Authenticating to the API

API URIs

API URIs have the following prefixs:

> https://admin.openathens.net/api/v1/<domain-id> (read and write, e.g. only for account management)
>
> https://login.openathens.net/api/v1/<domain-id> (read only, and high availability e.g. for user authentications, password resets, etc)

Where domain-id is the unique ID for the OpenAthens domain.

Authentication

All access to the API is authenticated. Authentication is used to

- control programmatic-access to the API from applications (such as an administration or registration UI).
- verify the credentials of accounts in the system, for instance to provide a login form.

The API supports the following methods of authentication:

- HTTP Basic Authentication, as described in RFC 2617.
- API key authentication, using a time-limited key previously allocated by the API.
- API key authentication, using a long-lived key previously allocated by the web console.

Prerequisites

In order to authenticate to the API, you must

- know the name of a valid OpenAthens domain.
- have an API client that is capable of connecting to an HTTPS URL and sending an receiving JSON objects.
- have the credentials of a user or administrator account within the OpenAthens domain or have a previously assigned API key for an administrator account.

HTTP Basic Authentication

Basic authentication can be used to verify the credentials for an account (these would probably have been collected via a login form), or to obtain an API key for subsequent access to the API.

For authenticating end-users, see the API usage examples section. To obtain or modify user account details use a management API key as follows.

Procedure

To authenticate to the API, an HTTP request (the method depends on the particular API call being made) should be made with an HTTP Basic Authentication header. The username must be the ‘username’ attribute for a valid account (that is an account that is active and not-expired), and the password for the account must be sent in clear-text.

Example

```
Request:
POST /api/v1/example.org/account/12345/api-keys/create HTTP/1.1
Authorization: Basic c3VwZXI6YWJjMTIz

Response (success):
HTTP/1.1 201 Created
Content-Type: application/vnd.eduserv.iam.apiKey-v1+json; charset=UTF-8
```

Authentication errors

Failed authentication will result in an HTTP 401 status from the API. Additional failure information is returned in the response body. This is encoded as an application/vnd.eduserv.iam.authenticationError-v1+json object.

application/vnd.eduserv.iam.authenticationError-v1+json object

This object is used to encode information about a failed authentication attempt.

Object field	Description

reason	A code indicating the reason for the authentication failure. This may be one of • 'badCredentials' – The supplied credentials were invalid. This means the username or password were invalid (for basic authentication) or the API key was invalid (for API key authentication). • 'accountExpired' – The account to which the credentials apply has expired. • 'invalidIP' – The supplied credentials cannot be used from the IP address that the client is connecting from. This applies to administration and access accounts only.
message	A human-readable message describing the failure. This may be used on a UI to provide a reason for the failure to the user.

API Key Authentication

API keys are temporary or long-lived authentication tokens that are associated with an OpenAthens account. They have exactly the same permissions as the account to which they are associated. They avoid the need for the account password to be sent on every API request.

Although HTTP Basic authentication can be used to authenticate to any API URL, the normal method to use the API is as follows:

1. Obtain a temporary API key from the API, using HTTP Basic authentication.
2. Add the API key obtained in step 1 to subsequent API calls.
3. Upon expiry of the API key, either renew as in step 1 or discard.

Obtaining a temporary API key

Temporary keys are time-limited and may only be used to authenticate to the API for a fixed period. A client application may store the key in a user session, or via a cookie.

Prerequisites

To obtain a temporary API key, a client application must

- have access to a username and password associated with an account in an OpenAthens domain.
- have a link to an account object from which to obtain an API key.

Procedure

To obtain a temporary key, a client should perform a POST request to the API key resource for the account to which the key should be associated.

> /api/v1/example.org/account/<id>/api-keys/create

The response returns an `application/vnd.eduserv.iam.apiKey-v1+json` object containing the API key.

Example

```
Request:
POST /api/v1/example.org/account/12345/api-keys/create HTTP/1.1
Authorization: Basic c3VwZXI6YWJjMTIz
Response:
HTTP/1.1 201 Created
Content-Type: application/vnd.eduserv.iam.apiKey-v1+json; charset=UTF-8


{
  "key" : "ed7efc59-7fe2-4e0c-b6f4-50439fcdb49a",
  "type" : "temporary",
  "expires" : "2012-11-23T14:43:34Z"
}
```

application/vnd.eduserv.iam.apiKey-v1+json object

Object field	Description
key	The API key.
type	The type of key. This may be one of: • temporary • assigned

expires	The expiry date/time for the key.

Using a temporary API key

Prerequisites

To use a temporary API key a client must

- have previously obtained an API key using the procedure detailed above.

Procedure

The API key should be passed in subsequent API requests as an HTTP Authorization header with ‘OAApiKey’ as the scheme, and the key as the credential.

Example

```
Request:
GET /api/v1/example.orp/account/12345 HTTP/1.1
Authorization: OAApiKey ed7efc59-7fe2-4e0c-b6f4-50439fcdb49a
Response:
HTTP/1.1 200 OK
Content-Type: application/vnd.eduserv.iam.account-v1+json; charset=UTF-8
```

Long-lived API keys

Long-lived API keys behave in the same way as temporary keys described earlier however they have an expiry date measured in years rather than minutes. Long-lived keys are intended for *applications* to use to authenticate against the API. This decouples an application from the account credentials and means if the administrator changes their password, the application won’t need to be updated to send the new value.

To obtain a long-lived key, you should log into the OpenAthens administration site (https://admin.openathens.net) as the administrator you wish to request a key for. Under ‘Management’, select ‘API keys’ and create a new key. See: API keys

Long-lived API keys are used in the same manner as temporary API keys (using the same HTTP Authorisation header and scheme).

Only organisation accounts may have long-lived API keys associated with them.

Authorisation

Regardless of whether an API key or HTTP Basic Authentication was used to authenticate to the API, all operations will only be performed with the privileges of the authenticated user. This means that when using an API key, the same permissions will be applied as those of the administration account from which it was originally requested. Therefore the key of an organisation will only be able to administer accounts belonging to that organisation, or sub-organisations. Authenticating with an end-user (non-administrator) account will only provide very restricted access to API functions.

Any request that is not authorised will fail with an HTTP 403 status code.

See also:

- API overview
- Authenticating to the API
- API entry-point
- Fetching attribute schemas via the API
- Fetching organisations via the API
- Fetching Groups via the API
- Account management via the API
- API bulk operations
- Fetching available service providers via the API
- Generating authentication tokens for end-users via the API
- API usage examples