
eContext Auth API Documentation

Release 0.0.1

Jonathan Spalink

Apr 16, 2021

Contents

1	Introduction	1
1.1	Input Format	1
1.2	Output Format	1
2	Concepts	3
2.1	Applications	3
2.2	Groups	3
2.3	Organizations	3
2.4	Users	3
2.5	“Custom Data”	4
3	Authentication	5
3.1	POST /authenticate	5
4	Applications	7
4.1	GET /applications	7
4.2	POST /applications/application	8
4.3	GET /applications/application/{appid}	10
4.4	PUT /applications/application/{appid}	11
4.5	DELETE /applications/application/{appid}	12
4.6	GET /applications/application/{appid}/users	13
5	Groups	15
5.1	GET /groups	15
5.2	POST /groups/group	17
5.3	GET /groups/group/{groupid}	18
5.4	PUT /groups/group/{groupid}	20
5.5	DELETE /groups/group/{groupid}	21
5.6	GET /groups/group/{groupid}/users	22
6	Users	25
6.1	GET /users	25
6.2	GET /users/search/{search}	27
6.3	POST /users/user	28
6.4	GET /users/user/{userid}	30
6.5	PUT /users/user/{userid}	32
6.6	POST /users/user/{userid}/apikey	34

6.7	GET /users/user/{userid}/apikey/{apikeyid}	35
6.8	PUT /users/user/{userid}/apikey/{apikeyid}	36
6.9	DELETE /users/user/{userid}/apikey/{apikeyid}	37
6.10	DELETE /users/user/{userid}/application/{appid}	38
6.11	DELETE /users/user/{userid}/group/{groupid}	39
6.12	DELETE /users/user/{userid}	40
7	Changelog	43
7.1	0.0.1 – 2017-04-20	43
8	Notes	45
9	Indices and tables	47
	Index	49

The eContext Authentication API allows provides an application and user management platform allowing and allows authentication and (eventually) authorization controls that can be used to control access to various resources.

1.1 Input Format

The input format to the eContext Authentication API is specified using the HTTP Content-Type header ([RFC 2616#section-14.17](#)). A Content-Type of `application/json` is preferred, and there is no guarantee that other Content-Types will be honored.

1.2 Output Format

The output format is generally set by using the HTTP Accept header ([RFC 2616#section-14.1](#)). The default output format for the eContext API is JSON (`application/json`). As the default output format, all examples in this documentation are displayed using JSON.

The following output formats are currently supported:

- `application/json` ([RFC 4627](#))

2.1 Applications

An Application is a concept which is used as the base resource against which Groups and Users are checked against. No Group or User may exist without a direct connection to an Application. An Application typically is comprised of an id, a name, and a description but may also contain Custom Data. When a User authenticates against the API the credentials are checked against the specified Application. Even if a username and password happen to be correct, if a User is not associated with the application specified in the authentication request, the request will fail.

The base Application inside the API is the “eContext Auth” Application which must be used in order to access the API.

2.2 Groups

A Group provides associations and Custom Data inside an Application. A particular group may belong to a single Application and may be associated with many Users.

2.3 Organizations

An Organization allows a structural hierarchy for storing Users. A User may not exist alone outside of an organization. An Organization may also contain custom-data that will then be associated with all of its users.

2.4 Users

A User is an account which may be used to authenticate against the API. A user may be associated with many Applications and many Groups but only with a single Organization. An authentication call against the API must include the specific Application which a client is seeking access to.

2.5 “Custom Data”

Additional data may be associated with Applications, Groups, and Users that may be used downstream by Applications. For example, an Application may look for certain attributes to be passed in with a User to override defaults, and those can be specified in the “Data” or “Custom Data” attributes of a User.

The main purpose of the eContext Auth API is to allow a user to authenticate against an Application defined in the system.

3.1 POST /authenticate

Authenticate a user against an application resource.

3.1.1 Resource URL

<https://auth.econtext.ai/api/authenticate>

3.1.2 Parameters

Parameter	Type	Description
<code>type</code>	string	What type of authentication to perform. Currently it must be either <code>username</code> or <code>apikey</code>
<code>application</code>	string	The id of the application to authenticate to
<code>credential</code>	object	The username and password to authenticate with
<code>credential.username</code>	string	The username to authenticate with (or the <code>apikey.id</code>)
<code>credential.password</code>	string	The password to authenticate with (or the <code>apikey.secret</code>)

3.1.3 Return

The `authenticate` call typically returns a `true` or `false` value to indicate whether the credentials are valid.

3.1.4 Example Request

POST Request

```
curl -X POST -u username:password \
  -H 'content-type: application/json' \
  -d '{
    "type": "username",
    "credential": {
      "username": USERNAME,
      "password": PASSWORD
    },
    "application": APP_ID
  }'
https://auth.econtext.ai/api/authenticate
```

POST Response

```
{
  "econtext": {
    "result": {
      "authenticated": true
    },
    "elapsed": 0.0038661956787109375
  }
}
```

An application is a resource that can be authenticated against. A good example is the eContext Auth Application which is the internal application that is used in order to connect to the eContext Auth API itself. Users who are associated with a particular application may authenticate against that application.

4.1 GET /applications

Retrieve a list of applications available in the system

- *Resource URL*
- *Return*
- *Example Request*
 - *GET Request*
 - *GET Response*

4.1.1 Resource URL

<https://auth.econtext.ai/api/applications>

4.1.2 Return

A list of applications that exist in the eContext Auth API.

4.1.3 Example Request

GET Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/applications
```

GET Response

```
{
  "econtext": {
    "result": {
      "applications": [
        {
          "status": "ENABLED",
          "href": "https://auth.econtext.ai/api/applications/application/
a136fbd-26cc-448f-af0a-cf98fe165cc6",
          "name": "eContext Auth",
          "custom_data": null,
          "created_at": "",
          "modified_at": "2017-04-14 16:38:05.025000+00:00",
          "id": "a136fbd-26cc-448f-af0a-cf98fe165cc6",
          "description": "eContext Authentication/Authorization Application"
        }
      ]
    },
    "elapsed": 0.0014569759368896484
  }
}
```

4.2 POST /applications/application

Create a new application object

- *Resource URL*
- *Parameters*
- *Return*
- *Example Request*
 - *POST Request*
 - *POST Response*

4.2.1 Resource URL

<https://auth.econtext.ai/api/applications/application>

4.2.2 Parameters

Parameter	Type	Description
name	string	A name for the Application
description	string	A description for the Application
custom_data	object	A JSON object containing arbitrary data

4.2.3 Return

A newly created Application Object

4.2.4 Example Request

POST Request

```
curl -X POST -u username:password \
  -H 'content-type: application/json' \
  -d '{
    "name": "Test Application",
    "description": "Test application for use in this round of tests",
    "custom_data": {
      "spam": "and eggs"
    }
  }' \
  https://auth.econtext.ai/api/applications/application
```

POST Response

```
{
  "econtext": {
    "result": {
      "application": {
        "status": "ENABLED",
        "href": "https://auth.econtext.ai/api/applications/application/bf59fd16-d003-45ae-889d-4ed06b804c21",
        "name": "Test Application",
        "custom_data": {
          "spam": "and eggs"
        },
        "created_at": "2017-04-19 19:51:14.409000+00:00",
        "modified_at": "2017-04-19 19:51:14.409000+00:00",
        "id": "bf59fd16-d003-45ae-889d-4ed06b804c21",
        "description": "Test application for use in this round of tests"
      },
      "elapsed": 0.03231406211853027
    }
  }
}
```

4.3 GET /applications/application/{appid}

Retrieve an existing application object identified by {appid}

- *Resource URL*
- *Return*
- *Example Request*
 - *GET Request*
 - *GET Response*

4.3.1 Resource URL

`https://auth.econtext.ai/api/applications/application/{appid}`

4.3.2 Return

An Application object identified by {appid}

4.3.3 Example Request

GET Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/applications/application/bf59fd16-d003-45ae-889d-4ed06b804c21
```

GET Response

```
{
  "econtext": {
    "result": {
      "application": {
        "status": "ENABLED",
        "href": "https://auth.econtext.ai/api/applications/application/bf59fd16-d003-45ae-889d-4ed06b804c21",
        "name": "Test Application",
        "custom_data": {
          "green eggs": "and ham"
        },
        "created_at": "2017-04-19 19:51:14.409000+00:00",
        "modified_at": "2017-04-19 19:52:16.618000+00:00",
        "id": "bf59fd16-d003-45ae-889d-4ed06b804c21",
        "description": "Test application for use in this round of tests"
      }
    },
    "elapsed": 0.01231406211853027
  }
}
```

```
}
}
```

4.4 PUT /applications/application/{appid}

Update an existing application object identified by {appid}

- *Resource URL*
- *Parameters*
- *Return*
- *Example Request*
 - *PUT Request*
 - *PUT Response*

4.4.1 Resource URL

`https://auth.econtext.ai/api/applications/application/{appid}`

4.4.2 Parameters

Parameter	Type	Description
name	string	A name for the Application
description	string	A description for the Application
custom_data	object	A JSON object containing arbitrary data
status	string	The status of the Application. Available options are ENABLED and DISABLED

4.4.3 Return

An updated Application Object

4.4.4 Example Request

PUT Request

```
curl -X PUT -u username:password \
  -H 'content-type: application/json' \
  -d '{
    "custom_data": {
      "green eggs": "and ham"
    }
  }' \
  https://auth.econtext.ai/api/applications/application/bf59fd16-d003-45ae-889d-4ed06b804c
```

PUT Response

```
{
  "econtext": {
    "result": {
      "application": {
        "status": "ENABLED",
        "href": "https://auth.econtext.ai/api/applications/application/
bf59fd16-d003-45ae-889d-4ed06b804c21",
        "name": "Test Application",
        "custom_data": {
          "green eggs": "and ham"
        },
        "created_at": "2017-04-19 19:51:14.409000+00:00",
        "modified_at": "2017-04-19 19:52:16.618000+00:00",
        "id": "bf59fd16-d003-45ae-889d-4ed06b804c21",
        "description": "Test application for use in this round of tests"
      }
    },
    "elapsed": 0.01231406211853027
  }
}
```

4.5 DELETE /applications/application/{appid}

Delete the Application identified by {appid}. An Application may not be deleted while there are users associated with it. You must first delete or remove the application association from any users before you will be able to delete the Application. This prevents orphaned User objects from being allowed to exist in the system. Additionally, the Application's status must be set to DISABLED.

- *Resource URL*
- *Return*
- *Example Request*
 - *DELETE Request*
 - *DELETE Response*

4.5.1 Resource URL

`https://auth.econtext.ai/api/applications/application/{appid}`

4.5.2 Return

A boolean response explaining whether the DELETE call was successful or not.

4.5.3 Example Request

DELETE Request

```
curl -X DELETE -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/applications/application/bf59fd16-d003-45ae-889d-4ed06b804c
```

DELETE Response

```
{
  "econtext": {
    "result": {
      "deleted": true
    },
    "elapsed": 0.0257871150970459
  }
}
```

4.6 GET /applications/application/{appid}/users

Retrieve a list of users currently associated with the Application identified by {appid}

- *Resource URL*
- *Return*
- *Example Request*
 - *GET Request*
 - *GET Response*

4.6.1 Resource URL

```
https://auth.econtext.ai/api/applications/application/{appid}/users
```

4.6.2 Return

A list of Users associated with the Application.

4.6.3 Example Request

GET Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/applications/application/bf59fd16-d003-45ae-889d-4ed06b804c
users
```

GET Response

```
{
  "econtext": {
    "result": {
      "users": [
        {
          "username": "test-user@econtext.ai",
          "status": "UNVERIFIED",
          "applications": [
            "bf59fd16-d003-45ae-889d-4ed06b804c21"
          ],
          "href": "https://auth.econtext.ai/api/users/user/
a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
          "groups": [],
          "apikeys": [],
          "id": "a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
          "name": "Test User",
          "created_at": "2017-04-20 15:55:08.339000+00:00",
          "modified_at": "2017-04-20 15:55:08.377000+00:00",
          "custom_data": null,
          "email": "test-user@econtext.ai"
        }
      ]
    },
    "elapsed": 0.0032689571380615234
  }
}
```

A group is a resource that can be used to store various information about a group of users. It can provide access to common data in the `custom_data` field and can be used inside a client application to restrict access to resources based on membership. For example, in the eContext API, users in the “admin” group have access to more endpoints and have encryption of eContext Category IDs turned off via the “_no_encrypt” flag found in the `custom_data` field of the “admin” group.

A group may only belong to a single Application and must have a unique name inside that Application. For example, there may be multiple “admin” groups so long as they belong to different Application objects. In a single Application, there may only be a single group named “admin”

5.1 GET /groups

Retrieve a list of groups available in the system

- *Resource URL*
- *Return*
- *Example Request*
 - *GET Request*
 - *GET Response*

5.1.1 Resource URL

<https://auth.econtext.ai/api/groups>

5.1.2 Return

A list of groups that exist in the eContext Auth API.

5.1.3 Example Request

GET Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/groups
```

GET Response

```
{
  "econtext": {
    "result": {
      "groups": [
        {
          "status": "ENABLED",
          "application": "6CyMm9ikHh5ASSFOZ9OUWo",
          "href": "https://auth.econtext.ai/api/groups/group/4QlXPskVltis43vjTNwF7n",
          "name": "api",
          "custom_data": {
            "classify_limit": 10,
            "tier_depth": 9
          },
          "created_at": "2017-04-12 21:46:02.264000+00:00",
          "modified_at": "2017-04-12 21:46:02.274000+00:00",
          "id": "4QlXPskVltis43vjTNwF7n",
          "description": "API user"
        },
        {
          "status": "ENABLED",
          "application": "6CyMm9ikHh5ASSFOZ9OUWo",
          "href": "https://auth.econtext.ai/api/groups/group/10XxLiRU7RvltY8zqI8fxK",
          "name": "api-free",
          "custom_data": {
            "classify_limit": 10,
            "tier_depth": 9,
            "monthly_limit": 10000
          },
          "created_at": "2017-04-12 21:46:02.332000+00:00",
          "modified_at": "2017-04-12 21:46:02.343000+00:00",
          "id": "10XxLiRU7RvltY8zqI8fxK",
          "description": "Free tier API users"
        },
        {
          "status": "ENABLED",
          "application": "6CyMm9ikHh5ASSFOZ9OUWo",
```

```

      "href": "https://auth.econtext.ai/api/groups/group/
7Pmrb9Leujmf8gcspoRvje",
      "name": "admin",
      "custom_data": {
        "company_id": 9999999,
        "_no_encrypt": true,
        "tier_depth": 9999
      },
      "created_at": "2017-04-12 21:46:02.301000+00:00",
      "modified_at": "2017-04-12 21:46:02.310000+00:00",
      "id": "7Pmrb9Leujmf8gcspoRvje",
      "description": "API admin users (typically admin or internal)"
    }
  ],
  "elapsed": 0.03382992744445801
}

```

5.2 POST /groups/group

Create a new group object

- *Resource URL*
- *Parameters*
- *Return*
- *Example Request*
 - *POST Request*
 - *POST Response*

5.2.1 Resource URL

<https://auth.econtext.ai/api/groups/group>

5.2.2 Parameters

Parameter	Type	Description
name	string	A name for the Group
description	string	A description for the Group
custom_data	object	A JSON object containing arbitrary data
application	string	An Application ID

5.2.3 Return

A newly created Group Object

5.2.4 Example Request

POST Request

```
curl -X POST -u username:password \
  -H 'content-type: application/json' \
  -d '{
    "name": "Test Group",
    "description": "A test group",
    "custom_data": {
      "tier_depth": 9
    },
    "application": "bf59fd16-d003-45ae-889d-4ed06b804c21"
  }' \
  https://auth.econtext.ai/api/groups/group
```

POST Response

```
{
  "econtext": {
    "result": {
      "group": {
        "status": "ENABLED",
        "application": "bf59fd16-d003-45ae-889d-4ed06b804c21",
        "href": "https://auth.econtext.ai/api/groups/group/16191e59-85e8-416f-826d-9cf8106c8cad",
        "name": "Test Group",
        "custom_data": {
          "tier_depth": 9
        },
        "created_at": "2017-04-20 15:37:46.092000+00:00",
        "modified_at": "2017-04-20 15:37:46.116000+00:00",
        "id": "16191e59-85e8-416f-826d-9cf8106c8cad",
        "description": "A test group"
      },
      "elapsed": 0.05748295783996582
    }
  }
}
```

5.3 GET /groups/group/{groupid}

Retrieve an existing group object identified by {groupid}

- *Resource URL*
- *Return*
- *Example Request*
 - *GET Request*
 - *GET Response*

5.3.1 Resource URL

<https://auth.econtext.ai/api/groups/group/\protect\Tl\textbraceleftappid\protect\Tl\textbraceright>

5.3.2 Return

A Group object identified by {groupid}

5.3.3 Example Request

GET Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/groups/group/16191e59-85e8-416f-826d-9cf8106c8cad
```

GET Response

```
{
  "econtext": {
    "result": {
      "group": {
        "status": "ENABLED",
        "application": "bf59fd16-d003-45ae-889d-4ed06b804c21",
        "href": "https://auth.econtext.ai/api/groups/group/16191e59-85e8-416f-826d-9cf8106c8cad",
        "name": "Test Group",
        "custom_data": {
          "tier_depth": 9
        },
        "created_at": "2017-04-20 15:37:46.092000+00:00",
        "modified_at": "2017-04-20 15:37:46.116000+00:00",
        "id": "16191e59-85e8-416f-826d-9cf8106c8cad",
        "description": "A test group"
      }
    },
    "elapsed": 0.05748295783996582
  }
}
```

5.4 PUT /groups/group/{groupid}

Update an existing group object identified by {groupid}

- *Resource URL*
- *Parameters*
- *Return*
- *Example Request*
 - *PUT Request*
 - *PUT Response*

5.4.1 Resource URL

`https://auth.econtext.ai/api/groups/group/{groupid}`

5.4.2 Parameters

Parameter	Type	Description
name	string	A name for the Group
description	string	A description for the Group
custom_data	object	A JSON object containing arbitrary data
status	string	The status of the Group. Available options are <code>ENABLED</code> and <code>DISABLED</code>

5.4.3 Return

An updated Group object

5.4.4 Example Request

PUT Request

```
curl -X PUT -u username:password \
  -H 'content-type: application/json' \
  -d '{
    "description": "A more detailed description of my group"
  }' \
  https://auth.econtext.ai/api/groups/group/16191e59-85e8-416f-826d-9cf8106c8cad
```

PUT Response

```
{
  "econtext": {
    "result": {
```



```

    "group": {
      "status": "ENABLED",
      "application": "bf59fd16-d003-45ae-889d-4ed06b804c21",
      "href": "https://auth.econtext.ai/api/groups/group/16191e59-85e8-416f-826d-9cf8106c8cad",
      "name": "Test Group",
      "custom_data": {
        "tier_depth": 9
      },
      "created_at": "2017-04-20 15:37:46.092000+00:00",
      "modified_at": "2017-04-20 15:40:10.116000+00:00",
      "id": "16191e59-85e8-416f-826d-9cf8106c8cad",
      "description": "A more detailed description of my group"
    },
    "elapsed": 0.05748295783996582
  }
}

```

5.5 DELETE /groups/group/{groupid}

Delete the Group identified by {groupid}. The Group's status must be set to DISABLED before deletion is possible.

- *Resource URL*
- *Return*
- *Example Request*
 - *DELETE Request*
 - *DELETE Response*

5.5.1 Resource URL

`https://auth.econtext.ai/api/groups/group/{groupid}`

5.5.2 Return

A boolean response explaining whether the DELETE call was successful or not.

5.5.3 Example Request

DELETE Request

```

curl -X DELETE -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/groups/group/16191e59-85e8-416f-826d-9cf8106c8cad

```

DELETE Response

```
{
  "econtext": {
    "result": {
      "deleted": true
    },
    "elapsed": 0.0257871150970459
  }
}
```

5.6 GET /groups/group/{groupid}/users

Retrieve a list of users currently associated with the Group identified by {groupid}

- *Resource URL*
- *Return*
- *Example Request*
 - *GET Request*
 - *GET Response*

5.6.1 Resource URL

`https://auth.econtext.ai/api/groups/group/{groupid}/users`

5.6.2 Return

A list of Users associated with the Group.

5.6.3 Example Request

GET Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/groups/group/16191e59-85e8-416f-826d-9cf8106c8cad/
users
```

GET Response

```
{
  "econtext": {
    "result": {
      "users": [
        {
```

```
    "username": "test-user@econtext.ai",
    "status": "UNVERIFIED",
    "applications": [
      "ec48dad3-ca61-4d74-a584-3ee3db4708ef"
    ],
    "href": "https://auth.econtext.ai/api/users/user/
a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
    "groups": [
      "16191e59-85e8-416f-826d-9cf8106c8cad"
    ],
    "apikeys": [],
    "id": "a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
    "name": "Test User",
    "created_at": "2017-04-20 15:55:08.339000+00:00",
    "modified_at": "2017-04-20 16:00:37.922000+00:00",
    "custom_data": null,
    "email": "test-user@econtext.ai"
  }
]
},
"elapsed": 0.004544973373413086
}
}
```


A User is a resource that can be authenticated in this API.

A User may belong to multiple Applications and Groups, and may authenticate successfully against any Application that it is associated with with.

6.1 GET /users

Retrieve a list of users available in the system. This call is resource intensive and should generally be avoided in favor of retrieving more targeted lists via search or application or group listings.

- *Resource URL*
- *Return*
- *Example Request*
 - *GET Request*
 - *GET Response*

6.1.1 Resource URL

<https://auth.econtext.ai/api/users>

6.1.2 Return

A list of Users

6.1.3 Example Request

GET Request

```
curl -X GET -u username:password \  
  -H 'content-type: application/json' \  
  https://auth.econtext.ai/api/users
```

GET Response

```
{  
  "econtext": {  
    "result": {  
      "users": [  
        {  
          "username": "test-user@econtext.ai",  
          "status": "UNVERIFIED",  
          "applications": [  
            {  
              "status": "ENABLED",  
              "href": "https://auth.econtext.ai/api/applications/application/  
609543d6-1cca-4039-9c1a-c843bda15ba4",  
              "name": "Test Application",  
              "custom_data": null,  
              "created_at": "2017-05-08 19:21:13.381000+00:00",  
              "modified_at": "2017-05-08 19:21:13.381000+00:00",  
              "id": "609543d6-1cca-4039-9c1a-c843bda15ba4",  
              "description": "Test application for use in this round of tests"  
            }  
          ],  
          "href": "https://auth.econtext.ai/api/users/user/  
a3bc334a-f9f2-4797-aaa2-1440811c0ec0",  
          "groups": [  
            {  
              "status": "ENABLED",  
              "application": "609543d6-1cca-4039-9c1a-c843bda15ba4",  
              "href": "https://auth.econtext.ai/api/groups/group/  
5fb6f5d6-3a6d-4e1d-83bb-7445274745bf",  
              "name": "Test Group",  
              "custom_data": {  
                "tier_depth": 9  
              },  
              "created_at": "2017-05-08 19:22:48.867000+00:00",  
              "modified_at": "2017-05-08 19:22:48.871000+00:00",  
              "id": "5fb6f5d6-3a6d-4e1d-83bb-7445274745bf",  
              "description": "A test group with some custom_data"  
            }  
          ],  
          "apikeys": [],  
          "id": "a3bc334a-f9f2-4797-aaa2-1440811c0ec0",  
          "name": "Test User",  
          "created_at": "2017-04-20 15:55:08.339000+00:00",  
          "modified_at": "2017-04-20 16:00:37.922000+00:00",  
        }  
      ]  
    }  
  }  
}
```

```

        "custom_data": null,
        "email": "test-user@econtext.ai"
    }
  ],
  "elapsed": 0.004544973373413086
}

```

6.2 GET /users/search/{search}

Retrieve a list of users available in the system that match the provided search term. This method performs a case-insensitive “contains” search against a User’s email, name, id, API Keys, and a company name (inside custom_data) if it exists.

- *Resource URL*
- *Return*
- *Example Request*
 - *GET Request*
 - *GET Response*

6.2.1 Resource URL

`https://auth.econtext.ai/api/users/search/{search}`

6.2.2 Return

A list of Users

6.2.3 Example Request

GET Request

```

curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/users/search/econtext.ai

```

GET Response

```

{
  "econtext": {
    "result": {
      "users": [
        {
          "username": "test-user@econtext.ai",

```

```
    "status": "UNVERIFIED",
    "applications": [],
    "href": "https://auth.econtext.ai/api/users/user/
a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
    "groups": [
      {
        "status": "ENABLED",
        "application": "609543d6-1cca-4039-9c1a-c843bda15ba4",
        "href": "https://auth.econtext.ai/api/groups/group/
5fb6f5d6-3a6d-4e1d-83bb-7445274745bf",
        "name": "Test Group",
        "custom_data": {
          "tier_depth": 9
        },
        "created_at": "2017-05-08 19:22:48.867000+00:00",
        "modified_at": "2017-05-08 19:22:48.871000+00:00",
        "id": "5fb6f5d6-3a6d-4e1d-83bb-7445274745bf",
        "description": "A test group with some custom_data"
      }
    ],
    "apikeys": [],
    "id": "a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
    "name": "Test User",
    "created_at": "2017-04-20 15:55:08.339000+00:00",
    "modified_at": "2017-04-20 19:46:25.426000+00:00",
    "custom_data": null,
    "email": "test-user@econtext.ai"
  }
],
"elapsed": 0.011507034301757812
}
```

6.3 POST /users/user

Create a new User object

- *Resource URL*
- *Parameters*
- *Return*
- *Example Request*
 - *POST Request*
 - *POST Response*

6.3.1 Resource URL

<https://auth.econtext.ai/api/users/user>

6.3.2 Parameters

Parameter	Type	Description
email	string	A valid email address
password	string	A password for the User - must be at least 7 characters long
name	string	A name for the User
custom_data	object	A JSON object containing arbitrary data
applications	array	A list of Application object IDs
groups	array	A list of Group object IDs

6.3.3 Return

A newly created User object

6.3.4 Example Request

POST Request

```
curl -X POST -H 'authorization: Basic b3BzQGluZm8uY29tOnAxdjBwcjBzMW0=' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '{
  "name": "Test User",
  "email": "test-user@econtext.ai",
  "password": "a new password",
  "applications": ["ec48dad3-ca61-4d74-a584-3ee3db4708ef"]
}' \
https://auth.econtext.ai/api/users/user
```

POST Response

```
{
  "econtext": {
    "result": {
      "user": {
        "username": "test-user@econtext.ai",
        "status": "UNVERIFIED",
        "applications": [
          {
            "status": "ENABLED",
            "href": "https://auth.econtext.ai/api/applications/application/609543d6-1cca-4039-9c1a-c843bda15ba4",
            "name": "Test Application",
            "custom_data": null,
            "created_at": "2017-05-08 19:21:13.381000+00:00",

```

```
    "modified_at": "2017-05-08 19:21:13.381000+00:00",
    "id": "609543d6-1cca-4039-9c1a-c843bda15ba4",
    "description": "Test application for use in this round of tests"
  }
],
"href": "api_url:users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
"groups": [
  {
    "status": "ENABLED",
    "application": "609543d6-1cca-4039-9c1a-c843bda15ba4",
    "href": "https://auth.econtext.ai/api/groups/group/
5fb6f5d6-3a6d-4e1d-83bb-7445274745bf",
    "name": "Test Group",
    "custom_data": {
      "tier_depth": 9
    },
    "created_at": "2017-05-08 19:22:48.867000+00:00",
    "modified_at": "2017-05-08 19:22:48.871000+00:00",
    "id": "5fb6f5d6-3a6d-4e1d-83bb-7445274745bf",
    "description": "A test group with some custom_data"
  }
],
"apikey": [],
"id": "a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
"name": "Test User",
"created_at": "2017-04-20 15:55:08.339000+00:00",
"modified_at": "2017-04-20 15:55:08.339000+00:00",
"custom_data": null,
"email": "test-user@econtext.ai"
}
},
"elapsed": 0.0005970001220703125
}
}
```

6.4 GET /users/user/{userid}

Retrieve an existing User object identified by {userid}

- *Resource URL*
- *Return*
- *Example Request*
 - *GET Request*
 - *GET Response*

6.4.1 Resource URL

`https://auth.econtext.ai/api/users/user/{userid}`

6.4.2 Return

A User object identified by {userid}

6.4.3 Example Request

GET Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0
```

GET Response

```
{
  "econtext": {
    "result": {
      "user": {
        "username": "test-user@econtext.ai",
        "status": "UNVERIFIED",
        "applications": [
          {
            "status": "ENABLED",
            "href": "https://auth.econtext.ai/api/applications/application/609543d6-1cca-4039-9c1a-c843bda15ba4",
            "name": "Test Application",
            "custom_data": null,
            "created_at": "2017-05-08 19:21:13.381000+00:00",
            "modified_at": "2017-05-08 19:21:13.381000+00:00",
            "id": "609543d6-1cca-4039-9c1a-c843bda15ba4",
            "description": "Test application for use in this round of tests"
          }
        ],
        "href": "https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
        "groups": [],
        "apikeys": [],
        "id": "a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
        "name": "Test User",
        "created_at": "2017-04-20 15:55:08.339000+00:00",
        "modified_at": "2017-04-20 15:55:08.339000+00:00",
        "custom_data": null,
        "email": "test-user@econtext.ai"
      }
    },
    "elapsed": 0.0007190704345703125
  }
}
```

6.5 PUT /users/user/{userid}

Update an existing User object identified by {userid}

- *Resource URL*
- *Parameters*
- *Return*
- *Example Request*
 - *PUT Request*
 - *PUT Response*

6.5.1 Resource URL

`https://auth.econtext.ai/api/users/user/{userid}`

6.5.2 Parameters

Parameter	Type	Description
email	string	A valid email address
password	string	A password for the User - must be at least 7 characters long
name	string	A name for the User
custom_data	object	A JSON object containing arbitrary data
applications	array	A list of Application object IDs
groups	array	A list of Group object IDs
status	string	The status of the User. Available options are ENABLED, UNVERIFIED, and DISABLED

6.5.3 Return

An updated User object

6.5.4 Example Request

PUT Request

```
curl -X PUT -u username:password \
  -H 'content-type: application/json' \
  -d '{
    "status": "ENABLED"
    "groups": ["16191e59-85e8-416f-826d-9cf8106c8cad"]
  }' \
  https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0
```

PUT Response

```

{
  "econtext": {
    "result": {
      "user": {
        "username": "test-user@econtext.ai",
        "status": "ENABLED",
        "applications": [
          {
            "status": "ENABLED",
            "href": "https://auth.econtext.ai/api/applications/application/609543d6-1cca-4039-9c1a-c843bda15ba4",
            "name": "Test Application",
            "custom_data": null,
            "created_at": "2017-05-08 19:21:13.381000+00:00",
            "modified_at": "2017-05-08 19:21:13.381000+00:00",
            "id": "609543d6-1cca-4039-9c1a-c843bda15ba4",
            "description": "Test application for use in this round of tests"
          }
        ],
        "href": "https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
        "groups": [
          {
            "status": "ENABLED",
            "application": "609543d6-1cca-4039-9c1a-c843bda15ba4",
            "href": "https://auth.econtext.ai/api/groups/group/5fb6f5d6-3a6d-4e1d-83bb-7445274745bf",
            "name": "Test Group",
            "custom_data": {
              "tier_depth": 9
            },
            "created_at": "2017-05-08 19:22:48.867000+00:00",
            "modified_at": "2017-05-08 19:22:48.871000+00:00",
            "id": "5fb6f5d6-3a6d-4e1d-83bb-7445274745bf",
            "description": "A test group with some custom_data"
          }
        ],
        "apikeys": [],
        "id": "a3bc334a-f9f2-4797-aaa2-1440811c0ec0",
        "name": "Test User",
        "created_at": "2017-04-20 15:55:08.339000+00:00",
        "modified_at": "2017-04-20 16:00:37.922000+00:00",
        "custom_data": null,
        "email": "test-user@econtext.ai"
      }
    },
    "elapsed": 0.0007190704345703125
  }
}

```

6.6 POST /users/user/{userid}/apikey

Create a new API Key object for a User

- *Resource URL*
- *Parameters*
- *Return*
- *Example Request*
 - *POST Request*
 - *POST Response*

6.6.1 Resource URL

`https://auth.econtext.ai/api/users/user/{userid}/apikey`

6.6.2 Parameters

Parameter	Type	Description
name	string	A name for the API Key
description	string	A description for the API Key

6.6.3 Return

A newly created API Key object including the generated secret. The secret is only returned as a result of the POST call. Subsequent calls to retrieve this key will not return the secret, and it is hashed in the database. Please be sure to pass the secret back to the user for safe storage.

6.6.4 Example Request

POST Request

```
curl -X POST \
  -H 'authorization: Basic b3BzQGluZm8uY29tOnAxdjBwcjBzMW0=' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '{
    "name": "Test API Key",
    "description": "An API Key for testing"
  }' \
  https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0/
apikey
```

POST Response

```
{
  "econtext": {
    "result": {
      "apikey": {
        "status": "ENABLED",
        "description": "An API Key for testing",
        "secret": "ODBkZjZiNzYtMzU1Ny00MDgxLWFiMDYtMWE1OGU5OTIxZGQ0",
        "href": "https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0/apkey/44A62GBSEB4F3S08VFLWI0YCY",
        "id": "44A62GBSEB4F3S08VFLWI0YCY",
        "name": "Test API Key"
      }
    },
    "elapsed": 0.06387519836425781
  }
}
```

6.7 GET /users/user/{userid}/apikey/{apikeyid}

Retrieve an existing API Key object for a User

- *Resource URL*
- *Return*
- *Example Request*
 - *GET Request*
 - *GET Response*

6.7.1 Resource URL

`https://auth.econtext.ai/api/users/user/{userid}/apikey/{apikeyid}`

6.7.2 Return

An API Key object

6.7.3 Example Request

GET Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0/apkey/44A62GBSEB4F3S08VFLWI0YCY
```

GET Response

```
{
  "econtext": {
    "result": {
      "apikey": {
        "status": "ENABLED",
        "description": "An API Key for testing",
        "href": "https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0/apkey/44A62GBSEB4F3S08VFLWI0YCY",
        "id": "44A62GBSEB4F3S08VFLWI0YCY",
        "name": "Test API Key"
      }
    },
    "elapsed": 0.06387519836425781
  }
}
```

6.8 PUT /users/user/{userid}/apikey/{apikeyid}

Update an API Key object. An API Key which has a status of `DISABLED` may not be used to authenticate a User.

- *Resource URL*
- *Parameters*
- *Return*
- *Example Request*
 - *PUT Request*
 - *PUT Response*

6.8.1 Resource URL

`https://auth.econtext.ai/api/users/user\protect\T1\textbraceleftuserid\protect\T1\textbraceright/apkey\protect\T1\textbraceleftapikeyid\protect\T1\textbraceright`

6.8.2 Parameters

Parameter	Type	Description
name	string	A name for the API Key
description	string	A description for the API Key
status	string	The status of the API Key. Available options are <code>ENABLED</code> and <code>DISABLED</code>

6.8.3 Return

An API Key object

6.8.4 Example Request

PUT Request

```
curl -X POST \
  -H 'authorization: Basic b3BzQGluZm8uY29tOnAxdjBwcjBzMW0=' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '{
    "status": "DISABLED"
  }' \
  https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0/
apikey/44A62GBSEB4F3S08VFLWI0YCY
```

PUT Response

```
{
  "econtext": {
    "result": {
      "apikey": {
        "status": "DISABLED",
        "description": "An API Key for testing",
        "secret": "ODbkZjZiNzYtMzU1Ny00MDgxLWFiMDYtMWE1OGU5OTIxZGQ0",
        "href": "https://auth.econtext.ai/api/users/user/
a3bc334a-f9f2-4797-aaa2-1440811c0ec0/apkey/44A62GBSEB4F3S08VFLWI0YCY",
        "id": "44A62GBSEB4F3S08VFLWI0YCY",
        "name": "Test API Key"
      }
    },
    "elapsed": 0.029818058013916016
  }
}
```

6.9 DELETE /users/user/{userid}/apikey/{apikeyid}

Remove an API Key from a User

- *Resource URL*
- *Return*
- *Example Request*
 - *DELETE Request*
 - *DELETE Response*

6.9.1 Resource URL

`https://auth.econtext.ai/api/users/user/{userid}/apikey/{apikeyid}`

6.9.2 Return

A boolean response explaining whether the DELETE call was successful or not.

6.9.3 Example Request

DELETE Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0/
apikey/44A62GBSEB4F3S08VFLWI0YCY
```

DELETE Response

```
{
  "econtext": {
    "result": {
      "deleted": true
    },
    "elapsed": 0.01317906379699707
  }
}
```

6.10 DELETE /users/user/{userid}/application/{appid}

Remove an Application from a User. Please note that a User must be associated with at least one Application. Removing the last Application from a User will result in a 409 Conflict error.

- *Resource URL*
- *Return*
- *Example Request*
 - *DELETE Request*
 - *DELETE Response*

6.10.1 Resource URL

`https://auth.econtext.ai/api/users/user/{userid}/application/{appid}`

6.10.2 Return

A boolean response explaining whether the DELETE call was successful or not.

6.10.3 Example Request

DELETE Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0/
application/aa136fbd-26cc-448f-af0a-cf98fe165cc6
```

DELETE Response

```
{
  "econtext": {
    "result": {
      "deleted": true
    },
    "elapsed": 0.030797958374023438
  }
}
```

6.11 DELETE /users/user/{userid}/group/{groupid}

Remove a Group from a User

- *Resource URL*
- *Return*
- *Example Request*
 - *DELETE Request*
 - *DELETE Response*

6.11.1 Resource URL

`https://auth.econtext.ai/api/users/user\protect\T1\textbraceleftuserid\protect\T1\textbraceright/group\protect\T1\textbraceleftgroupid\protect\T1\textbraceright`

6.11.2 Return

A boolean response explaining whether the DELETE call was successful or not.

6.11.3 Example Request

DELETE Request

```
curl -X GET -u username:password \
  -H 'content-type: application/json' \
```

```
https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0/group/16191e59-85e8-416f-826d-9cf8106c8cad
```

DELETE Response

```
{
  "econtext": {
    "result": {
      "deleted": true
    },
    "elapsed": 0.030797958374023438
  }
}
```

6.12 DELETE /users/user/{userid}

Delete the User identified by {userid}. The User's status must be set to DISABLED before deletion is possible.

- *Resource URL*
- *Return*
- *Example Request*
 - *DELETE Request*
 - *DELETE Response*

6.12.1 Resource URL

```
https://auth.econtext.ai/api/users/user/{userid}
```

6.12.2 Return

A boolean response explaining whether the DELETE call was successful or not.

6.12.3 Example Request

DELETE Request

```
curl -X DELETE -u username:password \
  -H 'content-type: application/json' \
  https://auth.econtext.ai/api/users/user/a3bc334a-f9f2-4797-aaa2-1440811c0ec0
```

DELETE Response

```
{
  "econtext": {
    "result": {
      "deleted": true
    },
    "elapsed": 0.0257871150970459
  }
}
```


CHAPTER 7

Changelog

7.1 0.0.1 – 2017-04-20

- Initial release

CHAPTER 8

Notes

Unless otherwise specified, all examples in this documentation use a Content-Type ([RFC 2616#section-14.17](#)) of JSON for both input and output.

CHAPTER 9

Indices and tables

- `genindex`
- `search`
- *Changelog*

R

RFC

RFC 2616#section-14.1, 1

RFC 2616#section-14.17, 1, 45

RFC 4627, 1