

Desenvolvimento de Software para Web



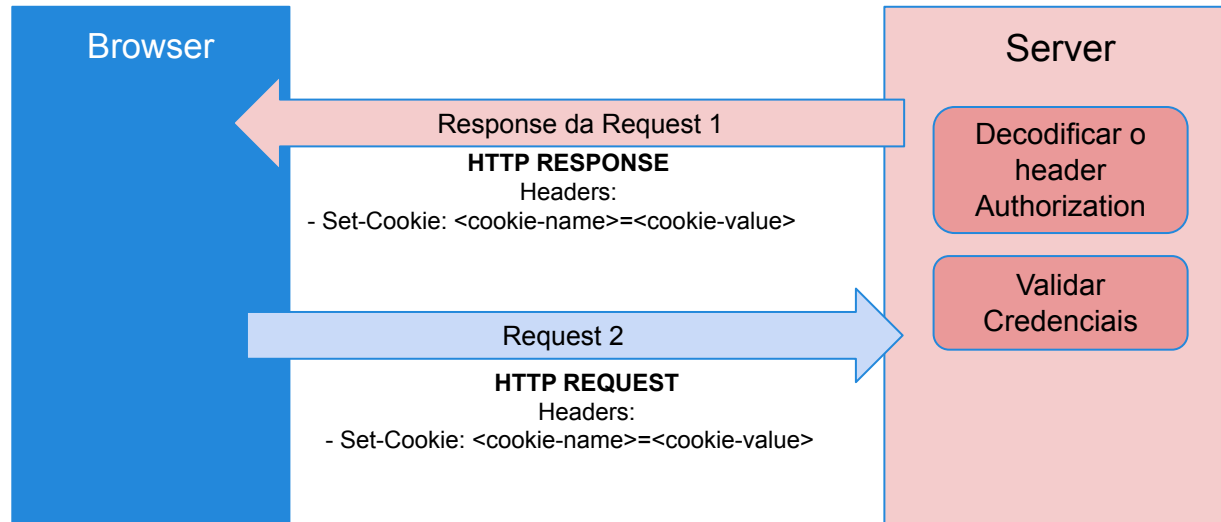
UFC - Universidade Federal do Ceará

André Meireles
andre@crateus.ufc.br

MPA - Sessão e Cookies

Cookies - Como funcionam

Referência: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Cookies>



Para aumentar a segurança, os valores podem ser criptografados

Cookies de Sessão

Node Express

Documentação do Express recomenda a utilização do [módulo cookie-session](#)

Para instalar:

```
$ npm install cookie-session
```

Express - Cookie Session

Importe o módulo e configure o middleware para todas as requisições:

```
var cookieSession = require('cookie-session')
var express = require('express')

var app = express()
.
.
.
app.use(cookieSession({
  name: 'session',
  // keys: ['c293x8b6234z82n938246bc2938x4zb234'],
  secret: 'c293x8b6234z82n938246bc2938x4zb234',
  // Cookie Options
  maxAge: 24 * 60 * 60 * 1000 // 24 hours
}))
```



Essa chave é usada para encriptar o valores

Express - Cookie Session

Para definir atributos na sessão do usuário, basta adicioná-los ao objeto req.session. Esse objeto estará sempre presente nas requisições seguintes que forem realizadas a partir de um mesmo cliente

```
database.getUser(credentials.login, credentials.password).then(user => {  
  if(user) {  
    user.password = null;  
    req.session.user = user;  
    res.redirect('loja/home');  
  }  
  else res.render('login', {errorMessage: 'Usuário ou senha inválidos'});  
})
```



Ao realizar o login, você pode adicionar os dados desejados à sessão do usuário

Express - Cookie Session

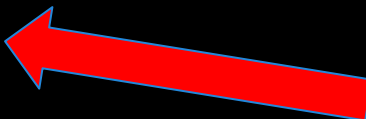
O objeto session possui atributos próprios que trazem informações adicionais sobre a sessão: se é uma sessão nova (`isNew`), se ela tem atributos (`isPopulated`) e se a sessão foi alterada (`isChanged`)

```
app.use((req, res, next) => {  
  console.log('app.use - * - req.session: ');  
  console.log(req.session);  
  console.log('app.use - * - req.session.isNew: ' + req.session.isNew);  
  console.log('app.use - * - req.session.isPopulated: ' + req.session.isPopulated);  
  console.log('app.use - * - req.session.isChanged: ' + req.session.isChanged);  
  next();  
});
```

Express - Cookie Session

Você pode acessar a sessão a partir de qualquer middleware para tomar ações, como proteger rotas e desviar fluxos

```
app.use('/loja*', (req, res, next) => {  
  if(req.session.user)  
    next();  
  else{  
    console.log('app.use - /loja* - Not authenticated, redirect to /');  
    res.redirect('/');  
  }  
});
```

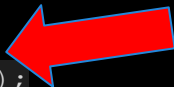


Se o usuário tentar acessar um endereço que requer autenticação mas ainda não estiver autenticado, ele é redirecionado para a página de login. Nesse caso, session.user é criado quando o usuário faz login. Mas também é possível usar sessão para usuário não logado

Express - Cookie Session

Você pode guardar quaisquer dados na sessão, porém, lembre-se que esses dados são trafegados a cada request, portanto, use com moderação, e, quando necessário, salve dados em banco de dados e recupere quando for usar

```
app.get('/loja/produto-add', (req, res) => {  
  if(req.query.item){  
    item = req.query.item;  
    if (!req.session.carrinho)  
      req.session.carrinho = [];  
    req.session.carrinho.push(item);  
  }  
  res.redirect('/loja/home');  
});
```



Um atributo chamado carrinho é criado na sessão quando ainda não existe.
Um item é adicionado ao carrinho

Express - Cookie Session

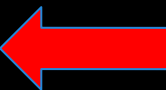
Usando os templates EJS você pode acessar e exibir os atributos da sessão

```
app.get('/loja/meucarrinho', (req, res) => {  
  res.render('loja/meucarrinho', {carrinho: req.session.carrinho});  
});
```



Adiciona o objeto carrinho para ser processado na view

```
//  
...  
<h2>Meu Carrinho</h2>  
<ul>  
  <% for (const i in carrinho) { %>  
    <li><%= carrinho[i] %> </li>  
  <% } %>  
</ul>  
...
```



Exibe os itens do carrinho

Links importantes

- <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Cookies>
- <https://expressjs.com/en/resources/middleware/cookie-session.html>