Account management via the API

Account management operations include creating accounts, modifying existing accounts and deleting accounts. As outlined in section 3.4, accounts belong to an organisation. Therefore many account operations are applied to an organisation object.

Creating new accounts

Prerequisites

To create new accounts, a client application must

- authenticate to the API as described in Authenticating to the API.
- obtain an organisation object for the organisation under which the account should be created, as described in Fetching organisations via the API.

Procedure

To create a new account underneath a given organisation, perform an HTTP POST request to the by following the link with a relation of ‘add’ from an organisation, sending a request body with an `application/vnd.eduserv.iam.admin.accountRequest-v1+json` object.

> /api/v1/example.org/organisation/<id>/accounts/create/<type>

Where id is the organisation ID and type is the type of account being created, usually 'personal'
(personal, organisation_administrator, user_administrator, self_registration, access).

Request querystring

The following querystring parameters affect the behaviour of the operation.

Querystring Parameter	Description
sendEmail	A Boolean indicating whether to send an email to the user. The email sent will be an activation email if the account status is set to ‘pending’ or a new account email containing the account credentials if the status is set to ‘Active’ and a password is supplied.
defaultPermissions	A Boolean indicating whether to assign all the default permission sets to this account upon creation. This is mutually exclusive with the ‘permissionSets’ field in the account request object. If ‘permissionSets’ is set in the request, this flag must be omitted.

Response payload

The response payload is the `application/vnd.eduserv.iam.account-v1+json` object that was created. This includes the allocated object ID, activation code (if status was set to ‘pending’) and permission sets (if applicable).

The ‘Location’ header contains a reference to the newly created account object.

Response codes

HTTP Response Code	Description
201	The account was created.
400	The request was invalid.

application/vnd.eduserv.iam.admin.accountRequest-v1+json object

Account request objects encode information about account creation and modification requests. They contain a sub-set of the fields of an account object (because some fields on accounts are read-only or not applicable at the time an account is created). When creating new accounts, requests must contain at least the minimum fields required, namely

- status
- password (if status is set to active)
- all mandatory attributes, as defined in the attribute schema for the type of account being created
- the account expiry date

When modifying existing accounts, only the fields that the client wishes to modify need be sent (see the account management section for further details).

Object field	Description
status	The activation status of the account. May be one of: • active • pending
activationCodeExpiry	The expiry for activation codes generated as part of this request (only applies if status = Pending).
groups	An array of group names of which the account is a member. Currently only one group can be specified.
permissionSets	A list of permission set names for this account.
expiry	The expiry date for this account. Expiry dates cannot be set more than 5 years in the future.
username	The username for the account. This only applies on account creation.
attributes	An object that maps attribute names to values (object key = field name, value = message). The available attribute names are available by querying the account schema.
password	The account password in clear-text. Passwords are stored in OpenAthens in one-way hashed form. Hence, they are not subsequently retrievable via an account object.
ipRanges	An array of IP address ranges from which the account will be restricted. Only applies when creating administration or access accounts.
organisationMove	This is used to move the account to a different parent organisation (see section 9.5). This only applies on account modification.

Account status

The status of the account designates whether it is ‘active’, or ‘pending activation’. Accounts marked as active have had a password assigned to them, either by an administrator, or as part of the account activation process. Active accounts may be used to access OpenAthens-protected resources until the account expiry date is reached. Pending accounts cannot login to resources or the API until a password is assigned to the account and the account status is changed to ‘active’.

The following table summarises the valid values for the account status.

Account status	Description
pending	The account requires activation by the user. If an account is created or changed to this status, an activation code will be automatically generated for the account. If the ‘sendEmail’ querystring parameter is set, an activation email will be sent to the user. An account cannot authenticate if it is in this state.
active	The account is active and may be used to authenticate by the user. If an account is created with this status and the ‘sendEmail’ querystring parameter is set, a new account email, containing the password, will be sent to the user.

Error handling

If an invalid account creation or modification request is sent an HTTP status of 400 will be returned. The response body will contain an `application/vnd.eduserv.iam.admin.accountError-v1+json` object detailing the reasons why the request failed.

application/vnd.eduserv.iam.admin.accountError-v1+json object

Object field	Description
message	A description of the error.
invalidFields	An object mapping request field to error messages (object key == field name, value == message).
invalidAttributes	An object mapping user attributes to error messages (object key == attribute name, value == message).

8.1.5. Example

```
Request:
POST /api/v1/example.org/organisation/12345/accounts/create/personal?sendEmail=true&defaultPermissions=true HTTP
/1.1
Content-Type: application/vnd.eduserv.iam.admin.accountRequest-v1+json
Authorization: OAApiKey <api-key>
```

```
{
    "expiry": "2017-01-01T00:00:00Z",
    "status": "pending",
    "username": "expuser01",
    "attributes": {
        "forenames": "first",
        "surname": "last",
        "emailAddress": "first.last@example.com"
    },
    "groups" : [
                   "group1",
                   "group2"
    ]
}
Response (success):
HTTP/1.1 201 Created
Location: /api/v1/example.org/account/1012345
Content-Type: application/vnd.eduserv.iam.account-v1+json


{
    "id" : "1012345",
    "status" : "Pending",
    "activationCode" : {
      "code": "ABC123DEF",
      "expires": "2017-01-01T00:00:00Z",
    }
    "attributes" : {
                   "username" : "expuser01",
                   "forenames": "first",
                   "surname": "last",
                   "emailAddress" : "first.last@example.org"
              },
    "groups" : ["group1"],
    "permissionSets" : [
                { "id": "5678",
                  "name": "exp#default"
                  "href": "/api/v1/example.org/permissionSet/5678"
                }
    ],
    "organisation" : { "id" : "1234",
                  "href" : "/api/v1/example.org/organisation/1234"
                 },
    "links" : [
                { "rel" : "self",
                  "type" : "application/vnd.eduserv.iam.account-v1+json",
                  "href" : "/api/v1/example.org/account/1012345",
                  "method" : "get",
                },
                { "rel" : "parent",
                  "type" : "application/vnd.eduserv.iam.admin.organisation-v1+json",
                  "href" : "/api/v1/example.org/organisation/1234",
                  "method" : "get"
                },
                { "rel" : "delete",
                  "href" : "/api/v1/example.org/account/1012345",
                  "method" : "delete",
                },
                { "rel" : "update",
                  "type" : "application/vnd.eduserv.iam.admin.accountRequest-v1+json",
                  "href" : "/api/v1/example.org/account/1012345/modify",
                  "method" : "post",
                },
                { "rel" : "child",
                  "type" : " application/vnd.eduserv.iam.serviceList-v1+json",
                  "href" : "/api/v1/example.org/account/1012345/services",
                  "method" : "get",
                }
    ]
```

```
}
Response (error):
HTTP/1.1 400 Bad Request
Content-Type: application/vnd.eduserv.iam.admin.accountError-v1+json


{
    "message" : "One or more supplied fields are invalid",
    "invalidFields" : {
                "username" : "This username is already in use"
    }
    "invalidAttributes" : {
                "emailAddress" : "This is not a valid email address"
            }
}
```

Fetching individual accounts

Account objects contain full information about an account, including the account status, attributes, expiry, group membership and permissions.

Prerequisites

To fetch an account object, a client application must

- authenticate to the API as described in Authenticating to the API.
- query for an account or follow an account link from another API object, or as the result of creating a new account.

Procedure

Individual accounts may be fetched by following the link with a relation of 'self' from an account object. Alternatively, accounts may be queried by following the 'account:query' link from the API entry point.

To fetch metadata about an organisation to which an account is a member follow the 'up' link from an account.

To fetch an individual account perform a GET request to:

> /api/v1/example.org/account/<id>

Where id is the value of the id field for the account object.

To search for an individual account based on username perform a GET request to:

> /api/v1/example.org/account/query?username=expuser01

To search for an individual account based on email address.

> /api/v1/example.org/account/query?email=alex@example.org

To search for an individual account based on persistent user identifier.

> /api/v1/example.org/account/query?persistentUID=xxxxxxxx:xxxxxxx

> It is only possible to search for accounts based on email address if the email address is *unique* for that account - that is the 'uniqueEmailAddress' attribute it set.
>
> These search methods also enable applications to check whether a given email address or username has been used. This may be used by a registration application to warn the user that the email address is not available before they have submitted the application form.
>
> If a user attempts to retrieve their own details then a 204 response code is returned with no content for personal, self-registration and access accounts when using basic authentication.

Response payload

The response payload is an `application/vnd.eduserv.iam.account-v1+json` object.

Response codes

HTTP Response Code	Description
200	The request was successful.

204	The request was successful but returned no content
404	Account does not exist with the given id.
403	Administrator does not have permission to view this account.

Example

```
Request:
GET /api/v1/example.org/account/1234567 HTTP/1.1
Authorization: OAApiKey <api-key>
Response:
HTTP/1.1 200 OK
Content-Type: application/vnd.eduserv.iam.account-v1+json


{
    "id" : "1234a67",
    "status" : "Active",
    "organisation" : {  "id" : "676573453",
                        "defer" : { "id": "6585674434353",
                                    "date" : "2012-11-22T13:24.345Z"
                                     }
    },
    "attributes" : {
      "username" : "expjohn",
      "emailAddress" : "john@example.org"
    },
    "memberOf": [ {
                   "href" : " /api/v1/example.org/group/234567",
                   "name" : "group1"
    } ],
    "links": {
     ... detail omitted
    },
}
```

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

This type represents account objects, together with metadata about their group membership, and organisation affiliation.

Object field	Description
id	The unique account identifier.
status	The activation status of the account. May be one of • active • pending
type	The type of account. May be one of • personal • organisation_administrator • user_administrator • self_registration • access
activationCode	The activation code for the account (if status = ‘pending’). This is an object with code and expires properties. These carry the code, and the associated expiry date, respectively.
created	The date on which the account was created.
modified	The date on which the account was last modified.
expiry	The date on which the account expires.
organisation	The identifier of the parent organization of the account (see the following section).

memberOf	An array of groups to which the account is a member.
permissionSets	A list of permission sets for this account.
attributes	An object that maps attribute names to values (object key = field name, value = message).
administersOrganisations	An array of organisations that this account administers (applies to administration accounts only).

Organisation field

The 'organisation' field in the response contains an object with the following parameters.

Object field	Description
id	The id of the new organisation (**required**).
name	The display name of the organisation.
defer	An object containing id and date parameters which specify a deferred move (see section below).

Modifying individual accounts

Prerequisites

To modify an account object, a client application must

- authenticate to the API as described in Authenticating to the API.
- query for an account or follow an account link from another API object, or as the result of creating a new account.

Procedure

To modify an individual account, follow the link with a relation of 'update' from an account object and perform a POST request with an application/vnd.eduserv.iam.admin.accountRequest-v1+json object in the request body. It is only necessary to include the fields that client wishes to modify. This means that

- if the 'organisation' field is omitted, the accounts organisation will be left unchanged (for more information on modifying an account's organisation see section 9.5).
- if the 'memberOf' field is omitted the accounts group membership will be left unchanged.
- if the 'permissionSets' field is omitted the accounts permission set assignments will be left unchanged.
- if the 'attributes' field is omitted the account attributes will be left unchanged. If one or more fields within the attributes are omitted then these will be left unchanged. It is only necessary to include attributes that wish to be changed.

Perform the POST request to:

> /api/v1/example.org/account/<id>/modify

Where id is the value of the id field for the account object.

Request querystring

The following querystring parameters affect the behaviour of the operation.

Querystring Parameter	Description
sendEmail	A Boolean value indicating whether to send an email to the user. The email sent will be an activation email if the account status is set to 'Pending' or a modified account email containing the account credentials if the status is set to 'Active' and a password is supplied.
defaultPermissions	A Boolean value indicating whether to assign all the default permission sets to this account.

Request payload

This should be an `application/vnd.eduserv.iam.admin.accountRequest-v1+json` object.

Response codes

HTTP Response Code	Description
200	The request was successful.

404	An account does not exist with the given id.
400	The request was invalid. The response should be an error indicating the reason.

Example

```
Request:
POST /api/v1/example.org/account/1234567/modify HTTP/1.1
Authorization: OAApiKey <api-key>
Content-Type: application/vnd.eduserv.iam.admin.accountRequest-v1+json


{
  "attributes" : {
    "forenames" : "john",
    "emailAddress" : "john@example.org"
  }
}
Response:
HTTP/1.1 200 OK
Content-Type: application/vnd.eduserv.iam.account-v1+json


... detail omitted
```

Deleting accounts

Deleting accounts removes the account from the organisation. This is an irreversible operation – once deleted, accounts cannot be recovered and they cannot be used to sign in.

Prerequisites

To delete an account object, a client application must

- authenticate to the API as described in Authentication to the API.
- query for an account or follow an account link from another API object, or as the result of creating a new account.

Procedure

To delete an individual account, follow the link with a relation of ‘delete’ from an account object, performing a DELETE request to

/api/v1/example.org/account/<id>

Where id is the value of the id field for the account object.

Example

```
Request:
DELETE /api/v1/example.org/account/12345 HTTP/1.1
Authorization: OAApiKey <api-key>
Response (success):
HTTP/1.1 204 No Content
```

Updating an organisation for an account

Updating an organisation for an account moves the account to a different parent organisation in the organisational hierarchy outlined in section PLACEHOLDER. Updating an organisation for an account follows the same method as modifying an account, detailed in section 9.3.

Prerequisites

To modify an account object, a client application must

- authenticate to the API as described in Authenticating to the API.
- query for an account or follow an account link from another API object, or as the result of creating a new account.
- obtain an organisation object for the desired destination organisation, as described in section 7.1.

Procedure

To update the organisation (move an account to a different organisation) for an account, perform an update request on the account (as shown in section 9.3), setting a value for the ‘organisation’ field in the `application/vnd.eduserv.iam.admin.accountRequest-v1+json` object.

Note that it is only possible to move accounts to/from sub-organisations of the organisation administered by the account that is authenticated to the API.

Perform a POST request to:

> /api/v1/example.org/account/<id>/modify

Where id is the value of the id field for the account object.

Request object

The ‘organisation’ field in the request must be an object with the following parameters.

Object parameter	Description
id	The id of the new organisation (**required**).

Example

```
Request:
POST /api/v1/example.org/account/12345/modify HTTP/1.1
Authorization: OAApiKey <api-key>

Content-Type: application/vnd.eduserv.iam.admin.accountRequest-v1+json

{
  "organisation": {
    "id" : "67575664534"
  }
}

Response:
HTTP/1.1 204 OK
Content-Type: application/vnd.eduserv.iam.account-v1+json
```

Password Reset

Send the user a standard password reset email. This allows the user to follow a link via the OpenAthens Authentication Point (AP) to change their password.

Prerequisites

To reset an account password, a client application must

- authenticate to the API as described in Authenticating to the API.
- Provide PUID of user, this can be gained from an account query, or as the result of creating a new account.

Procedure

Perform a GET request against the following URL, supplying the PUID for the account to reset.

> /api/v1/example.org/account/passwordreset/<PUID>

Where PUID is the persistent user identifier for the account.

Response codes

Response code	Description
204	The request was successful the user was emailed
400	The request is invalid, most commonly the user identifier is an invalid format
404	The username and or email cannot be matched on AP.

Example

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
Request:
GET /api/v1/example.org/account/passwordreset/abcd1234:456789a HTTP/1.1
Authorization: OAApiKey <api-key>

Response:
HTTP/1.1 204 No Content
Content-Type: application/vnd.eduserv.iam.account-v1+json
```

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