

Auth Strategies

- Basic HTTP auth
- Cookie based auth
- Token based auth
- One-Time Passwords

Basic HTTP Auth



Basic Auth

1. При обращении неавторизованного пользователя к защищенному ресурсу сервер возвращает «**401 Unauthorized**» и добавляет заголовок «**WWW-Authenticate**»
2. Браузер при получении ответа с заголовком «**WWW-Authenticate**» показывает форму для ввода логина и пароля.
3. При обращении к данному ресурсу передает заголовок «**Authorization**», где хранятся данные пользователя для аутентификации.

Basic Auth

- username and password соединяются в строку:
`username:password`
- строка кодируется в Base64
- перед этой строкой ставится ключевое слово `Basic`

Basic Auth

```
curl --header "Authorization: Basic am9objpzZWNyZXQ=" my-website.com
```

Basic Auth

×

Headers

Preview

Response

Cookies


Timing

▼General

Remote Address: [::1]:5000

Request URL: http://localhost:5000/

Request Method: GET

Status Code:  200 OK

▼Response Headers

view source

Connection: keep-alive

Content-Length: 69

Date: Mon, 23 Nov 2015 07:17:40 GMT

▼Request Headers

view source

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Accept-Encoding: gzip, deflate, sdch

Accept-Language: en-US,en;q=0.8,hu;q=0.6,nl;q=0.4,es;q=0.2,fr;q=0.2,de;q=0.2

Authorization: Basic am9objpzZWNYZXQ=

Cache-Control: max-age=0

Connection: keep-alive

Basic Auth - минусы

- на каждый запрос мы отправляем логин и пароль в открытом виде (риск утечки)
- невозможно разлогинить пользователя
- сложно инвалидировать данные логина (нужно просить пользователя их обновить)

Cookie Auth



Cookie Auth

1. Клиент отправляет логин запрос на сервер
2. При успешном логине сервер возвращает ответ с хедером Set-Cookie: **cookie name, value, expiry time** + возможно дополнительная информация.
3. Клиент отправляет поле Cookie с каждым дальнейшим запросом
4. При лог ауте, сервер отправляет обратно Set-Cooke хедер чтобы предыдущая кука устарела

Cookie Auth

▼General

Remote Address: 185.63.147.10:443

Request URL: https://www.linkedin.com/nhome/?trk=hb_signin

Request Method: GET

Status Code: 🟢 200 OK

▼Response Headers

cache-control: no-cache, no-store

content-encoding: gzip

content-type: text/html; charset=utf-8

date: Mon, 23 Nov 2015 07:25:35 GMT

expires: Thu, 01 Jan 1970 00:00:00 GMT

pragma: no-cache

server: Play

set-cookie: lidc="b=TB16:g=272:u=1:i=1448263535:t=1448349780:s=AQHTwo_H6eKMTc-ysMUDV4m_j19p94Ha"; Expires=Tue, 24 Nov 2015 07:23:00 GMT; domain=.linkedin.com;

Cookie Auth

Request Headers

```
:host: www.linkedin.com
:method: GET
:path: /home?trk=nav_responsive_tab_home
:scheme: https
:version: HTTP/1.1
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
accept-encoding: gzip, deflate, sdch
accept-language: en-US,en;q=0.8,hu;q=0.6,nl;q=0.4,es;q=0.2,fr;q=0.2,de;q=0.2
cookie: bcookie="v=2&8b44d748-8d82-46a4-8577-36c00172794b"; bscookie="v=1&201511230722494b3c36f8-d94e-4796-8b49-a5371e6804d1AQEWRbrseJHuyZDhtFipryvvTHp0Sh7o"; L1e=108b50b8; _gat=1; L1c=506114f5; visit="v=1&M"; _ga=GA1.2.96221765.1448263372; oz_props_fetch_size1_undefined=undefined; wutan=i4YA1rjunkpRvR+nkTLFo1Tc8VA9+A0Yr0i6eMU+s84=; sl="v=1&i-LQe"; li_at=AQEDARuYmFkFnETwAAABUTM7yS8AAAFRM6mmL04AHZ3tRDqVEG7yio6z6WHQZWKXW0KcCuaX8VpT1gUf8RaKq1YzYoXazo47r3u4tP28mIy5bTs3GLWoym5Xg9XQpqeT3lwdy3Ps5pLNICgTw0jCsoDa; JSESSIONID="ajax:8745093687238714270"; liap=true; lidc="b=TB16:g=272:u=1:i=1448263557:t=1448349780:s=AQH_zUY2E78NFUubZ0HCzlf0atKxLxxy"; RT=s=1448263559420&r=https%3A%2F%2Fwww.linkedin.com%2Fhome%2F%3Ftrk%3Dhb_signin; share_setting=PUBLIC; _lipt=0_1bmJJGSdPtMK4q9DLkNuLXXW5-iWZ3vLKJSmIXmUDL9otamnhKTL_Tp4xsiTWowWeXz07fMN_z7blHodRWXvgr9M72nA7ucghFT4sQG3E6fD3sF9zVmVHVynVYd9lijIAtUYh3Vz8BfPe0oLeRw dL8; lang="v=2&lang=en-us"; sdsc=1%3A1SZM1shxDNbLt36wZwCgPgVn58iw%3D
```

Cookie Auth - нюансы

- Всегда используйте HttpOnly куки. Тогда их не будет видно в `document.cookies`.
- Использовать только подписанные куки

Cookie Auth - минусы

- Появляется риск CSRF атак
- Не совместимо с REST (добавляет стейт)

Token Auth



Token Auth

- На каждый запрос отправляем токен

JWT

- Header, тип токена и алгоритм хеширования
- Payload, содержит данные
- Signature, подпись

<https://jwt.io/>

One Time Password

Когда что использовать

ВСЕМ СПАСИБО ЗА ВНИМАНИЕ!

Добавляйтесь в друзья



[@satansdeer](https://twitter.com/satansdeer)



youtube.com/user/satansdeer1



youtube.com/user/loftblog

#loftblog #loftschool