Exercise (Instructions): Using OAuth with Passport and Facebook

Objectives and Outcomes

In this exercise you will make use of Passport OAuth support through the passport-facebook module together with Facebook's OAuth support to enable user authentication within your server. At the end of this exercise, you will be able to:

- Configure your server to support user authentication based on OAuth providers
- Use Passport OAuth support through the passport-facebook module to support OAuth based authentication with Facebook for your users.

Installing passport-facebook Module

- In the *rest-server-passport* folder, install *passport-facebook* module by typing the following at the prompt:

```
     npm install passport-facebook --save
```

Refactoring Authentication Code

- Create a new file named authenticate.js and add the following code to it:

```
var passport = require('passport');
var LocalStrategy = require('passport-local').Strategy;
var User = require('./models/user');
var config = require('./config');

exports.local = passport.use(new LocalStrategy(User.authenticate()));
passport.serializeUser(User.serializeUser());
passport.deserializeUser(User.deserializeUser());
```

Updating app.js

- Update app.js to include authenticate.js module:

```
var authenticate = require('./authenticate');
```

- **Remove** the following line:

```
var LocalStrategy = require('passport-local').Strategy;
```

- Also, **remove** the following lines from the passport configuration part of app.js:

```
var User = require('./models/user');passport.use(new LocalStrategy(User.authenticate()));
passport.serializeUser(User.serializeUser());
passport.deserializeUser(User.deserializeUser());
```

Updating config.js

- Update config.js as follows:

```
module.exports = {
    'secretKey': '12345-67890-09876-54321',
    'mongoUrl' : 'mongodb://localhost:27017/conFusion',
    'facebook': {
        clientID: 'YOUR FACEBOOK APP ID',
        clientSecret: 'YOUR FACEBOOK SECRET',
        callbackURL: 'https://localhost:3443/users/facebook/callback'
    }
}
```

Updating User Model

- Open user.js from the models folder and update the User schema as follows:

```
var User = new Schema({
    username: String,
    password: String,
    OauthId: String,
    OauthToken: String,
    firstname: {
      type: String,
      default: ''
    },
    lastname: {
      type: String,
      default: ''
    },
    admin:   {
        type: Boolean,
        default: false
    }
});
```

Setting up Facebook Authentication

- Open authenticate.js and add in the following line to add Facebook strategy:

```
var FacebookStrategy = require('passport-facebook').Strategy;
```

- Then add the following code to initialize and set up Facebook authentication strategy in Passport:

```
exports.facebook = passport.use(new FacebookStrategy({
  clientID: config.facebook.clientID,
  clientSecret: config.facebook.clientSecret,
  callbackURL: config.facebook.callbackURL
  },
  function(accessToken, refreshToken, profile, done) {
    User.findOne({ OauthId: profile.id }, function(err, user) {
      if(err) {
        console.log(err); // handle errors!
      }
      if (!err && user !== null) {
        done(null, user);
      } else {
        user = new User({
          username: profile.displayName
        });
        user.OauthId = profile.id;
        user.OauthToken = accessToken;
        user.save(function(err) {
          if(err) {
            console.log(err); // handle errors!
          } else {
            console.log("saving user ...");
            done(null, user);
          }
        });
      }
    });
  }
));
```

Updating users.js

- Open users.js and add the following code to it:

```
router.get('/facebook', passport.authenticate('facebook'),
  function(req, res){});

router.get('/facebook/callback', function(req,res,next){
  passport.authenticate('facebook', function(err, user, info) {
    if (err) {
      return next(err);
    }
    if (!user) {
      return res.status(401).json({
        err: info
      });
    }
    req.logIn(user, function(err) {
      if (err) {
        return res.status(500).json({
          err: 'Could not log in user'
        });
      }
              var token = Verify.getToken(user);
              res.status(200).json({
        status: 'Login successful!',
        success: true,
        token: token
      });
    });
  })(req,res,next);
});
```

Registering your app on Facebook

- Go to https://developers.facebook.com/apps/ and register your app by following the instructions there and obtain your App ID and App Secret, and then update config.js with the information.
- Start your server and test your application. You can log in using Facebook by accessinghttps://localhost:3443/users/facebook which will redirect you to Facebook for authentication and return to your server.

Conclusions

In this exercise you learnt about using the Facebook OAuth support to enable authentication of your users and allowing them access to your server.