

Passport

Jogesh K. Muppala



THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系



香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

Passport

- Authentication middleware for Node.js
- Modular, flexible
- Supports various strategies for authentication:
 - Local strategy
 - OpenID
 - OAuth (Facebook, Twitter, G+ etc.) single sign-on
- Supports Sessions (optional)
- Installing
`npm install passport --save`

Passport Example

```
app.post('/login', passport.authenticate('local'),  
  function(req, res) {  
    // If this function gets called, authentication was successful.  
    // `req.user` contains the authenticated user.  
    res.redirect('/users/' + req.user.username);  
  });
```

Passport Example2

```
app.get('/login', function(req, res, next) {  
  passport.authenticate('local', function(err, user, info) {  
    if (err) { return next(err); }  
    if (!user) { return res.json({ . . . }); }  
  
    req.login(user, function(err) {  
      if (err) { return next(err); }  
      return res.json({ . . . });  
    });  
  })(req, res, next);  
});
```

Passport-Local

- Passport strategy for authenticating a user with a username and password
- Installing:
 - `npm install passport-local --save`

Passport-Local Usage

```
passport.use(new LocalStrategy( function(username, password, done) {  
  User.findOne({ username: username }, function (err, user) {  
    if (err) { return done(err); }  
    if (!user) { return done(null, false); }  
    if (!user.verifyPassword(password)) { return done(null, false); }  
    return done(null, user);  
  });  
}  
));
```

Passport-Local-Mongoose

- Mongoose plugin to simplify building username and password login
- Installing:
 - `npm install passport-local-mongoose --save`
- Makes available Mongoose schema support for managing users
- Adds username, hash and salt fields to store the username, the hashed password and the salt value
- Additional methods to the User schema to support local authentication

Passport-Local-Mongoose Example

```
var mongoose = require('mongoose'),  
    Schema = mongoose.Schema,  
    passportLocalMongoose = require('passport-local-mongoose');  
  
var User = new Schema({});  
  
User.plugin(passportLocalMongoose);  
  
module.exports = mongoose.model('User', User);
```


Passport/Passport-Local Configuration

```
// requires the model with Passport-Local Mongoose plugged in  
var User = require('./models/user');
```

```
// use static authenticate method of model in LocalStrategy  
passport.use(new LocalStrategy(User.authenticate()));
```

```
// use static serialize and deserialize of model for passport session support  
passport.serializeUser(User.serializeUser());  
passport.deserializeUser(User.deserializeUser());
```