

Exercise (Instructions): Express Sessions

Objectives and Outcomes

In this exercise you will use Express sessions to track authenticated users so as to enable authenticated access to server resources. At the end of this exercise you will be able to:

- Set up your Express server to use Express sessions to track authenticated users
- Enable clients to access secure resources on the server after authentication.

Installing express-session

- Still in the basic-auth folder, install *express-session* and *session-file-store* Node modules as follows:

```
npm install express-session session-file-store --save
```

Using express-session

- Then create a new file named *server-3.js* and add the following code to it:

```
var express = require('express');
var morgan = require('morgan');
var session = require('express-session');
var FileStore = require('session-file-store')(session);

var hostname = 'localhost';
var port = 3000;

var app = express();

app.use(morgan('dev'));
app.use(session({
  name: 'session-id',
  secret: '12345-67890-09876-54321',
  saveUninitialized: true,
  resave: true,
  store: new FileStore()
}));
```

```

function auth (req, res, next) {
  console.log(req.headers);
  if (!req.session.user) {
    var authHeader = req.headers.authorization;
    if (!authHeader) {
      var err = new Error('You are not authenticated!');
      err.status = 401;
      next(err);
      return;
    }
    var auth = new Buffer(authHeader.split(' ')[1], 'base64').toString().split(':');
    var user = auth[0];
    var pass = auth[1];
    if (user == 'admin' && pass == 'password') {
      req.session.user = 'admin';
      next(); // authorized
    } else {
      var err = new Error('You are not authenticated!');
      err.status = 401;
      next(err);
    }
  }
  else {
    if (req.session.user === 'admin') {
      console.log('req.session: ', req.session);
      next();
    }
    else {
      var err = new Error('You are not authenticated!');
      err.status = 401;
      next(err);
    }
  }
}

app.use(auth);

```

```
app.use(express.static(__dirname + '/public'));

app.use(function(err, req, res, next) {
    res.writeHead(err.status || 500, {
        'WWW-Authenticate': 'Basic',
        'Content-Type': 'text/plain'
    });
    res.end(err.message);
});

app.listen(port, hostname, function(){
    console.log(`Server running at http://${hostname}:${port}/`);
});
```

- Save the changes, run the server and examine the behavior.

Conclusions

In this exercise you set up the Express server to use express-session to track authenticated users to provide access to secure resources.