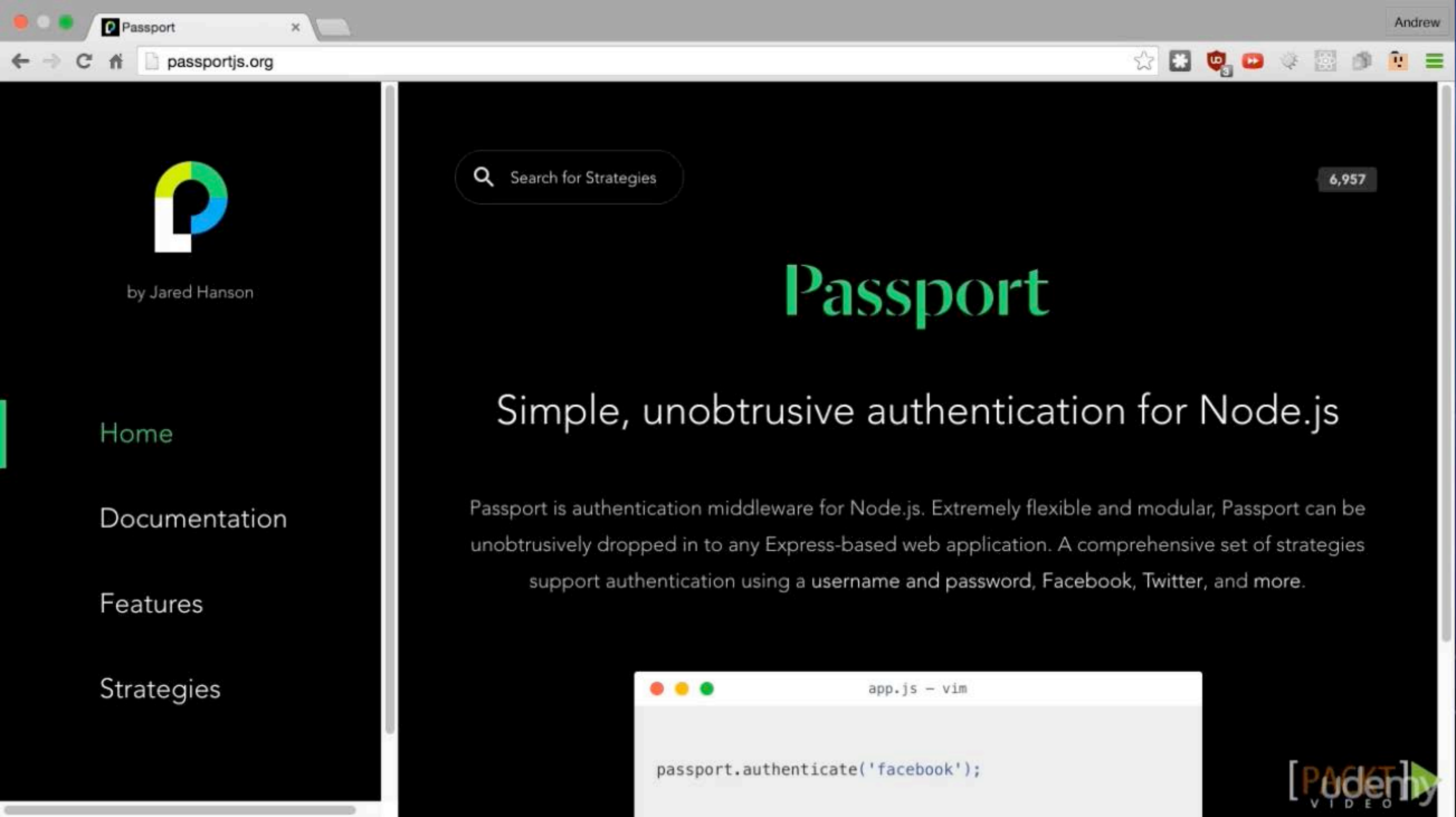
```
workout:server Listening on port 3000 +0ms
```

```
index.js
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function(req, res, next) {
6   res.render('index', { title: 'Express' });
7 });
8
9 module.exports = router;
10
```

```
app.js
1 var express = require('express');
2 var path = require('path');
3 var favicon = require('serve-favicon');
4 var logger = require('morgan');
5 var cookieParser = require('cookie-parser');
6 var bodyParser = require('body-parser');
7
8 var routes = require('./routes/index');
9 var users = require('./routes/users');
10
11 var app = express();
12
13 // view engine setup
14 app.set('views', path.join(__dirname, 'views'));
15 app.set('view engine', 'hbs');
16
17 // uncomment after placing your favicon in /public
18 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
19 app.use(logger('dev'));
20 app.use(bodyParser.json());
21 app.use(bodyParser.urlencoded({ extended: false }));
22 app.use(cookieParser());
23 app.use(express.static(path.join(__dirname, 'public')));
24
25 app.use('/', routes);
26 app.use('/users', users);
27
28 // catch 404 and forward to error handler
29 app.use(function(req, res, next) {
30   var err = new Error('Not Found');
```



Home

Documentation

Features

Strategies

Search for Strategies

6,957

Passport

Simple, unobtrusive authentication for Node.js

Passport is authentication middleware for Node.js. Extremely flexible and modular, Passport can be unobtrusively dropped in to any Express-based web application. A comprehensive set of strategies support authentication using a username and password, Facebook, Twitter, and more.

```
app.js - vim

passport.authenticate('facebook');
```




```
app.js
37
38 // catch 404 and forward to error handler
39 app.use(function(request, response, next) {
40     const error = new Error('Not Found');
41     error.status = 404;
42     next(error);
43 });
44
45 // error handlers
46
47 // development error handler
48 // will print stacktrace
49 if (app.get('env') === 'development') {
50     app.use(function(error, request, response, next) {
51         response.status(error.status || 500);
52         response.render('error', {
53             message: error.message,
54             error: error
55         });
56     });
57 }
58
59 // production error handler
60 // no stacktraces leaked to user
61 app.use(function(error, request, response, next) {
62     response.status(error.status || 500);
63     response.render('error', {
64         message: error.message,
65         error: {}
66     });
67 });
```

ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?]

\$ npm install --save passport passport-local

|||||



```
passport.js
1 'use strict';
2
3 // const passport = require('passport');
4 // const Strategy = require('passport-local').Strategy;
5 // const db = require('../db');
6
7 // passport.use(new Strategy(
8 //     function (username, password, done) {
9 //         db.users.findByUsername(username, function (error, user) {
10 //             if (error) {
11 //                 return done(error);
12 //             }
13
14 //             if (!user) {
15 //                 return done(null, false);
16 //             }
17
18 //             if (user.password !== password) {
19 //                 return done(null, false, { message: 'Invalid credentials' });
20 //             }
21
22 //             return done(null, user);
23 //         });
24 //     }
25 // ));
26
27 // passport.serializeUser(function (user, done) {
28 //     done(null, user.id);
29 // });
30
```

```
passport.js
1 'use strict';
2
3 const passport = require('passport');
4 const Strategy = require('passport-local').Strategy;
5 const db = require('../db');
6
7 // passport.use(new Strategy(
8 //     function (username, password, done) {
9 //         db.users.findByUsername(username, function (error, user) {
10 //             if (error) {
11 //                 return done(error);
12 //             }
13
14 //             if (!user) {
15 //                 return done(null, false);
16 //             }
17
18 //             if (user.password !== password) {
19 //                 return done(null, false, { message: 'Invalid credentials' });
20 //             }
21
22 //             return done(null, user);
23 //         });
24 //     }
25 // ));
26
27 // passport.serializeUser(function (user, done) {
28 //     done(null, user.id);
29 // });
30
```



```
passport.js
1 'use strict';
2
3 const passport = require('passport');
4 const Strategy = require('passport-local').Strategy;
5 const db = require('../db');
6
7 passport.use(new Strategy(
8   function (username, password, done) {
9     // db.users.findByUsername(username, function (error, user) {
10    //   if (error) {
11    //     return done(error);
12    //   }
13
14    //   if (!user) {
15    //     return done(null, false);
16    //   }
17
18    //   if (user.password !== password) {
19    //     return done(null, false, { message: 'Invalid credentials' });
20    //   }
21
22    //   return done(null, user);
23    // });
24   }
25 ));
26
27 // passport.serializeUser(function (user, done) {
28 //   done(null, user.id);
29 // });
30
```

```
passport.js
1 'use strict';
2
3 const passport = require('passport');
4 const Strategy = require('passport-local').Strategy;
5 const db = require('../db');
6
7 passport.use(new Strategy(
8   function (username, password, done) {
9     db.users.findByUsername(username, function (error, user) {
10       if (error) {
11         return done(error);
12       }
13
14       if (!user) {
15         return done(null, false);
16       }
17
18       if (user.password !== password) {
19         return done(null, false, { message: 'Invalid credentials' });
20       }
21
22       return done(null, user);
23     });
24   }
25 ));
26
27 // passport.serializeUser(function (user, done) {
28 //   done(null, user.id);
29 // });
30
```



```
passport.js
1 'use strict';
2
3 const passport = require('passport');
4 const Strategy = require('passport-local').Strategy;
5 const db = require('../db');
6
7 passport.use(new Strategy(
8   function (username, password, done) {
9     db.users.findByUsername(username, function (error, user) {
10       if (error) {
11         return done(error);
12       }
13
14       if (!user) {
15         return done(null, false);
16       }
17
18       if (user.password !== password) {
19         return done(null, false, { message: 'Invalid credentials' });
20       }
21
22       return done(null, user);
23     });
24   }
25 ));
26
27 // passport.serializeUser(function (user, done) {
28 //   done(null, user.id);
29 // });
30
```



```
passport.js
1 'use strict';
2
3 const passport = require('passport');
4 const Strategy = require('passport-local').Strategy;
5 const db = require('../db');
6
7 passport.use(new Strategy(
8   function (username, password, done) {
9     db.users.findByUsername(username, function (error, user) {
10       if (error) {
11         return done(error);
12       }
13
14       if (!user) {
15         return done(null, false);
16       }
17
18       if (user.password !== password) {
19         return done(null, false, { message: 'Invalid credentials' });
20       }
21
22       return done(null, user);
23     });
24   }
25 ));
26
27 passport.serializeUser(function (user, done) {
28   done(null, user.id);
29 });
30
```

```
passport.js
11         return done(error);
12     }
13
14     if (!user) {
15         return done(null, false);
16     }
17
18     if (user.password !== password) {
19         return done(null, false, { message: 'Invalid credentials' });
20     }
21
22     return done(null, user);
23 });
24 }
25 ));
26
27 passport.serializeUser(function (user, done) {
28     done(null, user.id);
29 });
30
31 passport.deserializeUser(function (id, done) {
32     db.users.findById(id, function (error, user) {
33         if (error) {
34             return done(error);
35         }
36
37         done(null, user);
38     });
39 });
40
```

1

ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?]

```
$ npm install --save express-session
```

workout@0.0.0 /Users/ifdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout

```
├─┬ express@4.13.3
```

```
| └─┬ serve-static@1.10.1
```

```
|   └─┬ send@0.13.1
```

```
|     └─┬ depd@1.1.0
```

```
└─┬ express-session@1.13.0
```

```
  └─┬ cookie@0.2.3
```

```
    └─┬ depd@1.1.0
```

ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?]

```
$ █
```



```
app.js
1  'use strict';
2
3  const express = require('express');
4  const path = require('path');
5  const favicon = require('serve-favicon');
6  const logger = require('morgan');
7  const cookieParser = require('cookie-parser');
8  const bodyParser = require('body-parser');
9  // const session = require('express-session');
10 // const passport = require('./lib/passport.js');
11 // const flash = require('connect-flash');
12
13 const routes = require('./routes/index');
14 const users = require('./routes/users');
15
16 const app = express();
17
18 // view engine setup
19 app.set('views', path.join(__dirname, 'views'));
20 app.set('view engine', 'hbs');
21
22 // uncomment after placing your favicon in /public
23 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
24 app.use(logger('dev'));
25 app.use(bodyParser.json());
26 app.use(bodyParser.urlencoded({ extended: false }));
27 app.use(cookieParser());
28 app.use(express.static(path.join(__dirname, 'public')));
29 // app.use(session({ secret: 'keyboard cat', resave: false, saveUninitialized:
false}));
```




```
app.js
2
3  const express = require('express');
4  const path = require('path');
5  const favicon = require('serve-favicon');
6  const logger = require('morgan');
7  const cookieParser = require('cookie-parser');
8  const bodyParser = require('body-parser');
9  const session = require('express-session');
10 const passport = require('./lib/passport.js');
11 // const flash = require('connect-flash');
12
13 const routes = require('./routes/index');
14 const users = require('./routes/users');
15
16 const app = express();
17
18 // view engine setup
19 app.set('views', path.join(__dirname, 'views'));
20 app.set('view engine', 'hbs');
21
22 // uncomment after placing your favicon in /public
23 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
24 app.use(logger('dev'));
25 app.use(bodyParser.json());
26 app.use(bodyParser.urlencoded({ extended: false }));
27 app.use(cookieParser());
28 app.use(express.static(path.join(__dirname, 'public')));
29 // app.use(session({ secret: 'keyboard cat', resave: false, saveUninitialized:
30 // app.use(flash());
```




```
app.js
5  const favicon = require('serve-favicon');
6  const logger = require('morgan');
7  const cookieParser = require('cookie-parser');
8  const bodyParser = require('body-parser');
9  const session = require('express-session');
10 const passport = require('./lib/passport.js');
11 // const flash = require('connect-flash');
12
13 const routes = require('./routes/index');
14 const users = require('./routes/users');
15
16 const app = express();
17
18 // view engine setup
19 app.set('views', path.join(__dirname, 'views'));
20 app.set('view engine', 'hbs');
21
22 // uncomment after placing your favicon in /public
23 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
24 app.use(logger('dev'));
25 app.use(bodyParser.json());
26 app.use(bodyParser.urlencoded({ extended: false }));
27 app.use(cookieParser());
28 app.use(express.static(path.join(__dirname, 'public')));
29 app.use(session({ secret: 'keyboard cat', resave: false, saveUninitialized: false }));
30 // app.use(flash());
31
32 app.use(passport.initialize());
33 app.use(passport.session());
34
```

FOLDERS

- ▼ code
 - ▶ 1
 - ▶ 2
 - ▼ 3
 - ▶ 3.1
 - ▶ 3.2
 - ▶ 3.3
 - ▼ 3.4
 - ▼ a
 - ▼ workout
 - ▶ bin
 - ▼ db
 - Index.js
 - users.js
 - ▼ lib
 - passport.js
 - ▶ node_modules
 - ▶ public
 - ▼ routes
 - index.js
 - users.js
 - ▼ views
 - error.hbs
 - index.hbs
 - layout.hbs
 - login.hbs
 - .gitignore
 - app.js
 - package.json
 - start.js
- ▶ workout
- ▶ workout-modified
- ▶ 3.5
- ▶ 4

index.js

```
1 |exports.users = require('./users');  
2
```



```
users.js
1 'use strict';
2
3 const records = [
4   { id: 1, username: 'Andrew', password: 'secret', role: 'admin' },
5   { id: 2, username: 'Cindy', password: 'forgot', role: 'user' }
6 ];
7
8 exports.findById = function (id, callback) {
9   process.nextTick(function () {
10     const idx = id - 1;
11
12     if (records[idx] && id == records[idx].id) {
13       callback(null, records[idx]);
14     } else {
15       callback(new Error('User ' + id + ' does not exist'));
16     }
17   });
18 }
19
20 exports.findByUsername = function (username, callback) {
21   process.nextTick(function () {
22     for (let i = 0, length = records.length; i < length; i++) {
23       let record = records[i];
24
25       if (username === record.username) {
26         return callback(null, record);
27       }
28     }
29
30     return callback(null, null);
```



```
index.js
1 'use strict';
2
3 const express = require('express');
4 const router = express.Router();
5 const passport = require('../lib/passport');
6 // const ensureLoggedIn = require('connect-ensure-login').ensureLoggedIn;
7
8 /* GET home page. */
9 router.get('/', function(request, response, next) {
10     response.render('index', { title: 'Express' });
11 });
12
13 router.route('/login')
14     .get(function(request, response) {
15         response.render('login');
16     })
17     .post(passport.authenticate('local', {
18         successRedirect: '/profile',
19         failureRedirect: '/login'
20         // failureFlash: true
21     })))
22 ;
23
24 // router.get('/logout', function(request, response) {
25 //     request.logout();
26 //
27 //     response.redirect('/');
28 // });
29
30 // router.get('/profile', function(request, response) {
```

```

1  'use strict';
2
3  const express = require('express');
4  const router = express.Router();
5  const passport = require('../lib/passport');
6  // const ensureLoggedIn = require('connect-ensure-login').ensureLoggedIn;
7
8  /* GET home page. */
9  router.get('/', function(request, response, next) {
10     response.render('index', { title: 'Express' });
11 });
12
13 router.route('/login')
14     .get(function (request, response) {
15         response.render('login');
16     })
17     .post(passport.authenticate('local', {
18         successRedirect: '/profile',
19         failureRedirect: '/login'
20         // failureFlash: true
21     }));
22 ;
23
24 router.get('/logout', function (request, response) {
25     request.logout();
26
27     response.redirect('/');
28 });
29
30 // router.get('/profile', function (request, response) {

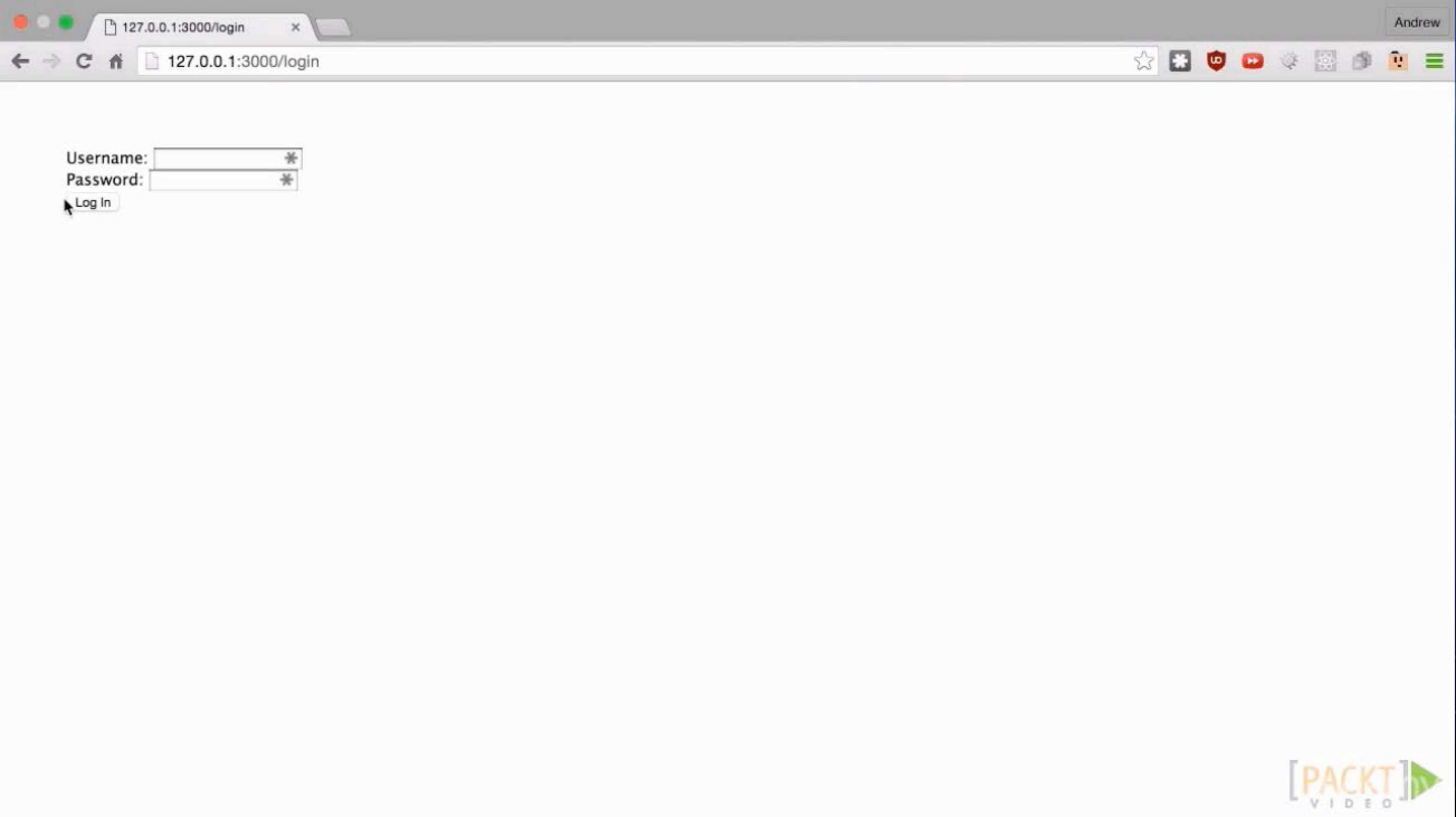
```




```
index.js
4  const router = express.Router();
5  const passport = require('../lib/passport');
6  // const ensureLoggedIn = require('connect-ensure-login').ensureLoggedIn;
7
8  /* GET home page. */
9  router.get('/', function(request, response, next) {
10     response.render('index', { title: 'Express' });
11 });
12
13 router.route('/login')
14     .get(function (request, response) {
15         response.render('login');
16     })
17     .post(passport.authenticate('local', {
18         successRedirect: '/profile',
19         failureRedirect: '/login'
20         // failureFlash: true
21     })))
22 ;
23
24 router.get('/logout', function (request, response) {
25     request.logout();
26
27     response.redirect('/');
28 });
29
30 router.get('/profile', function (request, response) {
31     response.send('Hello ' + request.user.username);
32 });
33
```


- ▼ code
 - ▶ 1
 - ▶ 2
 - ▼ 3
 - ▶ 3.1
 - ▶ 3.2
 - ▶ 3.3
 - ▼ 3.4
 - ▼ a
 - ▼ workout
 - ▶ bin
 - ▼ db
 - index.js
 - users.js
 - ▼ lib
 - passport.js
 - ▶ node_modules
 - ▶ public
 - ▼ routes
 - index.js
 - users.js
 - ▼ views
 - error.hbs
 - index.hbs
 - layout.hbs
 - login.hbs
 - ▶ 3.5
 - ▶ 4

```
login.hbs
1 <form action="/login" method="post">
2   <div>
3     <label>Username:</label>
4     <input type="text" name="username" />
5   </div>
6
7   <div>
8     <label>Password:</label>
9     <input type="password" name="password" />
10  </div>
11
12  <div>
13    <input type="submit" value="Log In" />
14  </div>
15 </form>
16
```



Username:

Password:

Log In

```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?]  
$ npm install --save connect-flash  
workout@0.0.0 /Users/ifdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout  
└-- connect-flash@0.1.1
```

```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?]  
$ █
```




```
app.js
1  'use strict';
2
3  const express = require('express');
4  const path = require('path');
5  const favicon = require('serve-favicon');
6  const logger = require('morgan');
7  const cookieParser = require('cookie-parser');
8  const bodyParser = require('body-parser');
9  const session = require('express-session');
10 const passport = require('./lib/passport.js');
11 const flash = require('connect-flash');
12
13 const routes = require('./routes/index');
14 const users = require('./routes/users');
15
16 const app = express();
17
18 // view engine setup
19 app.set('views', path.join(__dirname, 'views'));
20 app.set('view engine', 'hbs');
21
22 // uncomment after placing your favicon in /public
23 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
24 app.use(logger('dev'));
25 app.use(bodyParser.json());
26 app.use(bodyParser.urlencoded({ extended: false }));
27 app.use(cookieParser());
28 app.use(express.static(path.join(__dirname, 'public')));
29 app.use(session({ secret: 'keyboard cat', resave: false, saveUninitialized: false }));
30 // app.use(flash());
```




```
app.js
2
3  const express = require('express');
4  const path = require('path');
5  const favicon = require('serve-favicon');
6  const logger = require('morgan');
7  const cookieParser = require('cookie-parser');
8  const bodyParser = require('body-parser');
9  const session = require('express-session');
10 const passport = require('./lib/passport.js');
11 const flash = require('connect-flash');
12
13 const routes = require('./routes/index');
14 const users = require('./routes/users');
15
16 const app = express();
17
18 // view engine setup
19 app.set('views', path.join(__dirname, 'views'));
20 app.set('view engine', 'hbs');
21
22 // uncomment after placing your favicon in /public
23 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
24 app.use(logger('dev'));
25 app.use(bodyParser.json());
26 app.use(bodyParser.urlencoded({ extended: false }));
27 app.use(cookieParser());
28 app.use(express.static(path.join(__dirname, 'public')));
29 app.use(session({ secret: 'keyboard cat', resave: false, saveUninitialized: false }));
30 app.use(flash());
31
```




```
index.js
1 'use strict';
2
3 const express = require('express');
4 const router = express.Router();
5 const passport = require('../lib/passport');
6 // const ensureLoggedIn = require('connect-ensure-login').ensureLoggedIn;
7
8 /* GET home page. */
9 router.get('/', function(request, response, next) {
10     response.render('index', { title: 'Express' });
11 });
12
13 router.route('/login')
14     .get(function(request, response) {
15         response.render('login', { error: request.flash('error') });
16     })
17     .post(passport.authenticate('local', {
18         successRedirect: '/profile',
19         failureRedirect: '/login'
20         // failureFlash: true
21     }));
22 ;
23
24 router.get('/logout', function(request, response) {
25     request.logout();
26
27     response.redirect('/');
28 });
29
30 router.get('/profile', function(request, response) {
```



```
index.js
1 'use strict';
2
3 const express = require('express');
4 const router = express.Router();
5 const passport = require('../lib/passport');
6 // const ensureLoggedIn = require('connect-ensure-login').ensureLoggedIn;
7
8 /* GET home page. */
9 router.get('/', function(request, response, next) {
10     response.render('index', { title: 'Express' });
11 });
12
13 router.route('/login')
14     .get(function(request, response) {
15         response.render('login', { error: request.flash('error') });
16     })
17     .post(passport.authenticate('local', {
18         successRedirect: '/profile',
19         failureRedirect: '/login',
20         failureFlash: true
21     })))
22 ;
23
24 router.get('/logout', function(request, response) {
25     request.logout();
26
27     response.redirect('/');
28 });
29
30 router.get('/profile', function(request, response) {
```



```
login.hbs
1 <p>{{error}}</p>
2
3 <form action="/login" method="post">
4   <div>
5     <label>Username:</label>
6     <input type="text" name="username" />
7   </div>
8
9   <div>
10    <label>Password:</label>
11    <input type="password" name="password" />
12  </div>
13
14  <div>
15    <input type="submit" value="Log In" />
16  </div>
17 </form>
18
```


Cannot read property 'username' of undefined

TypeError: Cannot read property 'username' of undefined

```
at /Users/efdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout/routes/index.js:31:42
at Layer.handle [as handle_request] (/Users/efdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout/node_modules/express/lib/router/index.js:131:13)
at next (/Users/efdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout/node_modules/express/lib/router/index.js:112:3)
at Layer.handle [as handle_request] (/Users/efdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout/node_modules/express/lib/router/index.js:277:22)
at Function.process_params (/Users/efdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout/node_modules/express/lib/router/index.js:271:10)
at Function.handle (/Users/efdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout/node_modules/express/lib/router/index.js:176:3)
at router (/Users/efdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout/node_modules/express/lib/router/index.js:46:12)
```



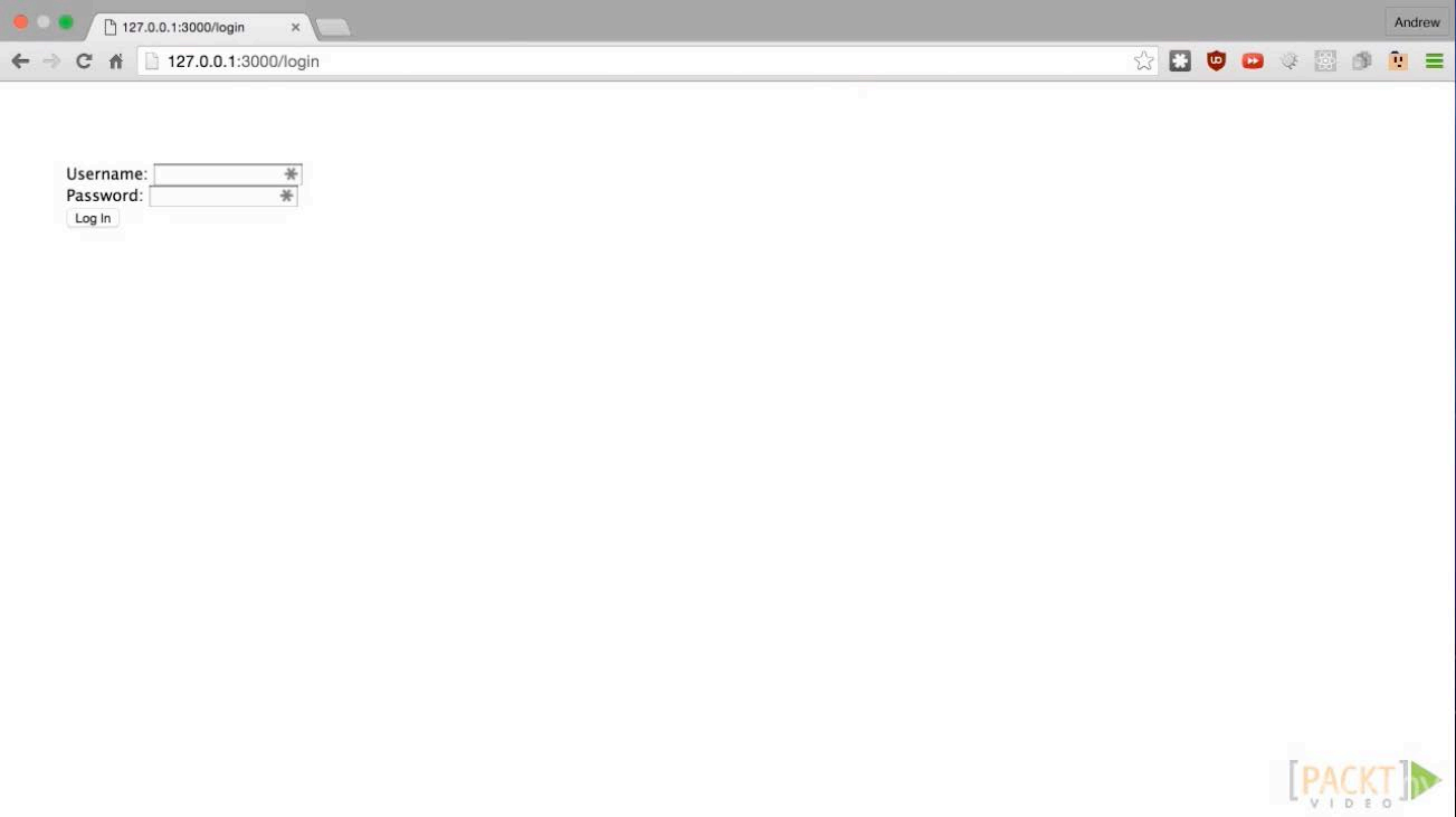
```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?]
$ npm install --save connect-ensure-login
workout@0.0.0 /Users/ifdattic/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout
└-- connect-ensure-login@0.1.1
```

```
ifdattic at ifdattic-mbp13 in ~/projects/nodejs-mongodb-video-course/code/3/3.4/a/workout on master [!?]
$ █
```

```
index.js
1 'use strict';
2
3 const express = require('express');
4 const router = express.Router();
5 const passport = require('../lib/passport');
6 const ensureLoggedIn = require('connect-ensure-login').ensureLoggedIn;
7
8 /* GET home page. */
9 router.get('/', function(request, response, next) {
10     response.render('index', { title: 'Express' });
11 });
12
13 router.route('/login')
14     .get(function(request, response) {
15         response.render('login', { error: request.flash('error') });
16     })
17     .post(passport.authenticate('local', {
18         successRedirect: '/profile',
19         failureRedirect: '/login',
20         failureFlash: true
21     }));
22 ;
23
24 router.get('/logout', function(request, response) {
25     request.logout();
26
27     response.redirect('/');
28 });
29
30 router.get('/profile', function(request, response) {
```



```
index.js
1 'use strict';
2
3 const express = require('express');
4 const router = express.Router();
5 const passport = require('../lib/passport');
6 const ensureLoggedIn = require('connect-ensure-login').ensureLoggedIn;
7
8 /* GET home page. */
9 router.get('/', function(request, response, next) {
10     response.render('index', { title: 'Express' });
11 });
12
13 router.route('/login')
14     .get(function(request, response) {
15         response.render('login', { error: request.flash('error') });
16     })
17     .post(passport.authenticate('local', {
18         successRedirect: '/profile',
19         failureRedirect: '/login',
20         failureFlash: true
21     })))
22 ;
23
24 router.get('/logout', function(request, response) {
25     request.logout();
26
27     response.redirect('/');
28 });
29
30 router.get('/profile', ensureLoggedIn(), function(request, response) {
```

Username:

Password:

Log In

Next Video

Templating with Handlebars