

REST API Authentication

Since the introduction of REST API v2 with eZ Publish 5.0, two authentication methods are supported: session, and basic.

- **Session-based authentication** is meant to be used for AJAX operations. It will let you re-use the visitor's session to execute operations with their permissions.
- **Basic authentication** is often used when writing cross-server procedures, when one remote application executes operations on one/several eZ Platform instances (remote publishing, maintenance, etc).

As of eZ Platform the default authentication method is Session-based as this is needed for UI.

Session based authentication

This authentication method requires a Session cookie to be sent with each request.

If this authentication method is used with a web browser, this session cookie is automatically available as soon as your visitor logs in. Add it as a cookie to your REST requests, and the user will be authenticated.

Logging in

It is also possible to create a session for the visitor if they aren't logged in yet. This is done by sending a **POST** request to `/user/sessions`. Logging out is done using a **DELETE** request on the same resource.

More information

[Session-based authentication chapter of the REST specifications](#)

HTTP Basic authentication

To enable HTTP Basic authentication, you need to edit `app/config/security.yml`, and add/uncomment the following block. Note that this is enabled by default.

ezplatform.yml

```
ezpublish_rest:
    pattern: ^/api/ezp/v2
    stateless: true
    ezpublish_http_basic:
        realm: eZ Publish REST API
```

Basic authentication requires the username and password to be sent (`username:password`), based 64 encoded, with each request, as explained in [RFC 2617](#).

Most HTTP client libraries as well as REST libraries do support this method one way or another.

Raw HTTP request with basic authentication

```
GET / HTTP/1.1
Host: api.example.com
Accept: application/vnd.ez.api.Root+json
Authorization: Basic QWxhZGRpbjpvcmVudVlHbnlc2FtZQ==
```