



express.js

- Сессии
- Cookies

1 | Сессии и cookies

Протокол HTTP является протоколом "без сохранения состояния". Это означает, что данный протокол не имеет встроенного способа сохранения состояния между двумя транзакциями. Т. е., когда пользователь открывает сначала одну страницу сайта, а затем переходит на другую страницу этого же сайта, то основываясь только на средствах, предоставляемых протоколом HTTP невозможно установить, что оба запроса относятся к одному пользователю. Для этого были разработаны механизмы сессий и cookies.

1 | Сессии и cookies

Сессии и cookies предназначены для хранения сведений о пользователях при переходах между несколькими страницами. При использовании сессий данные сохраняются во временных файлах на сервере. Файлы с cookies хранятся на компьютере пользователя, и по запросу отсылаются браузером серверу.

Использование сессий и cookies очень удобно и оправдано в таких приложениях как Интернет-магазины, форумы, доски объявлений, когда, во-первых, необходимо сохранять информацию о пользователях на протяжении нескольких страниц, а, во-вторых, своевременно предоставлять пользователю новую информацию.

1 | express.js - сессии

Для работы сессии в express необходимо установить пакет:

```
$ npm install express-session --save
```

1 | сессии - открытие

При работе с сессиями различают следующие этапы:

- открытие сессии
- регистрация переменных сессии и их использование
- закрытие сессии

```
var session = require('express-session');  
app.use(session({  
  secret: ' 23msdf134kasd ',  
  name: 'sid',  
  cookie: { secure: true, httpOnly: true, maxAge: 60000},  
}))
```

1 | сессии — регистрация переменных

```
app.get('/', function(req, res, next) {  
  var sess = req.session  
  if (sess.views) {  
    sess.views++  
    res.setHeader('Content-Type', 'text/html')  
    res.write('<p>views: ' + sess.views + '</p>')  
    res.write('<p>expires in: ' + (sess.cookie.maxAge / 1000) + 's</p>')  
    res.end()  
  } else {  
    sess.views = 1  
    res.end('welcome to the session demo. refresh! id' + req.session.id )  
  }  
})
```

1 | сессии — регистрация переменных

```
app.get('/', function(req, res, next) {  
  var sess = req.session  
  if (sess.views) {  
    sess.views++  
    res.setHeader('Content-Type', 'text/html')  
    res.write('<p>views: ' + sess.views + '</p>')  
    res.write('<p>expires in: ' + (sess.cookie.maxAge / 1000) + 's</p>')  
    res.end()  
  } else {  
    sess.views = 1  
    res.end('welcome to the session demo. refresh! id' + req.session.id )  
  }  
})
```


1 | уникальное id сессии

Express генерирует уникальное значение(id) для каждой сессии. Данное значение зашивается в cookies и передается клиенту.

```
app.get('/', function(req, res, next) {  
    ...  
    res.end('welcome to the session demo. refresh! id' +  
req.session.id )  
    })
```

1 | сессии - удаление

Для удаления текущей сессии служит метод `destroy()`.

```
app.get('/del', function(req, res, next) {  
  if (req.session) {  
    req.session.destroy(function() {});  
  }  
  res.end('Session destroyed ! id:' + req.session.id )  
})
```

2 | express.js — установка cookies

Установка cookies осуществляется по следующему шаблону

```
res.cookie(name, value [, options])
```

Значение куки должна быть строка либо объект json формата:

```
res.cookie('name', 'Peter', { expires: new Date(Date.now()  
+ 60000), httpOnly: true });
```

```
res.cookie('rememberme', '1', { maxAge: 900000, httpOnly:  
true });
```

```
res.cookie('cart', { items: [1,2,3] });
```

```
res.cookie('cart', { items: [1,2,3] }, { maxAge: 900000 });
```

```
res.cookie('name', 'Piter', { path: '/admin' });
```

2 | express.js — очистка cookies

Для того чтобы отменить установку переменной cookies можно воспользоваться методом `clearCookie()`

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

2 | express.js — секрет cookies

Для того чтобы отменить установку переменной cookies можно воспользоваться методом `clearCookie()`

```
app.use(cookieParser( '234kldsjf3453' ));  
res.cookie( 'name', 'tobi', { signed: true } );
```

Полезный ссылки

<http://expressjs.com/ru/> - express минималистичный веб фреймворк

<https://github.com/expressjs/session> - модуль express session

<http://expressjs.com/ru/api.html#res.cookie> - справочник