API overview

All API operations follow a RESTful design pattern. This means that the key entities are uniquely addressable resources that provide encoded representations of entities. Where data is related, hypermedia links are used to indicate relationships between entities. Performing HTTP requests on an appropriate resource URI carries out operations such as creating, reading or deleting an account.

The API currently implements only a subset of the functionality provided by the OpenAthens account management user-interface (https://admin.openathens.net). The functionality exposed via the public API will be expanded over time. The API described in this document is separate from the SAML APIs used to provide Single Sign-On (SSO) between the Authentication Point and Service Providers.

Transport security

The transport protocol for all API requests is HTTPS (HTTP 1.1 + TLS 1.2). Connections using TLS versions before 1.2 will be rejected.

Encoding

Most API operations encode data as simple objects, arrays or strings. In cases where more complex objects are used, these are detailed throughout this guide.

Object encoding

Data objects are represented as JSON objects, with a media type of `application/*+json`. These objects have a vendor-specific prefix of `vnd.eduserv.iam.*` that designates additional type information. These specific types are referred to throughout this document.

Clients of the API must send a suitable Content-type header when sending encoded data to the API. They should provide an Accept header when reading data from a resource, though if this is omitted, the API will default to sending a JSON representation in most cases.

Where API calls do not return JSON objects, the correct media type is always returned in the HTTP response Content-Type header. This document will detail where alternative media types area available, or where multiple representations are available for a given resource.

Date encoding

Fields that carry dates are encoded as ISO 8601 date/time strings, normalised to UTC. This encoding should also be used by API clients when passing date values in querystring parameters.

This is expressed as a Java DateFormat string of: `yyyy-MM-dd'T'HH:mm:ss'Z'`

Example

```
2012-10-22T16:48:54Z
```

Boolean encoding

Boolean values are encoded according to JSON encoding rules (true or false). This encoding should also be used by API clients when passing boolean flags in querystring parameters.

Errors

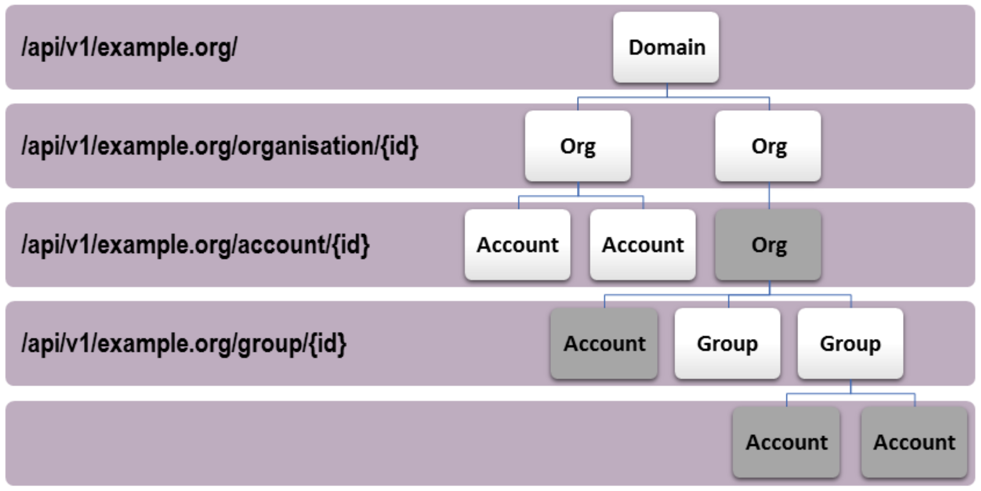
All errors are returned with an appropriate HTTP status code.

Some requests return extended error information, encoded as JSON objects. For example:

```
{
  "code" : "BadRequest",
  "message" : "An invalid request has been made."
}
```

API Objects

The key objects in the API have relationship as illustrated below. Each of the entities shown in the diagram are addressable via URIs, shown on the left.


/api/v1/example.org/
/api/v1/example.org/organisation/{id}
/api/v1/example.org/account/{id}
/api/v1/example.org/group/{id}
Domain
Org
Org
Account
Account
Org
Account
Group
Group
Account
Account

Domains

A domain is the highest-level grouping of API objects. A domain spans all organisations and their constituent accounts. Typically, a domain reflects an individual customer of the OpenAthens system. Domains exist to provide a namespace for that customer. The domain name forms a segment of the URL path for *every* API request to resources in that domain. Domains also provide a container for configuration of 'global' settings for all sub-organisations.

Organisations

Organisations provide a means to group accounts into a hierarchical structure. This structure is usually roughly based on the structure of the real-world organisation using OpenAthens, but may be any structure appropriate to the usage of the system. It provides a means to delegate administration for a sub-set of accounts. A domain will consist of at least one 'root' organisation. Sub-organisations are then created in a hierarchy below this root organisation.

A holder of an administration account administers an organisation. Although the API conceptually supports *multiple* administration accounts administering a single organisation, currently this is limited to one account (for legacy reasons).

Accounts

Accounts refer to accounts in the OpenAthens system that may belong to end-users or administrators. An account will always exist directly under exactly **o ne** organisation.

User accounts are used to access OpenAthens-protected content. Administrator accounts are used to administer organisations and access the user administration application (they may also be used to access the API).

Groups

Groups may contain zero or more accounts. Groups belong to, and are created by organisations. Although the API conceptually supports accounts being members of multiple groups, currently this behaviour is limited (for legacy reasons), and an individual account may only be a member of either zero or one group.

Object IDs

All API objects have an 'id' field. The ID always carries a globally unique (within the scope of the object type) identifier for the object it represents. The ID forms part of the resource URL used to address the object.

It is **not** recommended that clients of the API manually construct URLs by inferring their structure and inserting/replacing the values of IDs. Instead, objects should be addressed by following the relevant hypermedia links from related objects (as designated by the 'href' field on a link) – see section 3.6 for more information on links.

Object links and relationships

All API objects have zero or more 'link' fields. Links provides a means to identify the relationship of an object to other objects in the API and in some cases identify the types of operations that are available on that object.

Links have the following fields:

href	A URL reference to the linked object
rel	The relation of the linked object to the object carrying the link.

type	The type of the object referenced by the link (**optional**).
method	The HTTP verb describing the method to use when following the link (**optional**).
name	The name of the linked object (**optional**).

Object relationships

The ‘rel’ field on a link may be a number of values depending on the nature of the relationship. These values are defined as follows.

General-purpose relationships

These relations define general relationships between objects. Most objects will carry links with one or more of these relationships.

self	A reference to the current object.
down	The linked object is a child of the carrying object.
up	The linked object is a parent of the carrying object.
add	Perform an add operation to create a new object as a child of the current object.
update	Perform an update operation to update the current object.
delete	Delete the current object.

Object-specific relationships

These relations define relationships that are specific to objects of particular types.

organisation:root	Defines a link to the root organisation. The root organisation has no parent organisations.
organisation:query	Defines a link to a resource to query organisations.
account:query	Defines a link to a resource to query accounts.

Example

```
"links": [
           {
             "type" : "application/vnd.eduserv.iam.account-v1+json",
             "rel" : "self",
             "href" : "/api/v1/example.org/account/12345",
             "method" : "get"
           },
           {
             "rel" : "delete",
             "href" : "/api/v1/example.org/account/12345",
             "method" : "delete"
           }
]
```

See next:

- 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