Entendendo e Documentando RESTful APIs

Autenticação com HTTP

Autenticação com HTTP

Os mecanismos padrões de autenticação com HTTP são definidos como **Basic** (básica) e **Digest** (resumida).

Autenticação com HTTP

Esses dois mecanismos foram projetados seguindo as constraints REST, ou seja, eles são stateless.

Autenticação com HTTP

Nesses dois mecanismos o conjunto usuário/senha é incluído em cada requisição, **codificado em Base64** para a autenticação **Basic** e com um **hash MD5** para a autenticação **Digest**.

Autenticação com HTTP

A definição do funcionamento da autenticação em HTTP pode ser encontrada em…

`https://tools.ietf.org/html/rfc2617`

Autenticação com HTTP

A documentação informa que para uma autenticação o cliente deve enviar o header **Authorization** no seguinte formato:

```
Authorization: auth-scheme hashed-credentials
```

Autenticação com HTTP

Sendo assim, uma autenticação básica seria:

ex: `Authorization: Basic am9objpwYXNz`

`https://en.wikipedia.org/wiki/Basic_access_authentication`

Autenticação com HTTP

Para fazer uma requisição de autenticação básica HTTP via cURL teríamos:

```
curl -u jack:pass http://www.example.com
```

Autenticação com HTTP

Já para autenticação HTTP do tipo Digest, teríamos:

```
curl --digest -u jack:pass
http://www.example.com
```

Autenticação com HTTP

Em retorno à requisição feita com o header Authorization, caso as credenciais não sejam autorizadas, o servidor deve retornar o status code **401 Unauthorized** e setar o header **WWW-Authenticate** com o **tipo de autenticação** que deve ser usado e **qual o domínio** (realm).

```
WWW-Authenticate: Basic realm="Perfil"
```

Autenticação com HTTP

A diretiva de domínio “**realm**” é opcional e indica a proteção de um determinado espaço, pois uma mesma aplicação pode-se ter diferentes áreas protegidas usando diferente esquemas de autenticação.