

# HealthCare Authenticator (HCA)

## Quick Start Guide - February 2023

### Table des matières

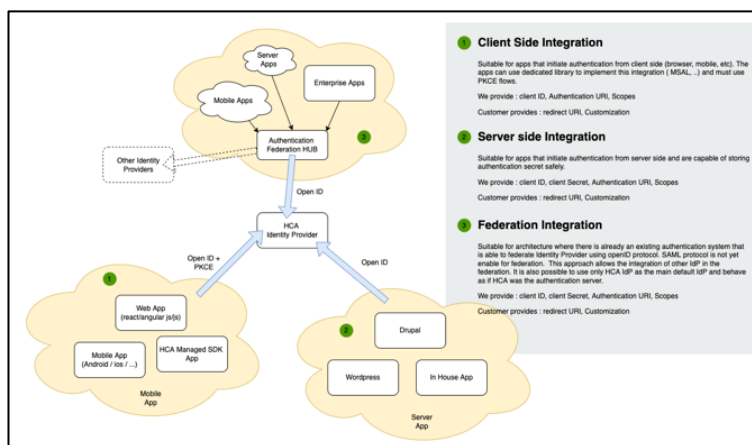
<b>1</b>	<b>Introduction.....</b>	<b>3</b>
1.1	Authentication.....	3
1.2	Identity verification .....	3
1.3	OneKey Profile .....	4
1.4	Connect with a OneKey / OWA account .....	4
1.5	Get a Free Trial Subscription .....	4
<b>2</b>	<b>Client-side integration .....</b>	<b>5</b>
2.1	Requirements .....	5
2.2	Downloading HCA SDK.....	5
2.3	Integration details.....	5
2.3.1	Initialize the HCA SDK.....	5
2.3.2	Sign-up .....	7
2.3.3	Sign-in .....	10
2.3.4	Call backs.....	13
2.4	HCA Discovery end point.....	14
2.4.1	Using HCA Discovery end point.....	14
2.4.2	Using HCA UserInfo end point .....	14
<b>3</b>	<b>Server-side integration .....</b>	<b>16</b>
3.1	Requirements .....	16
3.2	Integration details.....	16
3.2.1	Configuring HCA as identity provider .....	16
3.2.2	Using HCA Discovery end point.....	17
3.2.3	Using HCA UserInfo end point .....	18
3.2.4	Using HCA Me end points .....	19
3.3	Sign-up.....	20
3.3.1	Sign-up form.....	20
3.3.2	MFA form .....	22
3.3.3	Consent Form.....	22
3.4	Sign-in.....	23
3.5	Others.....	23
<b>4</b>	<b>Federation integration.....</b>	<b>24</b>
4.1	Requirements .....	24
4.2	Integration details.....	24
4.2.1	Configuring HCA as a new identity provider .....	24
4.2.2	Using HCA Discovery end point.....	25
4.2.3	Using HCA UserInfo end point .....	25
4.2.4	Using HCA Me end points .....	27

<b>4.3</b>	<b>Sign-up</b> .....	<b>28</b>
4.3.1	Sign-up form.....	28
4.3.2	MFA form .....	29
4.3.3	Consent Form.....	29
<b>4.4</b>	<b>Sign-in</b> .....	<b>30</b>
<b>4.5</b>	<b>Others</b> .....	<b>30</b>
<b>5</b>	<b><i>Get a Production Subscription</i></b> .....	<b>30</b>

# 1 Introduction

In this quick start guide you will find information about how to integrate HealthCare Authenticator in your web site or mobile application. There are different ways to perform such an integration depending on the technology your web site or mobile is using. As examples:

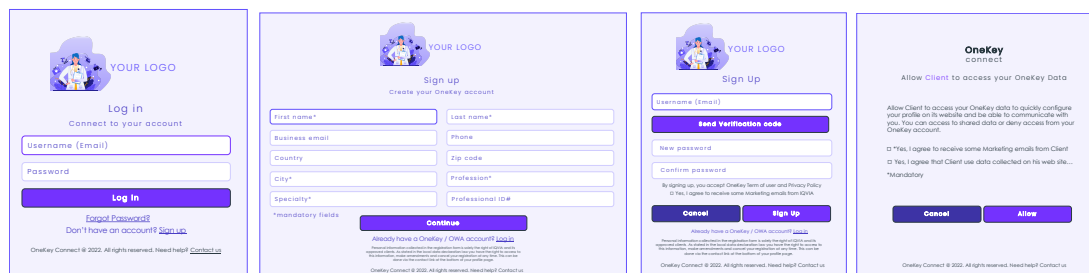
- if your web site uses Javascript, you will opt for a **Client-side integration** using HCA JS or Angular JS SDK using OpenId protocol and a Proof Key for Code Exchange (PKCE).
- If your web site uses a CMS (WordPress, Drupal...) you will opt for a **Server-side integration** using OpenId protocol.
- If your web site already has an authentication solution installed, then you will choose a **Federation integration** using OpenID protocol.



HCA modes of integration

## 1.1 Authentication

In all cases, HCA will manage fully users' authentication, including Sign-up, Multi-Factor authentication, Consent gathering, Sign-in and Forget my password processes.



Examples of screens provided by HCA

## 1.2 Identity verification

HCA uses sign-up data to perform automatic identity verification based on OneKey data. It can also continue the process with manual identity verification performed by our research associates located all around the world. HCA provides an endpoint that allows to get identity verification status.



HCA identity verification process

### 1.3 OneKey Profile

Once HCP identity has been verified, HCA endpoint can provide with additional data coming from our OneKey database.



*Example of data from HCP OneKey profile*

### 1.4 Connect with a OneKey / OWA account

Once an HCP will have created his OneKey account, he will be able to use his OneKey credentials to connect to any other web site or mobile application using HCA or OWA (previous IQVIA version of HCA). If the HCP had already an OWA account, then he can do the same with his OWA credentials.



*HCP can connect directly with his OneKey account*

### 1.5 Get a Free Trial Subscription

To provide you with a free trial version of the HealthCare Authenticator (English non-customizable version with limited features) we will need from you:

- The name of your company
- A technical contact (name and email address)
- Your Domain name
- Your Identity server tenant name
- Redirection endpoint URIs\*.

\*Note that we have set a default endpoint URI to <http://localhost:8080> to allow you a local test. You can provide us with different URIs (generally one for your development environment and one for your production environment). Bear in mind that if you do not use localhost, those URIs must be **https** ones.

In return we will provide you with:

- **A Client ID** that is required to authenticate requests from the HCA components and pre-built screens within your web site or mobile app.
- **Claims URLs**
- **Discovery end point (OpenID connect Discovery URL):**  
[https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c\\_1a\\_hca\\_signup\\_signin/v2.0/.well-known/openid-configuration](https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c_1a_hca_signup_signin/v2.0/.well-known/openid-configuration)

## 2 Client-side integration

### 2.1 Requirements

You must be comfortable developing in HTML5, CSS3, and JavaScript (jQuery and jQuery Mobile) at a minimum.

### 2.2 Downloading HCA SDK

HealthCare Authenticator SDK source code and binary files are available at the following GitHub location: <https://github.com/orgs/hca-sdk>

To integrate the HCA SDK within a website (using Angular JS) or JavaScript app, you can either:

- Load the HCA SDK directly from our CDN, by the following line of code inside your HTML template. We recommend placing it at the end of the body tag to avoid blocking the website initial rendering.

For example:

```
<body>
  <script src="https://static.healthcaresdks.com/hca/v1/hca-sdk.js"></script>
</body>
```

- Download HCA SDK from GitHub and host it by your own.

### 2.3 Integration details

#### 2.3.1 Initialize the HCA SDK

To initialize the HCA SDK, you need to add the following script at the end of the body:

#### JavaScript SDK within your JS/HTML app or website

```
<body>
  <!-- HCA SDK Set Configuration -->
  <script>
    hcaSdk.setHcaSdkConfig(<clientId>[,<scopes>]);
  </script>
</body>
```

#### JavaScript SDK within your Angular app

**// On the angular AppComponent, client should implement OnInit**

```
export class AppComponent implements OnInit {
  isAccountLogged: boolean = false;
  accountProfile!: UserProfile;
  isProfileReady: boolean = false;
```

**// In the OnInit Method configure the SDK:**

```
ngOnInit() {
  this.accountProfile = this.initaccountProfile();
  // HCA Set Configuration
  hcaSdk.setHcaSdkConfig(<clientId>,<scopes>);
```

```

        hcaSdk.setLoginCallBack(this.resultLogin.bind(this));
    }
    onLogin() {
        hcaSdk.signIn();
    }
    onSignup() {
        hcaSdk.signUp();
    }

    initaccountProfile(): UserProfile {
        let userProfile: UserProfile = {
            title: "",
            firstName: "",
            lastName: "",
            email: "",
            phone: "",
            trustLevel: 0
        };
        return userProfile;
    }
}

export interface UserProfile {
    title: string;
    firstName: string;
    lastName: string;
    email: string;
    phone: string;
    trustLevel: number
}

```

**ClientId:** is the unique ID that identifies the application that we provided you with.

**Scopes:** is a list of scopes that defines the information that the application would like to access to.

- <https://auth.onekeyconnect.com/x/profile.basic>

to access to:

- UserID
- TrustLevel

- <https://auth.onekeyconnect.com/x/profile.extended>

to access to:

- UserID
- Onekey ID
- TrustLevel
- Title
- Firstname
- Lastname
- ProfessionalType
- WorkplaceName
- Address
- PostalCode

- City
- County
- Country
- PhoneNumber
- ProfessionalCode
- Specialities

Note that using the 'extended' scope requires an existing OneKey subscription or a specific OneKey contract.

- [https://auth.onekeyconnect.com/x/m\\_regis](https://auth.onekeyconnect.com/x/m_regis)  
to collect mandatory consents from the user during sign-up process
- [https://auth.onekeyconnect.com/x/o\\_mail\\_mark](https://auth.onekeyconnect.com/x/o_mail_mark)  
to collect optional consents from the user during sign-up process

Exemple of configuration:

```
hcaSdk.setHcaSdkConfig(
  context.$config.env.HCA_CLIENT_ID,
  [
    'https://auth.onekeyconnect.com/x/profile.extended',
    'https://auth.onekeyconnect.com/x/m_regis',
    'https://auth.onekeyconnect.com/x/o_mail_mark',
  ],
  false
)
```

Note that you can replace extended by basic. You must not have both basic and extended profile scopes in the same configuration.

## 2.3.2 Sign-up

### 2.3.2.1 Integration of the Sign-up button

Display the 'Sign Up' button UI on your app or web site

To display a Sign-Up button (that will disappear after a user successfully signed up) you should add this script:

#### JavaScript SDK within your JS/HTML app or website

```
<body>
  <div>
    <!-- Place this div where want to display the Sign-Up button -->
    <div id="hca_signup">
    </div>
  </div>
</body>
```

#### JavaScript SDK within your Angular app

```
// In the AppComponent HTML add the Signup Call:  
<li class="nav-item"><a href="#" class="btn-signup waves-effect waves-light"  
(click)="onSignup()">Sign up</a></li>
```

### 2.3.2.2 Sign-up form

YOUR LOGO

### Sign up

Create your OneKey account

First name\* Last name\*

Business email Phone

Country Zip code

City\* Profession\*

Specialty\* Professional ID#

\*mandatory fields

[Continue](#)

Already have a OneKey / OWA account? [Login](#)

Personal information collected in the registration form is solely the right of IGIVA and its approved clients. As stated in the local data declaration law you have the right to access to this information, make amendments and cancel your registration at any time. This can be done via the contact link at the bottom of your profile page.

OneKey Connect © 2022. All rights reserved. Need help? [Contact us](#)

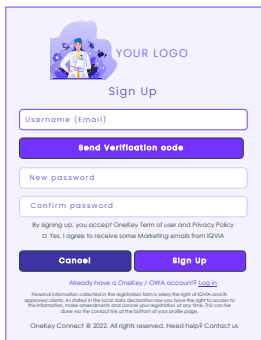
When a user clicks or taps on the sign-up button, he accesses to the sign-up form. The sign-up form contains fields to information on user's profile:

- First Name
- Last Name
- Profession (List of OneKey professional types)
- Specialty (List of OneKey specialties)
- Postcode
- City
- Country (List of countries)
- Professional Code
- Phone
- Business Email

Note that First Name, Last Name, Country, City, Profession and Specialty are mandatory if your web site or mobile app requires a manual identity verification.

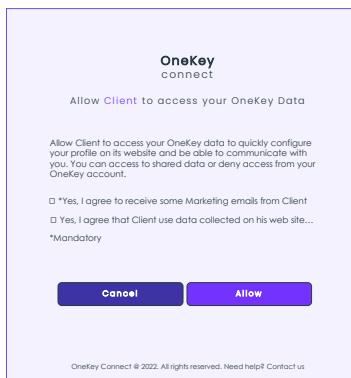


### 2.3.2.3 MFA form



During sign-up process, Multi-Factor Authentication (MFA) is activated and mandatory, and he is required to enter his email address to receive an activation code. Once the email received, he will enter the code and create his password. The MFA form also contains check boxes for user to agree with OneKey terms and conditions (mandatory) and to collect user's consent (optional).

### 2.3.2.4 Consent Form



After Automatic identity verification, user will have to give his consent for us to provide you with data coming:

- His sign-up & MFA forms if he has not yet been identified.
- His sign-up & MFA forms and his OneKey account if he has been identified automatically.

The Consent form can also contain check boxes for user to:

- agree with your web site terms of use and your privacy policy: in that case, you must add (mandatory) [https://auth.onekeyconnect.com/x/m\\_regis](https://auth.onekeyconnect.com/x/m_regis) in the scopes when configuring HCA as identity provider.
- collect user's consents to allow you to use some of his data (up to ten check boxes that can be mandatory or optional): in that case, you need to add [https://auth.onekeyconnect.com/x/o\\_mail\\_mark](https://auth.onekeyconnect.com/x/o_mail_mark) in the scopes when configuring HCA as identity provider.

Note that there are 72 different scopes for mandatory and optional consents (6 types and 6 channels). Connect with us to get the right ones.

### 2.3.3 Sign-in

#### 2.3.3.1 Integration of the Sign-in Button

To display a Sign-in button (that will change to a sign-out after a user successfully logged in / Name of logged user + icon) you should add this script:

Display the 'Sign In' button UI on your app or web site:

#### JavaScript SDK within your JS/HTML app or website

```
<body>
  <div>
    <!-- Place this div where want to display the Sign-In button -->
    <div id="hca_signin">
      </div>
    </div>
  </body>
```

#### JavaScript SDK within your Angular app

```
// In the AppComponent HTML add the LogIn Call:
<li class="nav-item"><a href="#" class="btn-login waves-effect waves-light"
(click)="onLogin()">Log in</a></li>
```

#### 2.3.3.2 Sign-in Form

When a user clicks or taps on the sign-in button, he accesses to the sign-in form. The sign-in form contains:

- Username, which is the email he used during HCA MFA process.
- Password, which is the password he defined during HCA MFA process.

Note that user can also log in with his OWA credentials (previous version of HCA).

From the sign-in form, user can also access to Forget Password form and to Sign-up form.

#### 2.3.3.3 Handle login

To Define the return function to handle the login, you should add this script:

```
<body>
  <!-- HCA SDK Set Configuration -->
  <script>
    hcaSdk.setLoginCallBack(updateUILogin);
```

```
</script>
</body>
```

The callback function below displays the ID of the logged user:

```
function updateUILogin(account) {
  console.log(account.userId);
}
```

#### 2.3.3.4 Connected user

When user successfully signs in, the sign-in button can be replaced by user's information (icon, first name, last name, connection's date & hour).

To do that, please use data coming from basic OneKey Connect user's profile.

Note that an example of HCA's integration is available on GitHub, on which you will see how this capability has been implemented.

#### 2.3.3.5 Service calls for additional data 1

The method `isAccountLogged` checks if the user is logged in or not (return true if the user is logged in and false if not):

```
hcaSdk.isAccountLogged()
```

Examples:

##### JavaScript SDK within your JS/HTML app or website

```
function isLogged() {
  console.log("isLogged:" + hcaSdk.isAccountLogged());
}
```

##### JavaScript SDK within your Angular app

```
isLogged() {
  console.log(hcaSdk.isAccountLogged());
}
```

#### 2.3.3.6 Sign-out link

In order the user to sign-out please use the following function:

##### JavaScript SDK within your JS/HTML app or website

```
hcaSdk.signOut()
```

##### JavaScript SDK within your Angular app

```
onLogout() {
  hcaSdk.signOut();
}
```

#### 2.3.3.7 Edit my profile

To edit his profile and manage his consents, user is redirected to OneKey connect web app at the following method: `hcaSdk.openUserProfile()`

Note that an example of HCA's integration is available on GitHub, on which you will see how this capability has been implemented.

### 2.3.3.8 Service calls for additional data 2

The method `getProfile()` retrieves the profile data of the logged user

Example:

#### JavaScript SDK within your JS/HTML app or website

```
function getProfile() {
  hcaSdk.getProfile(callbackProfile);
}

function callbackProfile(data) {
  console.log(data);
}
```

#### JavaScript SDK within your Angular app

```
getProfilePromise = () => {
  return new Promise((resolve, reject) => {
    hcaSdk.getProfile((data: any, err: any) => {
      if (err) return reject(err)
      resolve(data)
    })
  })
}
```

Note that Get Profile method's use Me API (Cf. below) that provides 2 endpoints to retrieve:

- [/api/hca/user/me/account](#): user's **sign-up data**, including validated email in MFA process and TrustLevel from HCA:
  - UserID
  - First Name
  - Last Name
  - Profession
  - Specialty
  - Postcode
  - City
  - Country
  - Professional Code
  - Phone
  - Business Email
  - Email
  - Trust level
- [/api/hca/user/me/profile](#): user's **OneKey data**. Available OneKey data depends on scopes defined when configuring HCA (Cf. 2.3.1) and on existing OneKey subscription / contract and identity verification process' status or result:

For a profile.basic scope. You will have access to:

- UserID
- TrustLevel

For a profile.extended scope. You will have access to:

- UserID
- Onekey ID
- TrustLevel
- Title
- Firstname
- Lastname
- ProfessionalType
- WorkplaceName
- Address
- PostalCode
- City
- County
- Country
- PhoneNumber
- ProfessionalCode
- Specialities

### 2.3.3.9 First Sign-in & Consent

If a user signs-in for the first time on your web site or mobile app using his OneKey credentials (and didn't sign-up on your web site), then HCA will proceed to an automatic sign-up and user won't have to fill a sign-up form.

In that case, user will have to give his consent for us to provide you with data coming his OneKey account using Consent form mentioned above.

Note: [At the end of the sign in process, our authentication server calls back the redirect URI that was given in the parameters, this triggers a reload of the webpage, the SDK process the information and make it available to your code. It is important to consider this workflow when using SDK's function to access user information. This information is only available once the redirect URI is called.](#)

### 2.3.4 Call backs

There are 3 more call back functions In additions the ones described above:

- A call back function to access Token

```
// HCA Set callback function to handle access Token
hcaSdk.setTokenCallBack(updateUIToken);
```

- A call back function to handle Cancel action

```
// HCA Set callback function to handle cancel action
hcaSdk.setCancelCallBack(onCancel);
```

Ex.: When a user cancels his sign-in or sign-up process, you need to intercept HCA error message to provide user with a message and/or redirect him (e.g. to your home page).

- A call back function to handle Cancel action

```
// HCA Set callback function to handle errors
```

```
hcaSdk.setErrorCallBack(onError);
```

## 2.4 HCA Discovery end point

### 2.4.1 Using HCA Discovery end point

The HCA discovery end point

([https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c\\_1a\\_hca\\_signup\\_signin/v2.0/well-known/openid-configuration](https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c_1a_hca_signup_signin/v2.0/well-known/openid-configuration)) defines a mechanism for you to discover HCA and obtain information needed to interact with it. It enables you to:

- Verify the identity of the End-User based on the authentication performed by Authorization Server,
- Obtain basic profile information about the End-User in an interoperable and REST-like manner (Cf. below: email, displayName, family\_name, trustLevel...),
- Obtain OAuth 2.0 endpoint locations (Cf. below: userinfo\_endpoint, end\_session\_endpoint...).

```
{
  "issuer": "https://auth.onekeyconnect.com/9e429805-fd56-44d0-bb47-085bb52898a8/v2.0/",
  "authorization_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/oauth2/v2.0/authorize?p=b2c_1a_hca_signup_signin",
  "token_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/oauth2/v2.0/token?p=b2c_1a_hca_signup_signin",
  "end_session_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/oauth2/v2.0/logout?p=b2c_1a_hca_signup_signin",
  "jwks_uri": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/discovery/v2.0/keys?p=b2c_1a_hca_signup_signin",
  "userinfo_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/openid/v2.0/userinfo?p=b2c_1a_hca_signup_signin",
  "response_modes_supported": [
    "query",
    "fragment",
    "form_post"
  ],
  "response_types_supported": [
    "code",
    "code id token",
    "code token",
    "code id token token",
    "id token",
    "id token token",
    "token",
    "token id token"
  ],
  "scopes_supported": [
    "openid"
  ],
  "subject_types_supported": [
    "pairwise"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "token_endpoint_auth_methods_supported": [
    "client_secret_post",
    "client_secret_basic"
  ],
  "claims_supported": [
    "email",
    "displayName",
    "family_name",
    "trustLevel",
    "userId",
    "sub",
    "idp",
    "tid",
    "scope",
    "iss",
    "iat",
    "exp",
    "aud",
    "acr",
    "nonce",
    "auth_time"
  ]
}
```

(More details on: [https://openid.net/specs/openid-connect-discovery-1\\_0.html](https://openid.net/specs/openid-connect-discovery-1_0.html))

### 2.4.2 Using HCA UserInfo end point

This HCA Discovery endpoint includes UserInfo endpoint URL, here:

[https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c\\_1a\\_hca\\_signup\\_signin/openid/v2.0/userinfo](https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c_1a_hca_signup_signin/openid/v2.0/userinfo)

METHOD: GET

URL: [https://auth.onekeyconnect.com/auth.onekeyconnect.com/openid/v2.0/userinfo?p=b2c\\_1a\\_hca\\_signup\\_signin](https://auth.onekeyconnect.com/auth.onekeyconnect.com/openid/v2.0/userinfo?p=b2c_1a_hca_signup_signin)

HEADER: Bearer <token>

This UserInfo endpoint is an OAuth 2.0 protected resource where you can retrieve consented claims, or assertions, about the end-user authenticated with his HCA/OneKey account. The claims

are typically packaged in a JSON object where the sub member denotes the subject (end-user) identifier.

You need to call UserInfo endpoint using signed JWT (JSON Web Token) that you obtain during authentication process. It will return a JSON object including UserInfo Fields with data coming from the sign-up and MFA process:

UserInfo field	Description	Source
objectID	UserID	HCA
displayName	Last Name	Sign-up form
givenName	First Name	Sign-up form
surname	Last Name	Sign-up form
zipCode	Postal Code	Sign-up form
city	City name	Sign-up form
country	Country	Sign-up form
phone	Phone number	Sign-up form
businessEmail	Email provided	Sign-up form
profession	Profession Type code	Sign-up form
specialty	Specialty code	Sign-up form
email	Email and username	Sign-up form (MFA)
uci	Professional Code	Sign-up form
oneKeyID	OneKey ID (If the client has access to OneKey extended profile and if user has been identified)	HCA / OneKey
trustLevel	User's trust level	HCA

## 3 Server-side integration

### 3.1 Requirements

#### CMS (Content Management System) users

- You must add a CMS (e.g. Drupal or WordPress) OpenID Connect (OIDC / openidconnect) Client plugin (e.g. <https://plugins.miniorange.com/wordpress-oauth-client-setup> or <https://plugins.miniorange.com/drupal>) to your web site.
- This plugin will be integrated with the HealthCare Authenticator. It will act as a Service provider to establish a trust between HCA (which is the OAuth/OpenID/JWT Identity Provider) and your CMS. This will allow users to quickly and securely login to your CMS site using HCA.

### 3.2 Integration details

#### 3.2.1 Configuring HCA as identity provider

To configure HCA as identity provider, you will need to log as an administrator and follow your CMS OpenID Connect Client plugin documentation. In general, you will be required to add a new identity provider (OAuth provider) by providing the following information:

- Identity provider name : **OneKey**
- Client ID: this refers to the **Client ID** you got from your account on our web site (<https://www.healthcaresdks.com/en/authenticator>).
- Client Secret: this refers to the **Client Secret** you got from your account on our web site. It will be used to create a Policy Key.

The screenshot displays two configuration panels. The left panel, 'Configure OAuth Provider', includes fields for 'Display App Name', 'Redirect / Callback URL', 'Client ID', 'Client Secret', 'Scope', 'azure Tenant', 'Send client credentials in', 'State Parameter', 'Group User Info Endpoint', 'JWKS URL', and 'Login Button'. The right panel, 'Update Application', includes fields for 'Application Name', 'Redirect / Callback URL', 'Client ID', 'Client Secret', 'Scope', 'Authorize Endpoint', 'Access Token Endpoint', 'Get User Info Endpoint', 'State Parameter & Nonce', 'Grant Type', 'Group User Info Endpoint', 'JWKS URL', and 'Login Button'. A 'Test Configuration' table is also visible, listing attributes like 'aud', 'iss', 'iat', 'rnf', 'exp', 'amr', 'email', 'family\_name', 'given\_name', 'idp', 'ipaddr', 'name', and 'oid' with their corresponding values.

Examples of configuration: <https://plugins.miniorange.com/wordpress-oauth-client-setup>

You will also need to provide a list of **scopes** separated by **space** that defines the information that the application would like to access to:

1. **openid**
2. <https://auth.onekeyconnect.com/x/profile.basis> or <https://auth.onekeyconnect.com/x/profile.extended>

to request access to OneKey data that you will get using Me API (Cf. below).

If you do not have a HCA/OneKey contract or OneKey subscription, use the basic profile.basis scope. You will have access to:

- UserID
- TrustLevel





- Obtain OAuth 2.0 endpoint locations (Cf. below: userinfo\_endpoint, end\_session\_endpoint...).

```

{
  "issuer": "https://auth.onekeyconnect.com/9e429805-fd56-44d0-bb47-085bb52898a8/v2.0/",
  "authorization_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/oauth2/v2.0/authorize?p=b2c_1a_hca_signup_signin",
  "token_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/oauth2/v2.0/token?p=b2c_1a_hca_signup_signin",
  "end_session_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/oauth2/v2.0/logout?p=b2c_1a_hca_signup_signin",
  "jwks_uri": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/discovery/v2.0/keys?p=b2c_1a_hca_signup_signin",
  "userinfo_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/openid/v2.0/userinfo?p=b2c_1a_hca_signup_signin",
  "response_modes_supported": [
    "query",
    "fragment",
    "form_post"
  ],
  "response_types_supported": [
    "code",
    "code id token",
    "code token",
    "code id token token",
    "id token",
    "id token token",
    "token",
    "token id token"
  ],
  "scopes_supported": [
    "openid"
  ],
  "subject_types_supported": [
    "pairwise"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "token_endpoint_auth_methods_supported": [
    "client_secret_post",
    "client_secret_basic"
  ],
  "claims_supported": [
    "email",
    "displayName",
    "family_name",
    "trustlevel",
    "userId",
    "sub",
    "log",
    "rid",
    "scope",
    "iss",
    "iat",
    "exp",
    "aud",
    "acr",
    "nonce",
    "auth_time"
  ]
}

```

(More details on: [https://openid.net/specs/openid-connect-discovery-1\\_0.html](https://openid.net/specs/openid-connect-discovery-1_0.html))

### 3.2.3 Using HCA UserInfo end point

This HCA Discovery endpoint includes UserInfo endpoint URL, here:

[https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c\\_1a\\_hca\\_signup\\_signin/openid/v2.0/userinfo](https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c_1a_hca_signup_signin/openid/v2.0/userinfo)

METHOD: GET

URL: [https://auth.onekeyconnect.com/auth.onekeyconnect.com/openid/v2.0/userinfo?p=b2c\\_1a\\_hca\\_signup\\_signin](https://auth.onekeyconnect.com/auth.onekeyconnect.com/openid/v2.0/userinfo?p=b2c_1a_hca_signup_signin)

HEADER: Bearer <token>

This UserInfo endpoint is an OAuth 2.0 protected resource where you can retrieve consented claims, or assertions, about the end-user authenticated with his HCA/OneKey account. The claims are typically packaged in a JSON object where the sub member denotes the subject (end-user) identifier.

You need to call UserInfo endpoint using signed JWT (JSON Web Token) that you obtain during authentication process.

It will return a JSON object including UserInfo Fields with data coming from the sign-up and MFA process:

Userinfo field	Description	Source
objectID	UserID	HCA
displayName	Last Name	Sign-up form
givenName	First Name	Sign-up form
surname	Last Name	Sign-up form
zipCode	Postal Code	Sign-up form

city	City name	Sign-up form
country	Country	Sign-up form
phone	Phone number	Sign-up form
businessEmail	Email provided	Sign-up form
profession	Profession Type code	Sign-up form
specialty	Specialty code	Sign-up form
email	Email and username	Sign-up form (MFA)
uci	Professional Code	Sign-up form
oneKeyID	OneKey ID (If the client has access to OneKey extended profile and if user has been identified)	HCA / OneKey
trustLevel	User's trust level	HCA

User Information Fields: These are the claims related to the OneKey account. Your authentication solution requests these fields from HCA SDK when a user is authenticated with his OneKey account.

### 3.2.4 Using HCA Me end points

HCA provides an API to retrieve User's data: **Me** API. Me API has 2 endpoints to allow you to retrieve user's **sign-up data** and user's **OneKey data** (Cf. below). You need to call Me endpoints using signed JWT (JSON Web Token) that you obtain during authentication process.

#### 3.2.4.1 *me/account*

User's **sign-up data**, including validated email in MFA process and TrustLevel (coming from OneKey):

METHOD: GET  
 URL: <https://apim-prod-westeu-onekey.azure-api.net/api/hca/user/me/account>  
 HEADER: Bearer <token>

- UserID
- First Name
- Last Name
- Profession
- Specialty
- Postcode
- City
- Country
- Professional Code
- Phone
- Business Email
- Email

- Trust level

### 3.2.4.2 *me/profile*

User's **OneKey data**. Available OneKey data depends on scopes defined when configuring HCA (Cf. 4.2.1) and on existing OneKey subscription / contract and identity verification process' status or result.

METHOD: GET  
URL: <https://apim-prod-westeu-onekey.azure-api.net/api/hca/user/me/profile>  
HEADER: Bearer <token>

For a profile.basic scope. You will have access to:

- UserID
- TrustLevel

For a profile.extended scope. You will have access to:

- UserID
- Onekey ID
- TrustLevel
- Title
- Firstname
- Lastname
- ProfessionalType
- WorkplaceName
- Address
- PostalCode
- City
- County
- Country
- PhoneNumber
- ProfessionalCode
- Specialities

## 3.3 Sign-up

### 3.3.1 Sign-up form

YOUR LOGO

Sign up  
Create your OneKey account

First name\* Last name\*

Business email Phone

Country Zip code

City\* Profession\*

Specialty\* Professional ID#

\*mandatory fields

Continue

Already have a OneKey / OWA account? [Log in](#)

Personal information collected in the registration form is solely the right of IGVA and its approved clients. As stated in the local data declaration law you have the right to access to this information, make amendments and cancel your registration at any time. This can be done via the contact link at the bottom of your profile page.

OneKey Connect @ 2022. All rights reserved. Need help? Contact us

When a user clicks or taps on the sign-up button, he accesses to the sign-up form. The sign-up form contains fields to information on user's profile:

- First Name
- Last Name
- Profession (List of OneKey professional types)
- Specialty (List of OneKey specialties)
- Postcode
- City
- Country (List of countries)
- Professional Code
- Phone
- Business Email

Note that:

- First Name, Last Name, Country, City, Profession and Specialty are mandatory if your web site or mobile app requires manual identity verification.

### 3.3.2 MFA form

During sign-up process, Multi-Factor Authentication (MFA) is activated and mandatory, and he is required to enter his email address to receive an activation code. Once the email received, he will enter the code and create his password. The MFA form also contains check boxes for user to agree with OneKey terms and conditions (mandatory) and to collect user's consent (optional).

### 3.3.3 Consent Form

After Automatic identity verification, user will have to give his consent for us to provide you with data coming:

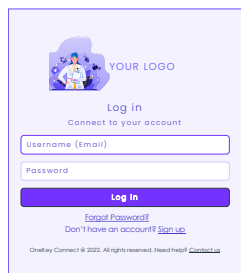
- His sign-up & MFA forms if he has not yet been identified.
- His sign-up & MFA forms and his OneKey account if he has been identified automatically.

The Consent form can also contain check boxes for user to:

- agree with your web site terms of use and your privacy policy: in that case, you must add (mandatory) [https://auth.onekeyconnect.com/x/m\\_regis](https://auth.onekeyconnect.com/x/m_regis) in the scopes when configuring HCA as identity provider.
- collect user's consents to allow you to use some of his data (up to ten check boxes that can be mandatory or optional): in that case, you need to add [https://auth.onekeyconnect.com/x/o\\_mail\\_mark](https://auth.onekeyconnect.com/x/o_mail_mark) in the scopes when configuring HCA as identity provider.

Note that there are 72 different scopes for mandatory and optional consents (6 types and 6 channels). Connect with us to get the right ones.

## 3.4 Sign-in



The image shows a login form with a light purple background. At the top left is a circular logo with a person icon and the text 'YOUR LOGO'. Below the logo is the text 'Log in' and 'Connect to your account'. There are two input fields: 'Username (Email)' and 'Password'. Below these is a purple 'Log In' button. At the bottom, there are links for 'Forgot Password?' and 'Don't have an account? Sign Up'. The footer contains the text 'OneKey Connect © 2022. All rights reserved. Need help? Contact Us'.

### First Sign-in & Consent

If a user signs-in for the first time on your web site or mobile app using his OneKey credentials (and didn't sign-up on your web site), then HCA will proceed to an automatic sign-up and user won't have to fill a sign-up form.

In that case, user will have to give his consent for us to provide you with data coming his OneKey account using Consent form mentioned above.

## 3.5 Others

### Data from Sign-up process

After Sign-up process HCA will give access to data from Sign-up and MFA form and from identity verification if purchased (OneKey record) using endpoints described above.

### Cancel

When a user cancels his sign-in or sign-up process, you need to intercept HCA error message to provide user with a message and/or redirect him (e.g. to your home page).

## 4 Federation integration

### 4.1 Requirements

Your authentication solution must be OpenID compatible.

### 4.2 Integration details

#### 4.2.1 Configuring HCA as a new identity provider

To configure HCA as a new identity provider within your authentication system, you will need to log as an administrator and follow corresponding documentation.

In general, you will be required to add a new identity provider (OAuth provider) by providing the following information:

- Identity provider name: **OneKey**
- Client ID: this refers to the **Client ID** you received from us.
- Client Secret: this refers to the **Client Secret** you received from us. It will be used to create a Policy Key.
- You will also need to provide a list of **scopes** separated by **space** that defines the information that the application would like to access to:
  - <https://auth.onekeyconnect.com/x/profile.basic> or <https://auth.onekeyconnect.com/x/profile.extended> to request access to OneKey data that you will get using Me API (Cf. below).

If you do not have a HCA/OneKey contract or OneKey subscription, use the basic profile.basic scope. You will have access to:

- UserID
- TrustLevel

If you have a HCA/OneKey contract or OneKey subscription, use the basic profile.extended scope. You will have access to:

- UserID
- Onekey ID
- TrustLevel
- Title
- Firstname
- Lastname
- ProfessionalType
- WorkplaceName
- Address
- PostalCode
- City
- County
- Country
- PhoneNumber
- ProfessionalCode
- Specialities



- [https://auth.onekeyconnect.com/x/m\\_regis](https://auth.onekeyconnect.com/x/m_regis)  
to collect mandatory consents from the user during sign-up process
  - [https://auth.onekeyconnect.com/x/o\\_mail\\_mark](https://auth.onekeyconnect.com/x/o_mail_mark)  
to collect optional consents from the user during sign-up process
- User Information Fields: User Information Fields are the claims related to the OneKey account. Your authentication solution requests these fields from HCA SDK when a user is authenticated with his OneKey account.

#### 4.2.2 Using HCA Discovery end point

The HCA discovery end point

([https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c\\_1a\\_hca\\_signup\\_signin/v2.0/well-known/openid-configuration](https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c_1a_hca_signup_signin/v2.0/well-known/openid-configuration)) defines a mechanism for you to discover HCA and obtain information needed to interact with it. It enables you to:

- Verify the identity of the End-User based on the authentication performed by Authorization Server,
- Obtain basic profile information about the End-User in an interoperable and REST-like manner (Cf. below: email, displayName, family\_name, trustLevel...),
- Obtain OAuth 2.0 endpoint locations (Cf. below: userinfo\_endpoint, end\_session\_endpoint...).

```
{
  "issuer": "https://auth.onekeyconnect.com/9e429805-fd56-44d0-bb47-085bb52898a9/v2.0/",
  "authorization_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/oauth2/v2.0/authorize?p=b2c_1a_hca_signup_signin",
  "token_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/oauth2/v2.0/token?p=b2c_1a_hca_signup_signin",
  "end_session_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/oauth2/v2.0/logout?p=b2c_1a_hca_signup_signin",
  "jwks_uri": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/discovery/v2.0/keys?p=b2c_1a_hca_signup_signin",
  "userinfo_endpoint": "https://auth.onekeyconnect.com/auth.onekeyconnect.com/openid/v2.0/userinfo?p=b2c_1a_hca_signup_signin",
  "response_modes_supported": [
    "query",
    "fragment",
    "form_post"
  ],
  "response_types_supported": [
    "code",
    "code id_token",
    "code token",
    "code id token token",
    "id token",
    "id token token",
    "token",
    "token id_token"
  ],
  "scopes_supported": [
    "openid"
  ],
  "subject_types_supported": [
    "pairwise"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "token_endpoint_auth_methods_supported": [
    "client_secret_post",
    "client_secret_basic"
  ],
  "claims_supported": [
    "email",
    "displayName",
    "family_name",
    "trustLevel",
    "userId",
    "sub",
    "idp",
    "tid",
    "scope",
    "iss",
    "iat",
    "exp",
    "aud",
    "acr",
    "nonce",
    "auth_time"
  ]
}
```

(More details on: [https://openid.net/specs/openid-connect-discovery-1\\_0.html](https://openid.net/specs/openid-connect-discovery-1_0.html))

#### 4.2.3 Using HCA UserInfo end point

This HCA Discovery endpoint includes UserInfo endpoint URL, here:

[https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c\\_1a\\_hca\\_signup\\_signin/openid/v2.0/userinfo](https://auth.onekeyconnect.com/auth.onekeyconnect.com/b2c_1a_hca_signup_signin/openid/v2.0/userinfo)

This UserInfo endpoint is an OAuth 2.0 protected resource where you can retrieve consented claims, or assertions, about the end-user authenticated with his HCA/OneKey account. The claims are typically packaged in a JSON object where the sub member denotes the subject (end-user) identifier.

You need to call UserInfo endpoint using signed JWT (JSON Web Token) that you obtain during authentication process.

METHOD: GET  
 URL: [https://auth.onekeyconnect.com/auth.onekeyconnect.com/openid/v2.0/userinfo?p=b2c\\_1a\\_hca\\_signup\\_signin](https://auth.onekeyconnect.com/auth.onekeyconnect.com/openid/v2.0/userinfo?p=b2c_1a_hca_signup_signin)  
 HEADER: Bearer <token>

It will return a JSON object including UserInfo Fields with data coming from the sign-up and MFA process:

Userinfo field	Description	Source
objectID	UserID	HCA
displayName	Last Name	Sign-up form
givenName	First Name	Sign-up form
surname	Last Name	Sign-up form
zipCode	Postal Code	Sign-up form
city	City name	Sign-up form
country	Country	Sign-up form
phone	Phone number	Sign-up form
businessEmail	Email provided	Sign-up form
profession	Profession Type code	Sign-up form
specialty	Specialty code	Sign-up form
email	Email and username	Sign-up form (MFA)
uci	Professional Code	Sign-up form
oneKeyID	OneKey ID (If the client has access to OneKey extended profile and if user has been identified)	HCA / OneKey
trustLevel	User's trust level	HCA

User Information Fields: These are the claims related to the OneKey account. Your authentication solution requests these fields from HCA SDK when a user is authenticated with his OneKey account.

#### 4.2.4 Using HCA Me end points

HCA provides an API to retrieve User's data: **Me** API. Me API has 2 endpoints to allow you to retrieve user's **sign-up data** and user's **OneKey data** (Cf. below). You need to call Me endpoints using signed JWT (JSON Web Token) that you obtain during authentication process.

##### 4.2.4.1 *me/account*

User's **sign-up data**, including validated email in MFA process and TrustLevel (coming from OneKey):

```
METHOD: GET
URL: https://apim-prod-westeu-onekey.azure-api.net/api/hca/user/me/account
HEADER: Bearer <token>
```

- UserID
- First Name
- Last Name
- Profession
- Specialty
- Postcode
- City
- Country
- Professional Code
- Phone
- Business Email
- Email
- Trust level

##### 4.2.4.2 *me/profile*

User's **OneKey data**. Available OneKey data depends on scopes defined when configuring HCA (Cf. 4.2.1) and on existing OneKey subscription / contract and identity verification process' status or result.

```
METHOD: GET
URL: https://apim-prod-westeu-onekey.azure-api.net/api/hca/user/me/profile
HEADER: Bearer <token>
```

For a profile.basic scope. You will have access to:

- UserID
- TrustLevel

For a profile.extended scope. You will have access to:

- UserID
- Onekey ID

- TrustLevel
- Title
- Firstname
- Lastname
- ProfessionalType
- WorkplaceName
- Address
- PostalCode
- City
- County
- Country
- PhoneNumber
- ProfessionalCode
- Specialities

## 4.3 Sign-up

### 4.3.1 Sign-up form

YOUR LOGO

Sign up  
Create your OneKey account

First name\* Last name\*

Business email Phone

Country Zip code

City\* Profession\*

Specialty\* Professional ID#

\*mandatory fields

Continue

Already have a OneKey / OWA account? [Log in](#)

Personal information collected in the registration form is solely the right of OWA and its approved clients. As stated in the local data declaration law you have the right to access to this information, make amendments and cancel your registration at any time. This can be done via the contact link at the bottom of your profile page.

OneKey Connect © 2022. All rights reserved. Need help? Contact us

When a user clicks or taps on the sign-up button, he accesses to the sign-up form. The sign-up form contains fields to information on user's profile:

- First Name
- Last Name
- Profession (List of OneKey professional types)
- Specialty (List of OneKey specialties)
- Postcode
- City
- Country (List of countries)
- Professional Code
- Phone
- Business Email

Note that:

- First Name, Last Name, Country, City, Profession and Specialty are mandatory if your web site or mobile app requires manual identity verification.

### 4.3.2 MFA form

During sign-up process, Multi-Factor Authentication (MFA) is activated and mandatory, and he is required to enter his email address to receive an activation code. Once the email received, he will enter the code and create his password. The MFA form also contains check boxes for user to agree with OneKey terms and conditions (mandatory) and to collect user’s consent (optional).

### 4.3.3 Consent Form

After Automatic identity verification, user will have to give his consent for us to provide you with data coming:

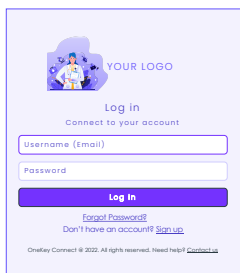
- o His sign-up & MFA forms if he has not yet been identified.
- o His sign-up & MFA forms and his OneKey account if he has been identified automatically.

The Consent form can also contain check boxes for user to:

- o agree with your web site terms of use and your privacy policy: in that case, you must add (mandatory) [https://auth.onekeyconnect.com/x/m\\_regis](https://auth.onekeyconnect.com/x/m_regis) in the scopes when configuring HCA as identity provider.
- o collect user’s consents to allow you to use some of his data (up to ten check boxes that can be mandatory or optional): in that case, you need to add [https://auth.onekeyconnect.com/x/o\\_mail\\_mark](https://auth.onekeyconnect.com/x/o_mail_mark) in the scopes when configuring HCA as identity provider.

Note that there are 72 different scopes for mandatory and optional consents (6 types and 6 channels). Connect with us to get the right ones.

## 4.4 Sign-in



### First Sign-in & Consent

If a user signs-in for the first time on your web site or mobile app using his OneKey credentials (and didn't sign-up on your web site), then HCA will proceed to an automatic sign-up and user won't have to fill a sign-up form.

In that case, user will have to give his consent for us to provide you with data coming his OneKey account using Consent form mentioned above.

## 4.5 Others

### Data from Sign-up process

After Sign-up process HCA will give access to data from Sign-up and MFA form and from identity verification if purchased (OneKey record) using endpoints described above.

### Cancel

When a user cancels his sign-in or sign-up process, you need to intercept HCA error message to provide user with a message and/or redirect him (e.g. to your home page).

## 5 Get a Production Subscription

You tested your app or web site with HCA free trial solution and now you want to go live with real data and a customized solution. To provide you with a Pro version, we will need from you the following information:

- List of countries and for each country:
  - Language(s)
  - Which level of identity verification do you require?
    - TL1: no identity verification
    - TL2A: automatic identity verification only
    - TL2: manual identity verification done by research associates using different data sources if TL2A failed
    - TL4: Identity verification made by phone by research associates.
  - Graphical elements for sign-in, sign-up, MFA and Consent forms.
  - Links to your web site terms and conditions and Privacy.
  - List of the fields (checkbox) you want us to add during sign-up process to collect mandatory and optional user's consents.
  - URL of your web site (https) and Production Redirection endpoint URIs.

We will provide with a Customization guide to help you to provide us all these information. Once we will provide you the customized pro version of HCA SDK, you will be good to go.