IBM Cloud Identity Service

EAI Me API Guide

IBM Cloud Identity Service

EAI Me API Guide

EAI Me API Guide

Contents

Cloud Identity Service EAI Me integration

Integrate Cloud Identity Service with the External Authentication Interface (EAI) application, the Me API adds a number of single-user API calls.

The External Authentication Interface (EAI) application handles authentication and session management for web resources that are protected by Cloud Identity Service.

Overview

Integrate Cloud Identity Service with the External Authentication Interface (EAI) application. The EAI application handles authentication and session management for protected web resources. The Me API adds a number of single-user API calls that have counterparts in the Cloud Identity Service API.

Authentication

Authentication with **OAuth** tokens.

Authenticate to the Me API

To get an **OAuth** token for a user, the API requires an **OAuth2** Basic Authorization header of a known **client ID**. The current implementation supports a hardcoded client ID, **eai-client**. The examples of the two Authentication calls that follow include the required **Basic Authorization** header to send.

Username and password authentication

The API supports authentication that uses the **OAuth 2.0 password** flow.

Method

```
POST /EAI/oauth/token
```

Content type

```
application/x-www-form-urlencoded
```

Example cURL requests

```
curl -X POST -H "Content-Type:application/x-www-form-urlencoded" -H
"Authorization: Basic ZWFpLWNsaWVudDo=" -d "grant_type=password
&username=gorditapassword=IluvTr3ats!" https://gateway.domain.com/EAI/
oauth/token

curl -X POST -H "Content-Type:application/json" -H "Authorization: Basic
ZWFpLWNsaWVudDo=" "https://gateway.domain.com/EAI/oauth/
```

```
token?grant_type=password&username=testuser &password=testpassword"
```

Request parameters

Table 1. Request parameters

Parameter name	Description
grant_type	The user's **password**.
username	The user's **username**.
password	The provided **password**.

Example response

```
{
access_token: "4ed14dd2-d4f3-4089-8f06-02ae42a08420"
token_type: "bearer"
refresh_token: "c11fbcad-fb04-4444-abce-1fd3923bc611"
expires_in: 3194
scope: "read"
}
```

Returns

200: OK for success.

401: Unauthorized for every other response.

403: Forbidden if the account is locked for any reason.

Social media federation

The API also supports federating a social media sign in.

The social media data is passed in through a **JSON Web Token** (JWT) and complies with the OAuth 2.0 JWT Assertion Profile.

Method

```
POST /EAI/oauth/token
```

Content type

```
application/x-www-form-urlencoded
```

Example cURL request

```
curl -X POST -H "Content-Type:application/x-www-form-urlencoded" -H
"Authorization: Basic ZWFpLWNsaWVudDo=" -d "grant_type=
urn%3Aietf%3Aparams%3Aoauth%3Agranttype% 3Ajwtbearer&assertion=eyJhbGciOi
Jub25lIiwidHlwIjoiSdUIn0=.eyJleHAiOiIxNDE zMjcxNDU0NjI2IiwicGxhdCI6ImZhY2
Vib29rIiwic3ViIjoiMjc2ODI3ODY5MTQxODU4Iiwi dG9rZW4iOiJDQUFEN3hnTE4yMkl
CQUpnQWlsbnBzSDdRZHg4Tk9KUmxGWTY4eVh5dURoZXBRS lB1QjJKV042dVJqdXVoc1pDZ0
YwWkNmN1luU0hGbW1yN1pDU0FWSkduOURHb2ZkVndlWVBHdj d4dXl4ejhZdzhvWkF6T3U0
WkJTSHRxbUhIOEZ0NHIzQU9JWkI2VUptWDNaQXFnZ3hmRThNWWh aQkkzN0c0em94VEROMWR
PRm5PdHV1SmV5NVI3RE9Wajd0YUpiVXdieVJjMjQyM3JKRW9vc3g2 UWVUbGFCIiwibmJm
IjoiMTQxMzI2Nzg1NDYyNiIsImlzcyI6Imh0dHBzOi8vd3d3LmxpbmNvb G4uY29tLmNu
IiwianRpIjoiNzk0M2RhNGQtMTBmYy00MWUwLThjNjgtNjIzZDUyYzk0MzRhIi
widHlwIjoiaHR0cHM6Ly93d3cuaWJtLmNvbS9nYXRld2F5L3NvY2lhbCIsImlhdCI6Ij
E0MTM yNjc4NTQ2MjYifQ==. "https://gateway.domain.com/EAI/oauth/token
```

Request parameters

Table 2. Request parameters

Parameter name	Description
grant_type	Must be `urn:ietf:params:oauth:grant-type:jwt-bearer` The assertion is a **JSON Web Token** (JWT) that contains the following claims: **exp**: The expiration time of the claim. Optional. **plat**: The social media platform on which the user authenticated. One of the following: *facebook*, *google*, *qq*, *renren*, *wechat*, *weibo*, and *yahoo*. **sub**: The ID of the social media platform application ID. **token**: The authorization **token** that was received when the social media authorization completed. **iss**: A URL representing who issued the **token**. **jti**: A unique ID for the token. Optional. **typ**: The type of token. Currently only `urn:com:ibm:cloud identity:social` is supported. The supported JWT timestamps: exp, nbf, iat. Optional. Currently, the **JSON Web Token** must be issued in plain text, with the header used in the example that follows this table.

When you are encoding in the the above **JWT** data structure in the Request parameters table, you follow the rules of the standard. The output looks similar to the codeblock. The example is formatted to break across lines. The actual submission must be a single line:

```
// header { "alg": "none", "typ": "JWT" }
// claims { "exp": "1413271454626", "plat":"Facebook",
"sub":"276827869141858", "token":"TOKEN", "nbf":"1413267854626",
"iss":"ISSUER", "jti":"ID", "typ":"urn:com:ibm:cloudidentity:
social", "iat":"1413267854626" } eyJhbGciOiJub25lIiwidHlwI
joiSldUIn0=.eyJleHAiOiIxNDEzMjcxNDU0NjI2IiwicGxhdCI6ImZhY2Vib
29rIiwic3ViIjoiMjc2ODI3ODY5MTQxODU4IiwidG9rZW4iOiJDQUFEN3hnTE
4yMklCQUpnQWlsbnBzSDdRZHg4Tk9KUmxGWTY4eVh5dURoZXBRSlB1QjJKV0
42dVJqdXVoc1pDZ0YwWkNmN1luU0hGbW1yN1pDU0FWSkduOURHb2ZkVndl
WVBHdjd4dXl4ejhZdzhvWkF6T3U0WkJTSHRxbUhIOEZ0NHIzQU9JWkI2VUpt
WDNaQXFnZ3hmRThNWWhaQkkzN0c0em94VEROMWRPRm5PdHV1SmV5NVI3RE9
Wajd0YUpiVXdieVJjMjQyM3JKRW9vc3g2UWVUbGFCIiwibmJmIjoiMTQxMz
I2Nzg1NDYyNiIsImlzcyI6Imh0dHBzOi8vd3d3LmxpbmNvbG4uY29tLmNu
IiwianRpIjoiNzk0M2RhNGQtMTBmYy00MWUwLThjNjgtNjIzZDUyYzk0Mz
RhIiwidHlwIjoiaHR0cHM6Ly93d3cuaWJtLmNvbS9nYXRld2F5L
3NvY2lhbCIsImlhdCI6IjE0MTMyNjc4NTQ2MjYifQ==.
```

Example response

```
{
access_token: "4ed14dd2-d4f3-4089-8f06-02ae42a08420"
token_type: "bearer"
refresh_token: "c11fbcad-fb04-4444-abce-1fd3923bc611"
expires_in: 3194
scope: "read"
}
```

Returns

200: OK for success.

401: Unauthorized for every other response.

403: Forbidden if the account is locked for any reason.

Refresh tokens

When the token is near expiry, you can use the refresh token to get a new access token.

Method

```
POST /EAI/oauth/token
```

Content type

```
application/x-www-form-urlencoded
```

Example cURL requests

```
curl -X POST -H "Content-Type:application/x-www-form-urlencoded" -H
"Authorization: Basic ZWFpLWNsaWVudDo=" -d "grant_type=refresh_token
&client_id=eai-client&refresh_token=c11fbcad-fb04-4444- abce-1fd3923bc611"
https://gateway.domain.com/EAI/oauth/token

curl -X POST -H "Authorization: Basic ZWFpLWNsaWVudDo=" -H "Content-Type:
application/json" "https://gateway.domain.com/EAI/oauth/
token?grant_type=refresh_token&client_id=eai-client&refresh_token=7123fb5e-
47a8-4c63-a913-85064b29dc0c"
```

Request parameters

Table 3. Request parameters

Parameter name	Description
grant_type	Required parameter is **refresh_token**.
client_id	Required parameter is **eai-client**.
refresh_token	Provides the **refresh_token** during the authentication.

Example response

```
{
access_token: "4ed14dd2-d4f3-4089-8f06-02ae42a08420"
token_type: "bearer"
refresh_token: "c11fbcad-fb04-4444-abce-1fd3923bc611"
expires_in: 3194
scope: "read"
}
```

Returns

200: OK for success.

401: Unauthorized for every other response.

403: Forbidden if the account is locked for any reason.

Validate tokens

Use this API to determine whether an Oauth token is valid now.

Method

```
GET /EAI/oauth/check_token
```

Example cURL requests

```
curl https://gateway.domain.com/EAI/oauth/check_token?token=4ed14dd2-d4f3-
4089-8f06-02ae42a08420
```

Request parameters

Table 4. Request parameters

Parameter name	Description
token	The Oauth token to check.

Example response

```
{
"authorities" : [ "ROLE_CLIENT" ],
"client_id" : "eai-client",
"exp" : 1418616268,
"scope" : [ "read" ],
"user_name" : "eaitest"
}
```

Returns

200: OK for success.

400: Bad request if the token is not recognized.

Single-user operations

Operations available as the user.

Authenticate to the Me API

All operations require that the **access token** from the authentication flow to be presented as a **Bearer token**.

Get user details

Retrieves the user details of the authenticated user. Returns all available attributes.

Method

```
GET /EAI/api/me
```

Content type:

```
application/json
```

Example cURL requests

```
curl -H "Authorization: Bearer access_token" https://gateway.domain.com/
EAI/api/me
```

Request parameters

None.

Example response

```
{"status" : "success",
"entry" : {
"status" : null,
"gtwayUUID" : "323966f1-0780-4fb4-928b-8fe3d4f19b94",
"uid" : "test",
"gma_isAccount" : true,
"uid" : "test",
"mail" : "test@us.ibm.com",
"gtwayPrincipalName" : "test",
"sn" : "testing",
"gtwayPrefLanguage" : "en-us",
"c" : "usa",
"cn" : "test testing",
"gtwayIsManager" : "true",
"gtwayUUID" : "323966f1-0780-4fb4-928b-8fe3d4f19b94",
"givenName" : "Test",
"employeeNumber" : "1234567890"
},
"totalCount" : 1}
```

Returns

200: OK for success.

Get services

Retrieves the services to which the authenticated user belongs.

Method

```
GET /EAI/api/me/services
```

Content type:

```
application/json
```

Example cURL requests

```
curl -H "Authorization: Bearer access_token" https://gateway.domain.com/
EAI/api/me/services
```

Request parameters

None.

Example response

```
{
"status" : "success",
"entry" : [ "svc_GatewayWAMService", "svc_test service" ],
"totalCount" : 2
}
```

Returns

200: OK for success.

Get roles

Retrieves the roles to which the authenticated user belongs.

Method

```
GET /EAI/api/me/roles
```

Content type:

```
application/json
```

Example cURL requests

```
curl -H "Authorization: Bearer access_token" https://gateway.domain.com/
EAI/api/me/roles
```

Request parameters

None.

Example response

```
{
"status" : "success",
"entry" : [ "Help Desk", "Manager", "Default" ],
"totalCount" : 3
}
```

Returns

200
OK for success.

404
HttpStatus.NOT_FOUND

Get KBA questions

Retrieves the Knowledge Based Authentication (KBA) questions, also known as security questions, that the user defines. The Get KBA questions method returns the question numbers only, and must be paired with the GmaApi's KBA methods to retrieve the actual question text in the desired language.

Method

```
GET /EAI/api/me/kba
```

Content type:

`application/json`

Example cURL requests

```
curl -H "Authorization: Bearer access_token" https://gateway.domain.com/
EAI/api/me/kba?showAnswers=true
```

Request parameters

Table 5. Request parameters

Parameter name	Description
showAnswers	Optional. Boolean value: whether to provide answers. When set to `true`, provide answers. When set to `false`, hide the answers. Defaults to `false`.

Example response

```
{
"status" : "success",
"entry" : [ {
"questionNumber" : 1,
"answer" : "1952"
}, {
"questionNumber" : 2,
"answer" : "1950"
}, {
"questionNumber" : 5,
"answer" : "smith"
} ],
"totalCount" : 3
}
```

Returns

200: OK for success.

Change password

Allows the user to change their password. The current password must be available.

Method

`POST /EAI/api/me/changePassword`

Content type:

`application/json`

Example cURL requests

```
curl -H "Authorization: Bearer access_token" -d "currentPassword=Passw0rd!
&newPassword=MyNewPassw0rd!" https://gateway.domain.com/EAI/api/me/
changePassword
```

Request parameters

Table 6. Request parameters

Parameter name	Description
currentPassword	Required: the user's **current password**. The user's **current password** is validated before you change the password.
newPassword	Required. The user's desired **new password**. All applicable password policies are applied to the password change attempt. Clients perform client-side verification of the password to ensure that it meets password complexity rules before submitting a request.

Example response

```
{
"status" : "success",
}
```

Returns

200: OK for success.

401: Unauthorized if the current password fails validation.

403: Forbidden if the new password fails validation.

412: Precondition that is failed if the new password was found in history.

Start a web session

Sets a **sessionVerificationToken**, allowing the user to create a web session at another location that uses the EAI's /api/session/createSessionFromToken API.

Method

```
[POST|GET] /EAI/api/me/startWebSession
```

Content type:

```
application/json
```

Example cURL requests

```
curl -H "Authorization: Bearer access_token" -d "tokenId=1234-abcd"
https://gateway.domain.com/EAI/api/me/startWebSession
```

Request parameters

Table 7. Request parameters

Parameter name	Description
tokenId	The identifier of the token to be set in a web session environment. The **tokenId** is the value of the user_session_id sent to a protected resource by **WebSEAL**. Otherwise, **tokenId** can be an arbitrary value.

Example response

```
{
"status":"success",
"entry" : "74018cff-724d-4c37-b0a5-1aff422afb4f",
"totalCount" : 1
}
```

Returns

200: OK for success.

Delete user sessions and tokens

Deletes a user's access token and refresh token, all verification tokens if any, and deletes all active Cloud Identity Service sessions if any.

Method

```
DELETE /EAI/api/me/userSessionsAndTokens
```

Content type

```
application/json
```

Example cURL requests

```
curl -H "Authorization: Bearer access_token"  -H "Content-Type:application/
json" -X DELETE https://gateway.domain.com/EAI/api/me/userSessionsAndTokens
```

Request parameters

None.

Example response

```
{   "status" : "success",   "entry" : "VerificationToken
e1b9b0ab-4391-8319-9d0e-1d21e5x6c147 deleted successfully.
VerificationToken 6d9o18e1-2113-4o9s-91d0-3eei9072d39b deleted
successfully. Oauth access and refresh tokens deleted successfully. Deleted
web sessions successfully.",   "totalCount" : 1 }
```

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 79758 U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM® trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Printed in USA