Search for Strategies

>

Introduction

Login Prompt

Verify Password

Establish Session

Log Out

Sign Up

Establish Session

Once we've verified the user's password, we need to a login session to remember the fact that the user has authenticated as they navigate the app.

To do that, we'll add session support. Begin by installing the necessary dependencies:

```
$ npm install express-session
$ npm install connect-sqlite3
```

Open `'app.js'` and `require` the additional dependencies at line 8, below where `'morgan'` is `require`'d:

```
var logger = require('morgan');
var passport = require('passport');
var session = require('express-session');

var SQLiteStore = require('connect-sqlite3')(session);
```

Add the following code at line 29, after `express.static` middleware, to maintain and authenticate the session.

```
app.use(express.static(path.join(__dirname, 'public')));
app.use(session({
  secret: 'keyboard cat',
  resave: false,
  saveUninitialized: false,
  store: new SQLiteStore({ db: 'sessions.db', dir: './var/db' })
}));
app.use(passport.authenticate('session'));
```

Finally, we need to configure Passport to manage the login session. Open `'routes/auth.js'` and add the following code at line 22:

```
passport.serializeUser(function(user, cb) {
  process.nextTick(function() {
    cb(null, { id: user.id, username: user.username });
  });
});

passport.deserializeUser(function(user, cb) {
  process.nextTick(function() {
    return cb(null, user);
  });
});
```

Now, let's retry signing in. Open http://localhost:3000, click "Sign in," and enter the following credentials:

```
Username: alice
Password: letmein
```

Click "Sign in."

We are logged in! Go ahead and enter some tasks you've been needing to get done.

At this point, users can sign in with a username and password! Next, we will add the ability to sign out.