March 28, 2025

Authentication and Pairing Process V2

Tamar Kariv

FOR ALIGN TECHNOLOGY, INC. USE ONLY | Created By iTero Professional Services

Table of Contents

1 Revision History

Revision Number	Revision Date	Nature of Revision
1	09 Jun 2020	Initial version
2	13 Sep 2020	4.3-Company Pairing 4.4-Scan Notifications Update scan notifications configuration and usage
3	11 Nov 2020	4.1.1-Authorization code grant request Add first time log in note
4	06 Jan 2021	Scan notification update
5	05 May 2021	4.3.1-Pair request Warning for multiple patients with the same chart number
6	23 Jun 2021	4.2.1 - Get related accounts list Criteria for pairing
7	29 Jun 2021	4.1 - Refresh token expiry
8	01 Nov 2021	4.1.2 - non SCT is no longer in use
9	29 Dec 2022	Minor changes
10	19 Jan 2023	Branch V1 to V2
11	14 Mar 2023	4.1.2 - Update response description 4.1.3 - Update response description 4.1.5 - Update response description 4.2.1 - Update response description 4.3.1 - Update response description 4.3.2 - Update response description
12	20 Jul 2023	4.1.2 - Auth code expiration 4.1.5 - When to revoke token
13	12 Oct 2023	3.1 - Required Customer Information 4.1.1 - Authorization code grant request
14	13 Feb 2024	Expand concepts, changes
15	11 Aug 2024	4.4.1 - Add error code 409

Revision Number	Revision Date	Nature of Revision
16	27 Sep 2024	Discovery, small changes
17	18 Oct 2024	Minor changes
18	12 Nov 2024	Discovery API URLs update
19	25 Mar 2025	Add error codes 409/504

2 Introduction

This document describes the Authentication and Pairing process to a 3rd party software, using Align Technologies APIs in sandbox and production environments.

The Authentication & Pairing process described in this document is the first step in 3rd party software integration.

Upon completion of this step, the caller receives an access token, to be used as authorization key for all Align generic APIs.

This document defines version 2 (V2) of the authentication and pairing process.

2.1 Integration Environments

iTero provides 3rd party partners the ability to train themselves with a staging (Sandbox) environment.

Once ready to move to production, contact iTero Support iTeroAPISupport@aligntech.com to enable the process and receive the following:

- Production URL.
- Credentials

2.2 Contacting Support

Please contact iTero Support at iTeroAPISupport@aligntech.com for API related issues.

2.3 Definitions, Acronyms, and Abbreviations

Myitero.com	iTero portal to manage account and cases
Production environment	Live environment
Sandbox/staging environment	Environment that enables testing applications outside the production environment
Redirect URL	URL to which the response should be returned
Access Token	Security credentials key for login session and user identification
Callback URL	URL to which a notification is sent upon scan completion

3 Authentication & Pairing Overview

Authentication and pairing are required to be completed before using Align APIs. Upon completion, the 3rd party software can be integrated with iTero software.

Align authentication API complies with OAuth 2 Authorization Code Grant. For more information, please visit:

https://www.oauth.com/oauth2-servers/server-side-apps/authorization-code/

3.1 Required Customer Information

To create an account in the authentication server, a company must provide Align the following information:

- Company Name
- Redirect URL: After a user successfully authorizes an application, the authorization server will redirect the user back to this URL.
    - This URL should be whitelisted on iTero side
    - It should use https
    - iTero doesn't support URLs with "#' in them

3.2 Provided Authentication Information

The following will be provided by iTero support:

- The application ID that is assigned to your app <ClientID> -
- The application secret assigned to your app <ClientSecret>
- MyiTero.com Credentials
- Server URL for authorization process – shown as {{oauthBase}}
- Server URL for pairing process – shown as {{baseUrl}}
- Align Login Page – shown as {{loginPage}}

Upon project completion, a 3rd party partner will receive 2 sets of ClientID and secret. One for Sandbox, and one for Production.

4 Authentication and Pairing Process

The Authentication and Pairing process consists of three steps:

1. Authentication process - concludes with access and refresh tokens
2. Discovery - returns regional URL's to use for a given account
3. Get available companies for pairing using returned token

4. Pairing to a chosen account - concludes with pairing to a selected account and new access and refresh tokens to use iTero APIs

The authentication process can work for a sever to server configuration or sever to client configuration. The token (after pairing) is associated with an account (and not with an individual user) such that In case of server to server configuration, the token can be securely stored by the 3rd party and shared by the callers. For client applications, the token will have to be stored at the client.

The advantages to use this mechanism are:

- Higher level of security
- Pairing is based on account and not on user. Account's users may change with time.

In case a server is not available to the account, both sets of tokens can be managed on the client side, however this option is less preferable.

Please refer to the chart at the end of this document for a graphical view of the process.

Note:

Note

- All users of the account will share the same tokens after pairing.
- Full authentication process is required when refresh token expires (valid for one year). As long as refresh token is valid and securely stored it is not necessary to repeat this process (however, there is no harm doing so).

Once the pairing process is completed, the 3rd party will be able to use Align APIs for the paired company.

4.1 Authentication Process

Authentication process returns Access and Refresh Tokens that are later used for all API calls.

4.1.1 Authorization code grant request

The 3rd party software should call the authorization page URL, based on the following:

- {{oauthBase}}
- Client_ID received from iTero Support
- State - Optional 3rd Party choice random string
- Redirect URL
- {{loginPage}}
- Scope - openid

Add the client ID received from iTero Support, and the redirect URL.

{{oauthBase}}/oauth2/authorize?client_id=<Client_ID>&response_type=code&state=<string>&redirect_uri=<Redirect URL>&loginPage={{loginPage}}&scope=openid

A Login page will open; enter your MyiTero.com credentials.

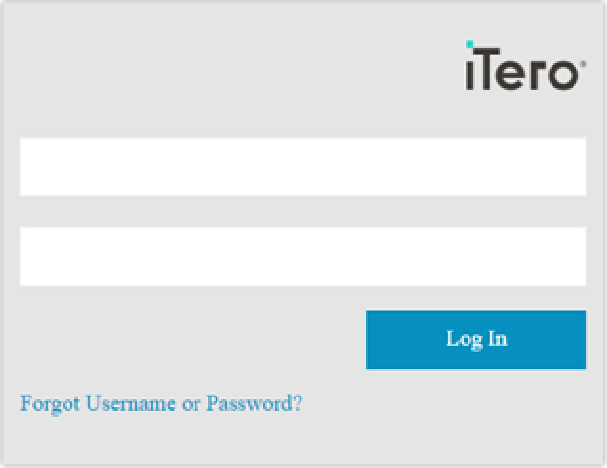

Figure 1: Enter your iTero credentials

The user will be redirected to the following address:

<RedirectURL>?**code**=<AuthorizationCode>&**state**=<string>&session_state=<string>

The 'code' parameter (auth code) in the URL is used in the next step, alongside the state parameter for added security.

4.1.1.1 Using State Parameter

For extra security, a 3rd party partner can use the state parameter with a random value that is sent with the authorization request and can compare it to the returned state parameter. Code values should be identical.

4.1.2 Exchange the authorization code for an access token

To exchange the authorization code for an access token, POST the following call using only raw format (This endpoint doesn't accept JSON):

{{oauthBase}}/oauth2/token

Headers

‘Content-Type’: ‘application/x-www-form-urlencoded'

Request body

code=<AuthorizationCode>

grant_type=authorization_code

client_id=<ClientID>

client_secret=<ClientSecret>

redirect_uri=<Redirect URL>

Provide the authorization code that was returned in the previous step.

Note

1. Auth code should be used immediately. It will expire within a few minutes.
2. The returned Auth code is single-use only.

Response

Response codes are:

HTTP Status Code	Description	Scenario	Recommendation
200	OK	/	/
400	Bad request	1. Wrong code 2. Missing code	Double check if call used the correct authorization code
401	Unauthorized error	The client is not authorized to perform this operation	Check if call is using the correct credentials
500	Internal Server error	/	A retry mechanism is recommend for this error message.
504	Gateway Timeout	Timed out	Retry the request after a brief wait time (1000ms)

In case of returned code of 200, the following response will be returned:

The call returns access and refresh token with 1-hour expiry, and a refresh token that lasts for 1 year.

See response example below:

```
{
    "access_token": access token,
    "refresh_token":refresh token
    “scope”: “openid”
    “id_token”: token ID
    "token_type":"Bearer",
    "expires_in":3600
}
```

Below is a table describing the information that has been returned in the Data section of the response:

Field	Type	Description
access_token	String	Access token to be used in APIs as a bearer token. Its size is up to 8K.
refresh_token	String	Refresh token to be used when access token is expired. Size is up to 40 characters.
scope	String	openid
Id_token	String	Cached user profile
token_type	String	Type of token: Bearer
expires_in	Number	Number of seconds left before token’s expiration. Initial value is 3600 (one hour)

4.1.3 Refresh token request

Callers should exchange a refresh token for an access token when the access token expires.

This allows keeping a valid access token without further interaction with the user.

To refresh an existing token, POST the following call:

{{oauthBase}}/oauth2/token

Headers

'Content-Type’: ‘application/x-www-form-urlencoded'

Request body

refresh_token={{refresh_token}}&

grant_type=refresh_token&

client_id=<ClientID>&

client_secret=<ClientSecret>

Response

Response codes are:

HTTP Status Code	Description	Scenario	Recommendation
200	OK	/	/
401	Unauthorized error	1. Wrong refresh token 2. Missing refresh token	Double check that call is using the correct refresh token
500	Internal Server error	/	A retry mechanism is recommend for this error message.
504	Gateway Timeout	Timed out	Retry the request after a brief wait time (1000ms)

In case of returned code of 200, the following response will be returned

The response includes a new access token with 1-hour expiry, while the refresh token remains the same.

```
{
 "access_token": string,
 "refresh_token": string,
 "scope": string,
 "token_type": string,
 "expires_in": number
}
```

Response keys are identical to the keys in the first-time token request.

4.1.4 Refresh Token Expiry

Refresh token can be used for one year. Upon expiration, the 3rd party app should make sure to trigger a new authentication request as described in 4.1.

Coupling "Refresh Token" call with a call to "Discovery" endpoint is recommended.
In rare cases of changes in the domain or URL, the change will be automatically adopted by the 3rd party app, once the refresh is called.
Refer to discovery API below.

4.1.5 Revoke token request

The Token Revocation extension defines a mechanism for partners to indicate to the authorization server that an access/refresh token is no longer needed. This is used to enable a "log out" feature at the client side, allowing the authorization server to clean up any security credentials associated with the authorization.

Note

- Token revocation will deprecate the refresh token. The next time a token will be requested, a new refresh token will be returned.
- Do not revoke the token unless:
  - The current token is no longer required
  - The current token was compromised
  - It is decided by design to proactively generate a new refresh token

To revoke an existing token, POST the following call:

{{oauthBase}}/oauth2/revoke

Headers

'Content-Type': ‘application/x-www-form-urlencoded''

Request body

Options response codes are:

HTTP Status Code	Description	Scenario	Recommendation
200	OK	/	/
401	Unauthorized error	1. Wrong token 2. Missing token	Double check that call is using the correct token
500	Internal Server error	/	A retry mechanism is recommend for this error message.
504	Gateway Timeout	Timed out	Retry the request after a brief wait time (1000ms)

In case of returned code of 200, the following response will be returned

```
token={{access_token or refresh_token}}

token_type_hint={{'access_token' or 'refresh_token'}}

client_id=<ClientID>

client_secret=<ClientSecret>
```

4.2 Discovery - Regional URL's for API Calls

4.2.1 Introduction

iTero services are regional, meaning that a caller should access his region in order to retrieve its account data. A highly recommended practice is to use endpoint "Discovery".

4.2.1.1 Benefits

1. iTero APIs are based on regional location. Each account is paired in a specific region and the calls are expected to be routed to that same region. DNS geo-based routing does not always route the caller to the correct region. With the regional base URL, the calls are guaranteed to go to the correct region.

2. End users can work from outside their permanent address. Caller app will be routed to the correct region in cases where it is outside its intended geographical region, such as a business trip or other scenarios.

3. The returned URL is regional. This makes sure that the caller access the correct region. This mitigates call failures due to wrong calls routing, which sometime happen by the different ISPs (Internet Service Providers).

4.2.1.2 When to use

A call to this endpoint should happen in two point in time:

1. After the initial authentication is completed and before pairing

2. With every call to refresh token

The first call should use the first generated token before pairing. This will ensure to use the correct regional URL, based on the location of the caller in iTero back office system (not geolocation).

The second case is to ensure to accommodate any future changes in the URL or domain (not likely to happen).

A simple use case would be:

A call returns 403 (Because access token has expired)
Make a call to Refresh Token
Make a call to "Discovery"
Resume initial call to the iTero API and continue normal usage.

> iTero operates in four regions: North America, Europe, Asia Pacific and China
> In sandbox, there is one region only so the returned domain will be the same..

4.2.1.3 How to use

The "Discovery" endpoint returns the regional base URL that must be used for all of future API calls.

The future calls will follow the overall structure: Domain/Path/endpoint

The first call to discovery is made with the {{baseUrl}}, but subsequent calls must be made with {{Domain}}/{{Path}} returned by discovery

Which endpoints do not use Discovery?

Authorization endpoints:

- Authorize
- Authorization Code
- Refresh Token
- Revoke Token

Which token should I use for Discovery?

The first call to discovery, since the user is not paired to any company, should be done with the access token.

After pairing, "Discovery" should be called with the pair access token

4.2.2 Discovery Structure

Send a GET call to get the relevant Domain and Path of the iTero APIs.

First call:{{baseUrl}}/api/third-party/v2/api-discovery-by-name-and-version?discoveryName=third-party&version=2

Subsequent calls:
{{Domain}}/{{Path}}/api-discovery-by-name-and-version?discoveryName=third-party&version=2

Header

Authorization: The access token received after the initial authentication step is provided as a Bearer Token authorization type.

Parameters

Call parameters must include the following mandatory fields and exact values:

Parameter	Value
discoveryName	third-party
version	2

Response

HTTP Status Code	Description	Scenario	Recommendation
200	False OK	Empty Array	An empty array can mean one of two things: 1. You've not yet requested access to the "Discovery". Therefore, your request will not be processed 2. The parameter values are incorrect, and since the resource does not exist, you get back an empty response.
200	OK	One entry in array	Correct behavior
400	Bad request	Wrong asset format	Double check the parameter fields and values
401	Unauthorized error	1. Wrong token 2. Missing token	Double check that call is using the first access token (Returned after Authorization Code endpoint), or, refresh the access token for a new one using 'Refresh Token' endpoint.
500	Internal Server error	/	A retry mechanism is recommend for this error message.
504	Gateway Timeout	Timed out	Retry the request after a brief wait time (1000ms)

Return Body General Structure

```
{
    "APIs": [
        {
            "Name": "third-party",
            "Version": "2",
            "Path": "/itero-gen-api/v2",
            "Domain": "regional URL"
        }
    ]
}
```

Description of the returned body:

Fields	Description	Notes
Name	Name of the service	All 3rd parties consuming iTero Generic API will get 'third-party'
Version	Service version	3rd parties consuming iTero Generic API will receive the API version they use (2 in most of the cases)
Path	The path for all APIs	All API calls should use the Domain/Path/endpoint
Domain	The Domain of all APIs	

4.2.3 Get Available Companies for Pairing

Send a GET call to get a list of all related accounts of the current user:

Discovery	{{Domain}}/{{Path}}/related-accounts
Legacy	{{baseUrl}}/api/third-party/v2/related-accounts

Header

Authorization: The access token received at the end of authentication process is provided as a Bearer Token authorization type.

Response

Options response codes are:

HTTP Status Code	Description	Scenario	Recommendation
200	OK	/	/
401	Unauthorized error	1. Wrong token	Double check that call is using the correct access token

HTTP Status Code	Description	Scenario	Recommendation
		2. Missing token	
500	Internal Server error	/	A retry mechanism is recommend for this error message.
504	Gateway Timeout	Timed out	Retry the request after a brief wait time (1000ms)

In case of returned code of 200, the following response will be returned

This GET request returns a JSON format containing the related companies' IDs, names and addresses.

The listed accounts meet the following criteria:

1. The user is related to these accounts
2. The account is enabled for that specific integration.

Both are configured in Align system. If one of the conditions is not met, the account will not be listed and cannot be paired to by the user.

```
{
 "Data": [
 {
  "AccountId ": number,
  "AccountName ": string,
  "AccountAddress ": string
       "IsPaired": true/false
 }
 ...
       ],
 "Status"://Success, Failure
 "Errors": [
 ]//array of errors if Status = Failure
}
```

Below is a table describing the information that has been returned in the Data section of the response:

Field	Type	Description
AccountId	String	ID of related account
AccountName	String	Name of related account

Field	Type	Description
AccountAddress	String	Address of related account
IsPaired	Boolean	True if the account is already paired with the user. To unset this field call UnPair endpoint

4.3 Company Pairing

Pairing process pairs the user to a selected account and returns a new set of access and refresh tokens. In case the user is already paired to the company, calling pair request will return the same refresh token that was generated before.

4.3.1 Pair request

To pair the user with the selected company from the list above, use the following PUT call:

Discovery	{{Domain}}/{{Path}}/pair-account
Legacy	{{baseUrl}}/api/third-party/v2/pair-account

Headers

'Content-Type’: ‘application/json'

Authorization: The access token received at the end of authentication process as a Bearer Token authorization type.

Request body

```
{
  "AccountId": number,
  "CallbackUrl": string
}
```

Request body includes:

- AccountId of the account to pair to
- CallbackUrl is a valid URL to which push notification will be sent. An order that is ready will trigger a notification sent to the the notification URL.

Response

Options response codes are:

HTTP Status Code	Description	Scenario	Recommendation
200	OK	/	/
400	Bad request	Wrong companyid	Double check if call uses the wrong companyid
401	Unauthorized error	1. Wrong token 2. Missing token	Double check if call is using the correct access token
409	Conflict	1. A paired account is trying to pair with a different 3rd party app	This is applicable for labs only. Lab accounts cannot have multiple pairing for 3rd party applications. In this case the lab should first unpair from the old integration and then pair to the new integration.
500	Internal Server error	/	A retry mechanism is recommend for this error message.
504	Gateway Timeout	Timed out	Retry the request after a brief wait time (2000ms)

In case of returned code of 200, the following response will be returned

This PUT request returns a JSON format containing the second access and refresh tokens, and time left for expiration.It is important to note that from this point on, these set of tokens must be used. To maintain these tokens, the same refresh and revoke endpoints mentioned before apply.

```
{
 "Data":{
    "OAuthResponse":{
  "access_token":string,
  "refresh_token":string,
  "scope":string
  "token_type":string,
  "expires_in":number
     },
 "CompanyId":number
 },
 "Status":1
}
```

Check Patients Information Coherence (Applies only for DPMS solutions)

Pairing request will return a warning in case there are multiple patients in the system with the same chart number or patient missing chart number. Please refer to the iTero API - DPMS V2.doc for more details.

```
 "Data": {
        "OAuthResponse": {
            "access_token": string
            "refresh_token": string,
            "scope": string,
            "token_type": string,
            "expires_in": number
        },
        "Warnings": [
            "Empty or duplicate patient identifiers (chart#) exists in iTero
patient database"
        ],
        "CompanyId": number
    },
    "Status": 1
}
```

4.3.2 Unpair account request

To unpair a user from an account call this DELETE request:

Discovery	{{Domain}}/{{Path}}/unpair-account
Legacy	{{baseUrl}}/api/third-party/v2/unpair-account

Headers

The access token received at the end of pair request as a Bearer Token authorization type

Request body

Format is raw/Json

```
{
  "AccountId": number
}
```

Body includes AccountId of paired account.

Note that since the pairing is based on account, un-pair will affect all paired users to this account, and all will be un-paired. Also, the callback URL will be deleted for the account.

> There is no need to unpair the account unless one wants to discontinue the service or upon a few special cases.
> Therefore, it is recommended to implement an unpair option in the 3rd party application. This option should be available upon permission to the designated personnel.

Response

Options response codes are:

HTTP Status Code	Description	Scenario	Recommendation
200	OK	/	/
400	Bad request	Wrong companyid	Double check if call used the wrong companyid
401	Unauthorized error	1. Wrong token 2. Miss token	Double check if call uses the correct pairing access token
500	Internal Server error	/	A retry mechanism is recommend for this error message.
504	Gateway Timeout	Timed out	Retry the request after a brief wait time (1000ms)

4.4 Scan Notifications

Scan notifications is a mechanism that implements a webhook to notify the account that a new scan is ready. This mechanism can be used in exchange or in addition to the polling mechanism of Get Orders. Please check for more details in the iTero API - DPMS V2.doc (for clinics) or in iTero API - Lab Connector V2.doc (for labs).

Note:

- A third party can choose to register an integration-wide URL with Align to which notifications will be sent. This URL will apply to all accounts that are associated with this integration.

- If no URL is provided by the third party, it can be provided during paring request. This will apply to the account that is paired.
- If both URLs are provided, the notification will be sent to the third party's integration-wide URL and not to the one provided in the pairing call.
- If not URL is provided, a notification will not be sent.
- If a third party is not using an integration-wide URL and wishes to change the URL listed in the Pair endpoint, the account must unpair, and then repair with the new callback URL.

4.5 Authentication & Pairing Process Summary

The following tables describe the authentication and pairing APIs in tabular format.

4.5.1 Authentication and Pairing process

4.5.1.1 Authorization code grant request:

Done by:	Third-party user
Authentication:	MyiTero credentials
URL:	{{oauthBase}}/oauth2/authorize?client_id=<Client_ID>&response_type=code&state=447&redirect_uri=<Redircet URL>&loginPage={{LoginPage}}
Parameters:	Client ID, Redirect URL
Returns:	<authorizationcode>
Description/Comments:	This is the first pairing step

4.5.1.2 Exchange the authorization code for an access token:

Done by:	Third-party user
URL:	{{oauthBase}}/oauth2/token
Authentication:	<authorizationcode>
Returns:	Access Token, Refresh Token
Request Type	POST

Note: Access token is valid for 1 hour.

4.5.1.3 Refresh Token:

Done by:	Third-party user
URL:	{{oauthBase}}/oauth2/token
Parameters:	Refresh Token, Client ID, Client Secret
Returns:	New access token

4.5.1.4 Revoke Token:

Done by:	Third-party user
URL:	{{oauthBase}}/oauth2/revoke
Parameters:	Access Token, Client ID, Client Secret

4.5.2 Discovery

Done by:	Third-party user
Authentication:	Access Token
URL:	**First call:**{{baseUrl}}/api/third-party/v2/api-discovery-by-name-and-version?... **Subsequent calls:** {{Domain}}/{{Path}}/api-discovery-by-name-and-version?
Returns:	Regional URL

4.5.3 Get related accounts for pairing

Done by:	Third-party user
Authentication:	Access Token
URL:	**Discovery** {{Domain}}/{{Path}}/related-accounts **Legacy** {{baseUrl}}/api/third-party/v2/related-accounts/
Returns:	Practice details

4.5.4 Account Pairing

Done by:	Third-party user
Authentication:	Access Token
URL:	**Discovery** {{Domain}}/{{Path}}/pair-account **Legacy** {{baseUrl}}/api/third-party/v2/pair-account/
Parameters	CompanyID, Callback URL
Parameter Format:	JSON
Returns	Access Token, Refresh Token

4.5.5 Account Un-Pairing

Done by:	Third-party user
Authentication:	Access Token

URL:	**Discovery**	{{Domain}}/{{Path}}/unpair-account
	Legacy	{{baseUrl}}/api/third-party/v2/unpair-account/
Parameters	CompanyID	
Parameter Format:	JSON	

4.6 Authentication & Pairing Sequence Diagram

The following table shows the authentication and pairing APIs integration diagram.

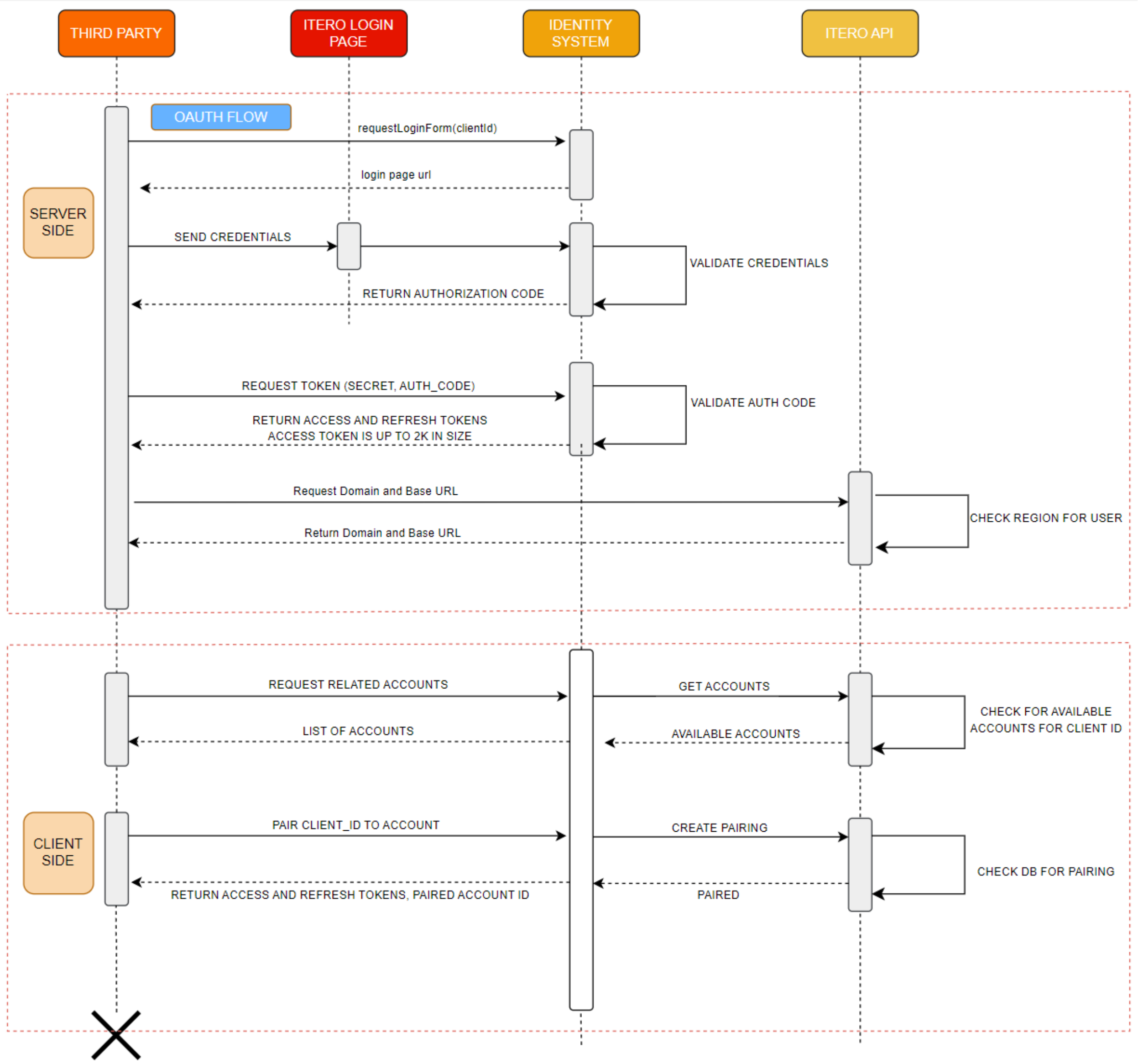
THIRD PARTY
ITERO LOGIN PAGE
IDENTITY SYSTEM
ITERO API
OAUTH FLOW
requestLoginForm(clientId)
login page url
SERVER SIDE
SEND CREDENTIALS
VALIDATE CREDENTIALS
RETURN AUTHORIZATION CODE
REQUEST TOKEN (SECRET, AUTH_CODE)
VALIDATE AUTH CODE
RETURN ACCESS AND REFRESH TOKENS
ACCESS TOKEN IS UP TO 2K IN SIZE
Request Domain and Base URL
CHECK REGION FOR USER
Return Domain and Base URL
REQUEST RELATED ACCOUNTS
GET ACCOUNTS
CHECK FOR AVAILABLE ACCOUNTS FOR CLIENT ID
LIST OF ACCOUNTS
AVAILABLE ACCOUNTS
PAIR CLIENT_ID TO ACCOUNT
CREATE PAIRING
CLIENT SIDE
CHECK DB FOR PAIRING
RETURN ACCESS AND REFRESH TOKENS, PAIRED ACCOUNT ID
PAIRED

Figure 3: Authentication & Pairing Sequence Diagram