Урок N6

Куки и сессии

Куки и сессии - стандартные механизмы для аутентификации пользователей на сайте. В этом уроке мы научимся использовать их как "напрямую", так и в рамках express. Помимо этого, мы научимся использовать для аутентификации модуль **passport**, который помогает легко встроить на сайт аутентификацию с помощью сторонних сервисов (Twitter, Facebook и других).

Содержание урока

Чтобы передать cookie, браузер при отправке запроса отправляет заголовок 'Cookie', и при создании традиционного сервера на node мы можем прочитать его в функции-обработчике запроса:

```
console.dir(request.headers['cookie']);
```

Для установки cookie нужно отправить заголовок set-cookie вместе с ответом:

```
response.setHeader("Set-Cookie",
["type=ninja", "language=javascript"]);
```

К сожалению, удобных стандартных функций парсинга и формирования заголовков для cookie в node нет. На помощь как всегда приходят модули. В частности, модуль cookies.

```
var http    = require( "http" )
var Cookies = require( "cookies" )

// работаем с cookies в req и res
var cookies = new Cookies(req, res, keys);

// устанавливаем cookie
cookies.set( "tampered", "baz");

// читаем cookie
unsigned = cookies.get("unsigned");

// удаляем cookie через expires в прошлом
cookies.set( "tampered", "baz", {expires: Date.now()-1});
```

Для парсинга cookie подкючите middleware cookie-parser

```
var express = require('express');
var cookieParser = require('cookie-parser');

express()
        .use(cookieParser('optional secret string'))
        .use(function(req, res, next){
                res.end(JSON.stringify(req.cookies));
        }
)
```

Установить cookie легко с помощью стандартных методов объекта res:

```
res.cookie('name', 'tobi', {domain: '.example.com', path: '/admin'});
```

Удаление cookie:

```
res.clearCookie('name');
```

Для работы с сессиями подкючите middleware cookie-session

При использовании этого middleware сессии будут автоматически создаваться для кажого пользователя и все сессионные переменные автоматически попадут в req.session

Для установки переменной просто добавьте ее в объект req.session, а для удаления удалите. Для уничтожения сессии: **req.session = null**
Переменная req.session.isNew будет равна true, если сессия была создана только что.

Для более "тонкой" настройки и ручного управления сессиями и сессионными переменными также можно использовать express-session

Самым популярным модулем для аутентификации в node является passport. Он не только предоставляет удобные стандартные методы для создания сессий для аутентифицированных пользователей, но и поддерживает аутентификацию на сторонних сайтах (Twitter, Facebook и другие) с помощью готовых модулей, а также удобно интегрируется с express.

Для аутентификации с помощью passport нужно подключить сам passport (для express - как middleware) и один или несколько модулей со стратегией аутентификации. Подробный разбор примера использования passport смотрите на видео к уроку и в исходном файле passport.js

Самоконтроль

✓ Работа с cookie напрямую
✓ Работа с cookie и сессиями в express
✓ Назначение и преимущества модуля passport

Домашнее задание

1) Добавьте в созданное после предыдущего урока приложение простую аутентификацию на основе логина и пароля с функцией "запомнить меня", используя middleware для работы с cookie и сессиями.

2) Добавьте в приложение дополнительную возможность аутентификации через сторонний сервис с помощью passport