
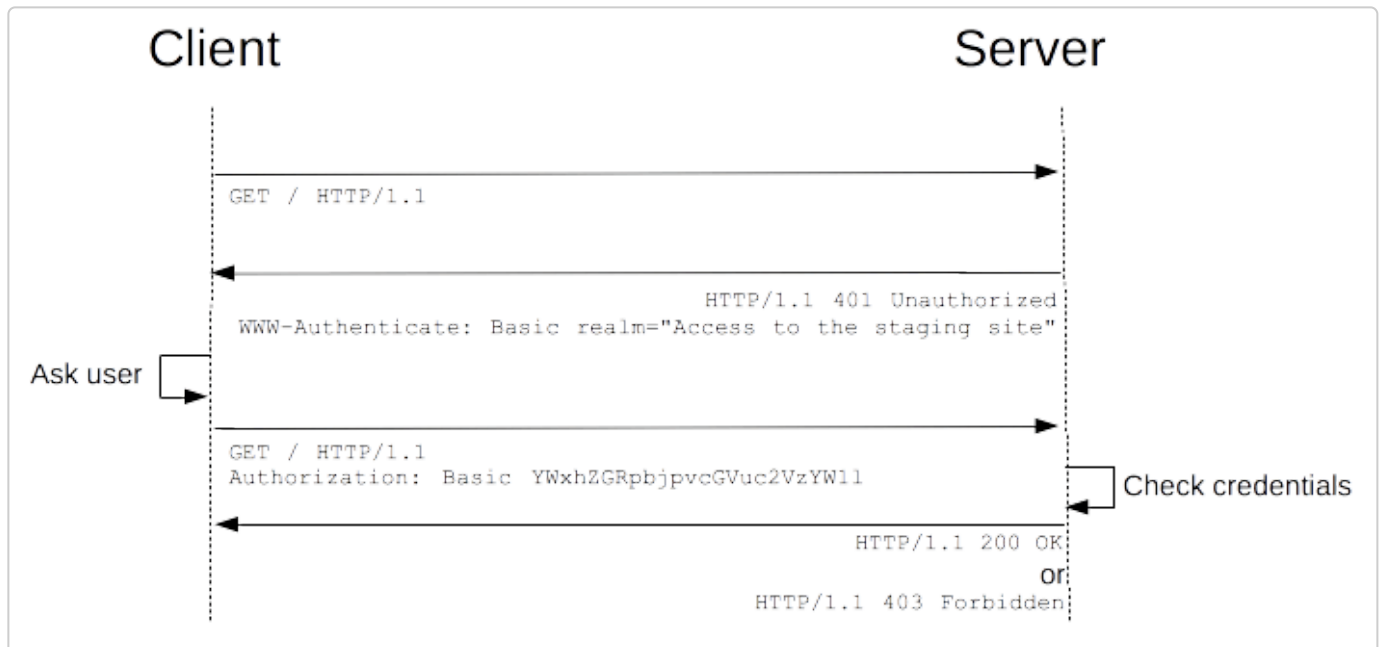
 This page was translated from English by the community. [Learn more and join the MDN Web Docs community.](#)

Autenticação HTTP

O HTTP fornece uma estrutura geral para controle de acesso e autenticação. A autenticação HTTP mais comum é fundamentada no esquema "Basic". Esta página introduz a estrutura HTTP para autenticação e mostra como restringir acesso ao seu servidor usando o esquema "Basic".

A estrutura geral de autenticação HTTP

[RFC 7235](#)  define a estrutura de autenticação HTTP que pode ser usada por um servidor para definir uma solicitação ("[challenge_\(en-US\)](#)") do cliente e para um cliente fornecer informações de autenticação. A pergunta e resposta segue um caminho como esse: O servidor responde ao cliente com uma mensagem do tipo [401](#) (Não autorizado) e fornece informações de como autorizar com um cabeçalho de resposta [WWW-Authenticate](#) contendo ao menos uma solicitação. Um cliente que deseja autenticar-se com um servidor pode fazer isso incluindo um campo de cabeçalho de solicitação [WWW-Authenticate](#) com as credenciais. Usualmente um cliente apresentará uma solicitação de senha ao usuário e, em seguida, emitirá uma solicitação incluindo o cabeçalho [Authorization](#) correto.



No caso de uma autorização "Basic" (como a mostrada na figura), a troca **deve** acontecer por meio de uma conexão HTTP (TLS) para ser segura.

Autenticação de Proxy


O mesmo mecanismo de solicitação e resposta pode ser usado para uma *autenticação de proxy*. Neste caso, é um proxy intermediário que requer autenticação. Como ambas autenticação de recurso e autenticação de proxy podem coexistir, um conjunto diferente de códigos de cabeçalhos e status torna-se necessário. No caso de proxys, o código de status de solicitação é [407](#) (Autenticação de Proxy necessária), o cabeçalho de resposta [Proxy-Authenticate](#) contém ao menos uma solicitação aplicável para o proxy, e o cabeçalho de pedido [Proxy-Authorization](#) é usado para fornecer as credenciais ao servidor proxy.

Acesso proibido


Se um servidor proxy recebe credenciais válidas, mas que não são adequadas para ter acesso a um determinado recurso, o servidor responderá com o código de status [Forbidden 403](#). Ao contrário de [401 Unauthorized](#) ou [407 Proxy Authentication Required](#), a autenticação é impossível para este usuário.

Autenticação de imagens de origem cruzada

Um potencial buraco de segurança que foi corrigido recentemente pelos navegadores é a autenticação de imagens cross-site (origem cruzada). Do [Firefox 59 \(en-US\)](#) em diante,

recursos de imagem carregados de diferentes origens não são mais capazes de adicionar diálogos de autenticação HTTP ([bug 1423146](#) ) , impedindo que as credencias do usuário sejam roubadas se invasores conseguissem incorporar uma imagem arbitrária em uma página de terceiros.

A codificação de caracteres da autenticação HTTP

Os navegadores usam a codificação `utf-8` para nomes de usuários e senhas. Firefox usava `ISO-8859-1` , mas alterou para `utf-8` por questões de compatibilidade com outros navegadores, assim como para evitar os potenciais problemas descritos em [bug 1419658](#) .

Cabeçalhos `WWW-Authenticate` e `Proxy-Authenticate`

Os cabeçalhos de resposta [WWW-Authenticate](#) e [Proxy-Authenticate](#) definem o método de autenticação que deve ser usado para ganhar acesso a um recurso. Eles precisam especificar que esquema de autenticação é usado para que o cliente que deseja autorizar saiba como fornecer as credenciais. A sintaxe para esses cabeçalhos é a seguinte:

```
WWW-Authenticate: <type> realm=<realm>  
Proxy-Authenticate: <type> realm=<realm>
```

`<type>` é o esquema de autenticação ("Basic" é o esquema mais comum e será introduzido abaixo). O *realm* é usado para indicar a área protegida ou o escopo de proteção. Poderia ser uma mensagem parecida com "Access to the staging site" (Acesso ao site de teste), portanto o usuário saberá qual área ele está tentando acessar.


Cabeçalhos `Authorization` e `Proxy-Authorization`








Os cabeçalhos de solicitação [Authorization](#) e [Proxy-Authorization](#) contém as credenciais para autenticar um agente de usuário com um servidor proxy. Aqui o tipo é novamente necessário, seguido pelas credenciais, que podem ser codificadas ou criptografadas dependendo do esquema de autenticação usado.

```
Authorization: <type> <credentials>  
Proxy-Authorization: <type> <credentials>
```


Esquemas de autenticação

A estrutura geral de autenticação HTTP é usado por vários esquemas de autenticação. Os esquemas podem divergir na força da segurança e na disponibilidade do software cliente ou servidor.

O esquema mais comum de autenticação é o "Basic", que é introduzido com mais detalhes abaixo. IANA mantém uma [lista de esquemas de autenticação](#) , mas existem outros esquemas oferecidos por serviços de hospedagem, como Amazon AWS. Os esquemas de autenticação comuns incluem:

- **Basic** (veja [RFC 7617](#) , credenciais codificadas em base64. Veja abaixo mais informações.),
- **Bearer** (veja [RFC 6750](#) , tokens bearer (de portador) para acessar recursos protegidos por OAuth 2.0),
- **Digest** (veja [RFC 7616](#) , apenas hash md5 é suportado no Firefox, veja [bug 472823](#)  para o suporte de encriptação SHA),
- **HOBA** (veja [RFC 7486](#) , esboço), **HTTP Origin-Bound Authentication** (Autenticação Vinculada à Origem HTTP), baseado em assinatura digital),
- **Mutual** (veja [draft-ietf-httpauth-mutual](#) ,)
- **AWS4-HMAC-SHA256** (veja [Documentação AWS](#) .

Esquema Basic de autenticação

O esquema "Basic" de autenticação HTTP é definido em [RFC 7617](#) , transmitindo credenciais como pares de ID/senhas de usuários, codificadas usando base64.

Segurança da autenticação básica

Como o ID e senha do usuário são transmitidos através da rede como texto claro (é codificado em base64, mas base64 é uma codificação reversível), o esquema básico de autenticação não é seguro. HTTPS / TLS devem ser usados em conjunto com autenticação básica. Sem esses aprimoramentos de segurança adicionais, a autenticação básica não deve ser usada para proteger informação sensível ou valiosa.

Restringindo acesso no Apache e autorização básica

Para proteger com senha um diretório em um servidor Apache, você precisará de um arquivo `.htaccess` e um `.htpasswd`.

O arquivo `.htaccess` normalmente parece com isso:

```
AuthType Basic
AuthName "Access to the staging site"
AuthUserFile /path/to/.htpasswd
Require valid-user
```

O arquivo `.htaccess` referencia um arquivo `.htpasswd` em que cada linha contém um nome de usuário e senha separados por dois pontos (":"). Você não pode ver as senhas reais porque foram [criptografadas](#) (em md5, neste caso). Note que você pode renomear seu arquivo `.htpasswd` caso queira, mas tenha em mente que este arquivo não deve ser acessado por ninguém. (Apache normalmente é configurado para prevenir acesso aos arquivos `.ht*`).

```
aladdin:$apr1$ZjTqBB3f$IF9gdYAGlMrs2fuINjHsz.
user2:$apr1$004r.y2H$/vEkesPhVInBByJUKXitA/
```

Restringindo acesso no nginx e autenticação básica

No nginx, você precisará especificar uma área que que você protegerá e a diretiva `auth_basic` que fornece o nome para a área protegida por senha. A diretiva `auth_basic_user_file` aponta para um arquivo `.htpasswd` contendo as credenciais do usuário criptografadas, assim como no exemplo Apache acima.

```
location /status {
    auth_basic "Access to the staging site";
    auth_basic_user_file /etc/apache2/.htpasswd;
}
```

Acesso usando as credenciais na URL

Vários clientes também permitem que você evite o prompt de login usando uma URL codificada contendo o nome de usuário e senha como esta:

https://username:password@www.example.com/



O uso destas URLs está obsoleto. No Chrome, a parte `username:password@` nas URLs é [retirada](#) por razões de segurança. No Firefox, é verificado se o site realmente requer autenticação e, se não, Firefox alertará o usuário com uma mensagem "Você está prestes a logar no site ["www.example.com"](#) com seu nome de usuário "username", mas o website não requer autenticação. Isso pode ser uma tentativa de enganá-lo".

Veja também

- [WWW-Authenticate](#)
- [Authorization](#)
- [Proxy-Authorization](#)
- [Proxy-Authenticate](#)
- [401](#) , [403](#) , [407](#)

Last modified: 7 de dez. de 2022, [by MDN contributors](#)