**AUTH**

An auth (Authorization code) is a unique string of random letters and numbers assigned to a domain. When you [transfer a domain](https://www.dynadot.com/domain/transfer.html) to another registrar, its auth code is required to start the transfer. This helps prevent [domain hijackers](https://www.dynadot.com/community/help/question/domain-hijacking) from transferring a domain without the owner's permission.

Authorization codes must be retrieved from the current domain registrar where the domain is registered.

Auth codes are also sometimes referred to as EPP codes or authinfo codes.

Basic authentication (BA) implementation is the simplest technique for enforcing [access controls](https://en.wikipedia.org/wiki/Access_controls) to web resources because it does not require [cookies](https://en.wikipedia.org/wiki/HTTP_cookie), session identifiers, or login pages; rather, HTTP Basic authentication uses standard fields in the [HTTP header](https://en.wikipedia.org/wiki/HTTP_header).

**Server side**:When the server wants the user agent to authenticate itself towards the server, the server must respond appropriately to unauthenticated requests.

**Client side**:When the server wants the user agent to authenticate itself towards the server, the server must respond appropriately to unauthenticated requests.

**URL Encoding**:A client may avoid a login prompt when accessing a basic access authentication by prepending *username*:*password*@ to the hostname in the URL.

**CORS**

**Cross-Origin Resource Sharing** ([CORS](https://developer.mozilla.org/en-US/docs/Glossary/CORS)) is a mechanism that uses additional [HTTP](https://developer.mozilla.org/en-US/docs/Glossary/HTTP) headers to tell browsers to give a web application running at one [origin](https://developer.mozilla.org/en-US/docs/Glossary/origin), access to selected resources from a different origin. A web application executes a cross-origin HTTP request when it requests a resource that has a different origin (domain, protocol, or port) from its own.

**What requests use CORS?**

This [cross-origin sharing standard](https://fetch.spec.whatwg.org/#http-cors-protocol) can enable cross-site HTTP requests for:

* Invocations of the [XMLHttpRequest](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest) or Fetch APIs
* Web Fonts (for cross-domain font usage in @font-face within CSS), [so that servers can deploy TrueType fonts that can only be cross-site loaded and used by web sites that are permitted to do so.](https://www.w3.org/TR/css-fonts-3/#font-fetching-requirements)
* [WebGL textures](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Using_textures_in_WebGL).
* Images/video frames drawn to a canvas using [drawImage()](https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/drawImage).
* [CSS Shapes from images.](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Shapes/Shapes_From_Images)

A “simple request” is one that **meets all the following conditions**:

* One of the allowed methods:
  + [GET](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/GET)
  + [HEAD](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/HEAD)
  + [POST](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/POST)
* Apart from the headers automatically set by the user agent (for example, [Connection](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Connection), [User-Agent](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent), or [the other headers defined in the Fetch spec as a “forbidden header name”](https://fetch.spec.whatwg.org/#forbidden-header-name)), the only headers which are allowed to be manually set are [those which the Fetch spec defines as a “CORS-safelisted request-header”](https://fetch.spec.whatwg.org/#cors-safelisted-request-header), which are:
  + [Accept](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Accept)
  + [Accept-Language](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Accept-Language)
  + [Content-Language](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Language)
  + [Content-Type](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Type) (but note the additional requirements below)
  + [DPR](http://httpwg.org/http-extensions/client-hints.html#dpr)
  + Downlink
  + [Save-Data](http://httpwg.org/http-extensions/client-hints.html#save-data)
  + [Viewport-Width](http://httpwg.org/http-extensions/client-hints.html#viewport-width)
  + [Width](http://httpwg.org/http-extensions/client-hints.html#width)
* The only allowed values for the [Content-Type](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Type) header are:
  + application/x-www-form-urlencoded
  + multipart/form-data
  + text/plain
* No event listeners are registered on any XMLHttpRequestUpload object used in the request; these are accessed using the [XMLHttpRequest.upload](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/upload) property.
* No [ReadableStream](https://developer.mozilla.org/en-US/docs/Web/API/ReadableStream) object is used in the request.