

CamDigiKey Integration

RUNNING AND USING CAMDIGIKEY CLIENT

CAMDIX TEAM

Version	Description
1.0.0	<ul style="list-style-type: none"> • Get QR code for scanning. • Get user access token. • Get organization access token. • Renew access token. • Verify access token. • Log out access token.
1.1.0	<ul style="list-style-type: none"> • Lookup user account. • Verify user account token.
1.1.1	<ul style="list-style-type: none"> • Update library version

Introduction

CamDigiKey is a single sign on authentication system in CamDX ecosystem. It provides convenient and optimal authentication process. Basically, users need to register themselves to the application by providing their identity documents such as ID card or passport. Once the user's account is approved, they can use CamDigiKey mobile application to login to any CamDigiKey partners' portals. This document will focus on how to integrate with CamDigiKey on portal.

Register Account

Before integration with CamDigiKey, partners need to contact us for account registration. In the registration process, partners must provide only their system's domain name. Once, the account has been created, partners will receive information:

- Client access credential parameters
 - Domain
 - Client ID
 - HMAC key
 - IV Param key
 - Secret key
- TLS key store and trust store

How does it work?

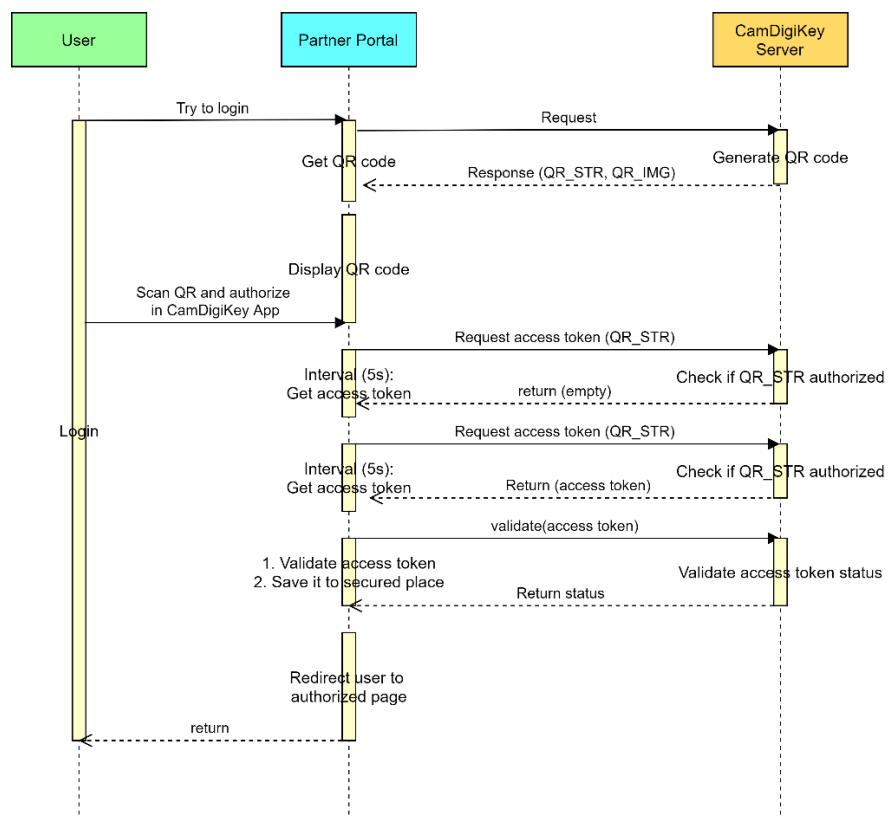


Figure 1. Login process

The figure 1 shows the processes behind when user tries to login with CamDigiKey:

1. Portal gets QR code from CamDigiKey server
2. Portal shows QR code to user
3. User scans QR code and authorizes within CamDigiKey mobile application
4. Meanwhile, portal does the interval checkup (5 seconds) the access token from CamDigikey server
5. After receiving access token successfully, portal needs to validate, saves, and extracts payload of the access token. The extracted payload can be used to map with local user of the portal system
6. Finally, portal redirects user to login successful page

CamDigiKey Client APP (Recommended)

CamDigiKey client application is a java application that build on top of CamDigiKey client library. The client application provides API access (REST) without any authentication. **For security concern, please install the client application for local system access only.** Please follow the steps to config app credential:

1. Edit src/main/resources/application.properties:
 - a. add client credential properties:
 - i. camdigikey.client-id = [client_id] (provided by us)
 - ii. camdigikey.hmac-key = [hmac_key] (provided by us)
 - iii. camdigikey.aes-secret-key = [secret_key] (provided by us)
 - iv. camdigikey.aes-iv-params = [iv_params] (provided by us)
 - v. camdigikey.client-domain = [your_application_domain]
 - vi. camdigikey.server-based-url= [CamDigiKey Url] (DEV: <https://mobile-id.demo.camdx.io:8443>, Prod: <https://camdigikey.camdx.gov.kh:8443>)
 - b. add client connection key store file configuration:
 - i. camdigikey.client-keystore-file = [path to client key store file]
 - ii. camdigikey.client-keystore-file-password = [password of key store file]
 - iii. camdigikey.client-keystore-client-key-entry-name= [client_key_entry_name]
 - iv. camdigikey.client-keystore-client-key-entry-password= [client_key_entry_password]
 - v. camdigikey.client-trust-store-file = [path of the trust store]
 - vi. camdigikey.client-trust-store-file-password = [trust_password]
2. Rebuild and run app
 - a. Go to app folder
 - b. Run: mvn package
 - c. Run: java -jar target/application-1.0.0.jar

CamDigiKey client application provide APIs:

1. Get QR code

URL	/qr-code
Method	GET
Response	Object: <ul style="list-style-type: none">• authCode: String (base 64)• qrImage: String (base 64) Portal system must render QR image in the img tag. For instance,

2. Get user access token

URL	/access-token
Method	GET
Request	Object: <ul style="list-style-type: none">• qrCode: String (base 64)
Response	Object: <ul style="list-style-type: none">• accessToken: String (base64)

3. Validate access token

URL	/
Method	POST
Request	Object: <ul style="list-style-type: none">• token: String
Response	Object: <ul style="list-style-type: none">• payload can be:<ul style="list-style-type: none">○ user key payload: object<ul style="list-style-type: none">▪ first_name_en: String▪ last_name_en: String▪ first_name_kh: String▪ last_name_kh: String▪ gender: String (M: Male, F: Female)▪ iss: String (CamDX-CamDigiKey)▪ type: String (USER)▪ personal_code: String (user id card number or passport number)▪ mobile_phone: String▪ email: String▪ exp: timestamp (expired date)▪ nbf: timestamp (not before date)▪ iat: timestamp (issued at date)▪ jti: String (JWT ID)○ organization key payload: object<ul style="list-style-type: none">▪ domain: String▪ iss: String (CamDX-CamDigiKey)▪ type: String (ORGANIZATION)

	<ul style="list-style-type: none"> ▪ exp: timestamp (expired date) ▪ iat: timestamp (issued at date) ▪ jti: String (JWT ID) • is_valid: boolean
--	---

4. Refresh user access token

Note: refresh token API should be called before access token expired.

URL	/refresh-user-access-token
Method	GET
Request	Object: <ul style="list-style-type: none"> • accessToken: String (base 64)
Response	Object: <ul style="list-style-type: none"> • accessToken: String (base64)

5. Get organization access token

Note: CamDigiKey also provides organization level access token which is used for API interaction between CamDX participants.

URL	/organization-access-token
Method	GET
Response	Object: <ul style="list-style-type: none"> • accessToken: String (base64)

6. Logout access token

URL	/logout-access-token
Method	GET
Request	Object: <ul style="list-style-type: none"> • accessToken: String (base 64)
Response	Object: <ul style="list-style-type: none"> • jwtId: String • status: int (0: logout)

7. Lookup CamDigiKey user account

URL	/lookup-user-profile
Method	GET
Request	Object: <ul style="list-style-type: none"> • accessToken: String (base 64) • peronalCode: String
Response	Object: <ul style="list-style-type: none"> • personal_code: String • surname_kh: String • given_name_kh: String • surname_en: String

	<ul style="list-style-type: none"> • given_name_en: String • mobile_phone_number: String • email_address: String • expired_date: timestamp • account_token: String
--	--

8. Verify user account token

URL	/verify-user-profile
Method	GET
Request	Object: <ul style="list-style-type: none"> • accountToken: String (base 64)
Response	Object: <ul style="list-style-type: none"> • personal_code: String • surname_kh: String • given_name_kh: String • surname_en: String • given_name_en: String • mobile_phone_number: String • email_address: String

CamDigiKey Client Library (Java only)

This is a good option for partner system that has built on Java. To use CamDigiKey client library in your project, please follow the steps below:

1. Register this Jar file in local Maven
2. Add dependencies libraries in POM.xml

```

<dependency>
  <groupId>kh.gov.camdx.camdigkey.client.application</groupId>
  <artifactId>camdigikey_client</artifactId>
  <version>1.1.1</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-webflux</artifactId>
</dependency>
<dependency>
  <groupId>org.bouncycastle</groupId>
  <artifactId>bcprov-jdk15on</artifactId>
  <version>1.70</version>
</dependency>

```

3. Add library properties

a. add client credential properties:

- i. camdigikey.client-id = [client_id] (provided by us)
- ii. camdigikey.hmac-key = [hmac_key] (provided by us)
- iii. camdigikey.aes-secret-key = [secret_key] (provided by us)
- iv. camdigikey.aes-iv-params = [iv_params] (provided by us)
- v. camdigikey.client-domain = [your_application_domain]
- vi. camdigikey.server-based-url= [CamDigiKey Url] (DEV: <https://mobile-id.demo.camdx.io:8443>, Prod: <https://camdigikey.camdx.gov.kh:8443>)

b. add client connection key store file configuration:

- i. camdigikey.client-keystore-file = [path to client key store file]
- ii. camdigikey.client-keystore-file-password = [password of key store file]
- iii. camdigikey.client-keystore-client-key-entry-name= [client_key_entry_name]
- iv. camdigikey.client-keystore-client-key-entry-password= [client_key_entry_password]
- v. camdigikey.client-trust-store-file = [path of the trust store]
- vi. camdigikey.client-trust-store-file-password = [trust_password]

4. User @Autowired annotation of class CamDigiKeyClient

The client Jar library provides methods:

1. validateJwt(String accessToken): HashMap<String, Object>
2. getOrganizationAccessToken(): HashMap<String, Object>
3. getQRCode(): HashMap<String, Object>
4. getUserAccessToken(String qrCode): HashMap<String, Object>
5. refreshUserAccessToken(String accessToken):
HashMap<String, Object>
Note: refresh token should be called before access token expired.
6. logoutAccessToken(String accessToken): HashMap<String, Object>
7. lookupAccountByPersonalCode(String personalCode):
HashMap<String, Object>


```
8. verifyAccountToken(String accountToken): HashMap<String,  
    Object)
```

[Contact us](#)

Tel: +855 81 922 329

Email: info@techostartup.center

Address: RUPP's Compound Russian Federation Blvd., Toul Kork, Phnom Penh, Cambodia [Get Direction](#)