IBM Cloud Identity Service

EAI Integration Guide

IBM

IBM Cloud Identity Service

EAI Integration Guide

IBM

Contents

Cloud Identity Service EAI integration

Integrate Cloud Identity Service with the External Authentication Interface (EAI) application.

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.

API status

Each API in this EAI Integration Guide has a verified available status. The available EAI applications are:

- Authenticate with standard HTTPS
- Authenticate with REST
- Check authentication status
- WebSEAL Logout
- Authenticate a user with social media
- Authenticate a user with social media with REST
- Transfer a session that uses SMS
- Create a web browser session
- Retrieve a web browser session

Related topics include:

- Responses to specific requests
  - Default response pages
  - Configurable responder

Authentication and authorization

Authentication uses standard HTTPS and REST.

REST stands for Representational State Transfer. A REST API is a service that handles any number of authorization requests: by users, groups, and group members, for example, for access to a server. The request types include GET, POST, PUT, and DELETE. Clients, or different types of users of the administrative portal, request access by sending requests and receiving responses that use HTTP protocols. The REST API service then responds. Requests and responses for the Cloud Identity Service are formatted as JSON objects.

A service based on REST is called a RESTful service.

Authenticate with standard HTTPS

Authenticate a user with standard HTTPS.

Method

```
POST /EAI/Login
```

Attempt to authenticate a user with a standard POST and response redirect.

Example cURL request

```
curl -X POST -d "username=gordita&password=IluvTr3ats!&redirect=https://
your.site.com/protected/index.html&reprompt=https://your.site.com/
index.html" https://gateway.domain.com/EAI/Login
```

Request parameters

Table 1. Request parameters

Parameter name	Description
username	The user's **username**.
password	The provided **password**.
redirect	The URL where the user is to be sent upon successful authentication.
reprompt	The URL where the user is to be sent upon a failed authentication attempt.

Returns

200: Customer-specific configuration might result in a 200, with **JavaScript** redirect to the location specified. If the customer-specific configuration is successful, the user is sent to the redirect URL. If the customer-specific configuration fails, the user is sent to the reprompt URL.

302: Redirect in all cases.

autherror: A query string parameter is appended to the reprompt URL. The reprompt URL indicates the error that occurred during the authentication attempt. If a failure occurs, WebSEAL redirects to the configured error page.

Authenticate with REST

Authenticate a user with REST.

Method

```
POST /EAI/api/login
```

Attempt to authenticate a user.

Example cURL request

```
curl -X POST -H "Content-Type:application/x-www-form-urlencoded" -d
"username=gordita&password=IluvTr3ats!" https://gateway.domain.com/EAI/api/
login
```

Request parameters

Table 2. Request parameters

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

Example response

```
{
"status" : "Authentication successful."
}
```

Returns

status

200: OK for success.

401: Unauthorized for every other response.

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

50X: If error on the server.

Upon success, the user's session cookies are returned as well.

Check authentication status

Check the authentication status of a user.

Method

```
GET /EAI/api/session/isAuthenticated
```

Attempt to check the authentication status of a user.

Example cURL request

```
curl https://gateway.domain.com/EAI/api/session/isAuthenticated
```

Request parameters

None.

Example response

```
{ status: "no"}
```

Returns

200: OK for all requests. The status attribute of the payload indicates whether the user is authenticated or not.

Status:

- yes: An authenticated session is in place.
- no: No authenticated session is in place.

Reset a user password

Reset a user password.

Method

```
POST /EAI/api/resetPassword
```

Before you reset a password by using this method, you must first create a passwordResetToken for the user by using the Cloud Identity Portal API (previously known as the GMA API). For more information about creating a token, see **Create a verification token for a user**.

Example cURL request

```
curl -X POST -d '{"newPassword" : "titnp4ME", "token" : "c8d28a7f-94e9-
4c60-9042-991cb14de6d4"}' -H "Content-Type: application/json"
```

Request parameters

A JSON payload that contains the following parameters.

Table 3. Request parameters

Parameter name	Description
token	The **passwordResetToken** set for the user.
newPassword	The new user password.
currentPassword	Optional. The current user password. If the current password is provided, the password is reset as if by the user. If the current password is not provided, the password is reset as if by an administrator.

Example response

```
{status: success}
```

Returns

- **200:** OK for success.
- **401:** Current password is invalid.
- **403:** Password policy violation.
- **412:** Password in history.

Session termination

You can terminate a session that uses traditional **WebSEAL**. **WebSEAL** is the only session termination method available currently.

WebSEAL, /pkmslogout, allows for removal of all **WebSEAL** cookies and clears the user's session. The user is redirected after the process completes.

WebSEAL Logout

End a user's session, clearing all **WebSEAL** session cookies and redirecting the user to a logout landing page.

The redirect can be configured either in this call, or through the Configurable Responder. If you would like to consume the redirect so that the browser does not get redirected, but the cookies are still removed, you can still call this URL from within a hidden image tag.

```
<img src="/pkmslogout" style="display:none" height="1" width="1" />
```

Method

```
GET /pkmslogout
```

Example cURL request

```
curl https://gateway.domain.com/pkmslogout?redirect=http://
gateway.domain.com
```

Request parameters

The request is for a JSON payload that contains the following attributes.

Table 4. Request parameters

Parameter name	Description
redirect	The location to send the user when the session is terminated.

Example response

```
HTTP/1.1 302 Moved Temporarily content-length: 1680 content-type:
text/html ... location: <logout location> ...
Set-Cookie: PD-ID=;
Max-Age=0;
Domain=.pb.com;
Path=/; Expires="Sun,
01-Jan-1995 01:00:00
GMT"; Secure
Set-Cookie: PD-ECC=;
Max-Age=0;
Domain=.pb.com;
Path=/; Expires="Sun,
01-Jan-1995 01:00:00
GMT"; Secure
```

Returns

status

200: OK for success.

401: Unauthorized if failed for any reason.

500: If error on the server. Upon success, user's session is terminated on **WebSEAL**.

Social media integration

User authentication with social media.

Authenticate a user with social media

Attempts to authenticate a user by calling social media and by using a traditional POST.

Method

```
POST /EAI/Login/social/{platform}
```

platform corresponds to the social media platform for authentication. Support for social media is provided by Spring Social.

Planned **provider** support: *facebook*, *google*, *qq*, *renren*, *wechat*, *weibo*, and *yahoo*.

Example cURL request

```
curl -X POST -H "Content-Type: application/x-www-form-urlencoded" -d
"token=123445&appId=com.your.site.app&redirect=https://your.site.com/
protected/

index.html&reprompt=https://your.site.com/index.html"https://
gateway.domain.com/EAI /Login/social/facebook
```

Request parameters

A JSON payload that contains the following parameters.

Table 5. Request parameters

Parameter name	Description
token	The user's access **token**.
appId	A pre-shared application identifier that allows the Cloud Identity Service to determine which API key should be used.
redirect	The URL where the user is to be sent on a successful authentication.
reprompt	The URL where the user is to be sent on a failed authentication attempt.

Returns

200: OK for success. Customer-specific configuration might result in a 200 with JavaScript redirect. On `success`, the user is sent to the redirect URL. On `failure`, the user is sent to the reprompt URL. A query string parameter, **autherror**, is appended to the reprompt URL to indicate the error that occurred during the authentication attempt.

302: Redirect in all cases.

401: Unauthorized if failed for any reason. Error messages are configurable to any wanted string and can be translated (localized) based on locale or preferred language. Contact your IBM® Delivery Lead for details on customizing this **401 unauthorized** for every other response.

On success, the user's session cookies are returned as well.

Authenticate a user with social media with REST

Attempts to authenticate a user by calling social media and the REST API.

Method

```
POST /EAI/api/login/social/{platform}
```

Platform corresponds to the social media **platform** for authentication. Support for social media is provided by Spring Social.

Planned **provider** support: *facebook*, *google*, *qq*, *renren*, *wechat*, *weibo*, and *yahoo*.

Example cURL requests

```
curl -X POST -d '{"token":"123445","appId":"com.your.site.app"}' -H
"Content-Type: application/json" https://gateway.domain.com/EAI/api/login/
social/facebook

curl -X POST -H "Content-Type: application/json" "https://
gateway.domain.com/EAI/api/login/social/yahoo?token=12345
&appId=com.your.site.app"
```

Request parameters

Content-type: **application/json**.

Table 6. Request parameters

Parameter name	Description
token	The user's access token.
appId	Shared application identifier that allows the Cloud Identity Service to determine which API key should be used.

Example response

```
{status: success}
```

Returns

200: OK for success.

401: Unauthorized if failed for any reason.

403: Forbidden if the user's social media account is incomplete and a user profile cannot be created.

500: If error on the server.

On success, user's session cookies are returned as well.

Session management

Creating, transferring, and retrieving sessions.

Transfer a session by using SMS

Once a user is already authenticated, creates a session in a new Domain Name Service (DNS) domain.

A valid session must exist. The environment must be configured for Short Message Service (SMS).

Method

```
POST /EAI/api/session/resumeSession
```

Example cURL request

```
curl -X POST -d "sessionID=123456&redirect=https://your.site.com/
protectedResource" https://gateway.domain.com/EAI/api/session/resumeSession
```

Request parameters

A request includes a JSON payload that contains the following parameters.

Table 7. Request parameters

Parameter name	Description
`sessionID`	The user's SMS session ID.
`redirect`	The URL to send the user after you resume the session.

Returns

`200:` Customer-specific configuration might result in a **`200`** with **`JavaScript:`** redirect to the location specified.

`302:` Redirect.

`Failure:` If a failure occurs, **`WebSEAL`** redirects to the configured error page.

On success, the user's session cookies are returned as well.

Create a web browser session

Create a web browser session from the session verification token.

Create a web session for the current domain. Creating a web session requires that the user is already authenticated through the **`GmaApi`**, and that a session verification token exists for the user.

Method

```
[GET | POST]
```

```
/EAI/api/session/createSessionFromToken
```

Example cURL request

```
curl -X POST -d "token=7470f51f-2f5f-470e-8bea-402ae678bafb
&redirect=https://your.site.com/protectedResource" https://
gateway.domain.com/EAI/api/session/createSessionFromToken
```

Request parameters

A JSON payload that contains the following parameters.

Table 8. Request parameters

Parameter name	Description
`token`	Optional. The user's **`sessionVerificationToken`** value, if you are creating a session for a user who authenticated through the **`GmaApi`**.
`redirect`	Optional. The URL to send the user after you resume the session.

Returns

`200:`. Customer-specific configuration might result in a **`200`** with **`JavaScript:`** redirect to the location specified.

`302:` Redirect.

`LSG-SESSION-ID:` LSG-SESSION-ID cookie that represents the user's SMS session ID handle.

`WebSEAL:` Session cookies for the user's **`WebSEAL`** session (PD-S-SESSION-ID). If a failure occurs, **`WebSEAL`** redirects to the configured error page.

Example sequence of commands

1. Authenticate the user by invoking the **`username`** and **`password`** form data:
   ```
   curl -X POST -H "Content-Type:application/x-www-form-urlencoded" -H
   "Authorization: Basic ZWFpLWNsaWVudDo=" -d "grant_type=password
   &username=userid&password=user_password" https://gateway.domain.com/EAI/
   oauth/token
   ```
   Where the **`userid`** value is the user's **`gtwyPrincipalName`** attribute value, and **`user_password`** is the user's password attribute value.
2. Make a GET request, placing the **`OAuthbearer`** token, returned from step 1, into the **`Authentication Header`**:
   ```
   curl -H "Authorization: Bearer 56d512a9-4fa34ac6-a72a-76d66ed84d21"
   https://gateway.do main.com/EAI/api/me/startWebSession
   ```
3. Make a GET request, placing the **`entry`** value returned into the **`query string`**. Use a field name of **`token`** for this value:
   ```
   curl https://gateway.domain.com/EAI/api/session/
   createSessionFromToken?token =4683caf7-c937-4edc-8105-bfa075f4d6ff -v
   ```

The return here includes a **`PD-S-SESSION-ID`** that can be used for **`getSession`**.

Note: There are differences in syntax between the operating systems: Windows, Linux, and Mac. For instance, in Windows, you do not need quotation marks.

Retrieve a web browser session

Retrieves the Short Message Service (SMS) cookie for the user's session.

Requires that the user is authenticated through the External Authentication Interface (EAI).

Method

```
[POST] /EAI/api/session/getSession
```

Example cURL request

```
curl https://gateway.domain.com/EAI/api/session/getSession
```

Request parameters

None.

Returns

200: OK for success.

LSG-SESSION-ID: LSG-SESSION-ID cookie that represents the user's SMS session ID handle.

Responses to specific requests

The EAI determines the right page to serve in response to specific requests.

Default response pages

For a set of special operations, the EAI has a component that is called the **Responder** that determines the right page to serve.

For example, assume that a user attempts to access a protected resource, but needs to authenticate first. **WebSEAL** sends a request to the **Responder** indicating that the user needs to log in. The **Responder** then serves the login page. These login pages, and other authentication pages, are configurable for each domain, and the services team can supply authentication page templates.

The authentication pages are:
- **Login**
- **Logout**
- **Password change** (for expired passwords)
- **Successful password change**, presented only when a user does not have another resource.
- **Error**, for **WebSEAL** server errors only.
- **Step-up**, for step-up authentication only.
- **Help**, for operations that **WebSEAL** cannot service.

Configurable responder

Instead of using the default pages, the **Responder** can also be configured to redirect to a location of the customer's choosing.

For each operation supported, the **Responder** can be configured to analyze the incoming referring URL, determine whether that URL matches a particular pattern, and redirect the browser to a configured location.

See the following Operation destinations table:

Table 9. Operation destinations

Operation	Referrer	Destination
login	`https://*.foo.com`	`https://www.foo.com/login`
logout	`https://*.foo.com`	`https://www.foo.com/logout`

Table 9. Operation destinations (continued)

Operation	Referrer	Destination
password	`https://*.foo.com`	`https://www.foo.com/selfservice`
postlogin	`https://*.foo.com`	`https://www.foo.com/postlogin`

As described in the table, the **`Referrer URL`** can be wild-carded based on regular expressions. If this wildcard feature is enabled, and a particular **`operation`** and **`referrer`** matches a destination, then the browser is sent to the destination URL. The destination URL includes the **`Referrer URL`** as a **`query string`** parameter.

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.

Trademarks

Printed in USA