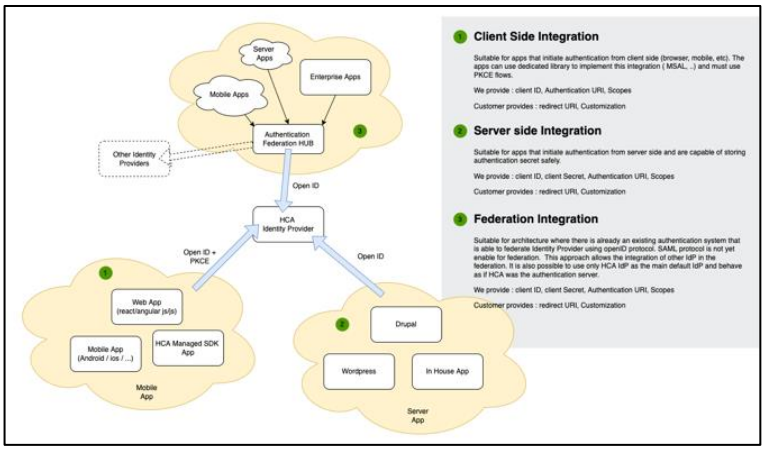
HealthCare Authenticator (HCA)

Quick Start Guide - September 2024

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:

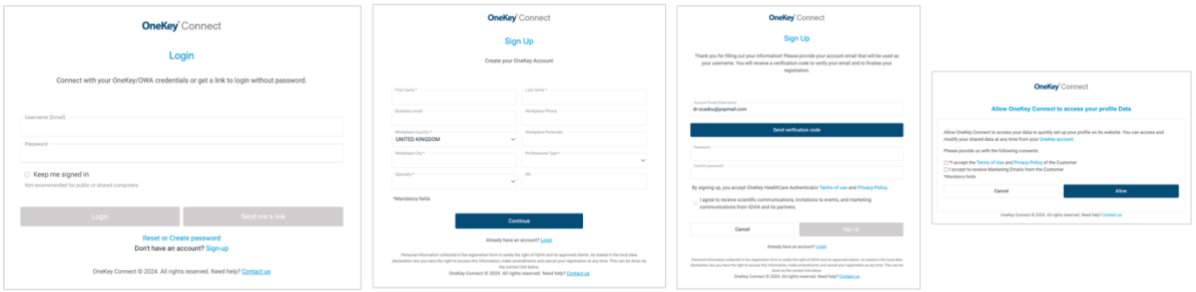
- If your web site uses Javascript, you will opt for a Client-side integration using HCA JS or Angular JS SDK using OpenId protocol and a Proof Key for Code Exchange (PKCE).
- If your web site uses a CMS (WordPress, Drupal…) you will opt for a Server-side integration using OpenId protocol.
- If your 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 get a HealthCare Authenticator Free Trial, please read the Free Trial Quick Start Guide.

Note that:

- HCA Free Trial version is a standard English version without any customization and no access to OneKey Data.
- 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

2 Client-side integration

2.1 Requirements

You should feel comfortable working with HTML5, CSS3, and JavaScript (JQuery and JQuery Mobile) at the very least.

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.

  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>[,<displaySignInButton>, <displaySignUpButton>, <scopes>, <knownAuthorities>, <tenantDoman>, <policyId>, <signupPolicyId>, <apimSubscriptionKey>, <apiBasePath>, <errorRedirectUrl>, <redirectUrl>]);
  </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>, <displaySignInButton>, <displaySignUpButton>, <scopes>, <knownAuthorities>, <tenantDoman>, <policyId>, <signupPolicyId>, <apimSubscriptionKey>, <apiBasePath>, <errorRedirectUrl>, <redirectUrl>);
      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 (string, **required**): is the unique ID that identifies the application that we provided you with.
- displaySignInButton (boolean, **optional**, default: true): controls whether the sign-in button is displayed. Set to false to hide the sign-in button.
- displaySignUpButton (boolean, **optional**, default: true): controls whether the sign-up button is displayed. Set to false to hide the sign-up button.
- Scopes (array, optional, default: [“https://auth.onekeyconnect.com/x/profile.basic“]): is a list of scopes that defines the information that the application would like to access to.
- knownAuthorities (array, **optional**, default: ["auth.onekeyconnect.com"]): Known authorities define the trusted authentication providers.
- tenantDomain (string, **optional**, default: "auth.onekeyconnect.com"): The domain of the tenant used for authentication.
- policyId (string, **optional**, default: "b2c_1a_hca_signup_signin"): The policy identifier for combined sign-up and sign-in flow.
- signupPolicyId (string, **optional**, default: "b2c_1a_hca_signuponly"): The policy identifier for sign-up only, used when the user needs to create a new account.
- apimSubscriptionKey (string, **optional**, default: ""): API Key of the application used for accessing the API.
- apiBasePath (string, **optional**, default: https://api.healthcaresdks.com/api/hca/user/me): The base path of the API used to retrieve user information.
- errorRedirectUrl (string, **optional**, default: ""): The URL to which users will be redirected in case of an error during the authentication. If left empty, no redirection will occur.
- redirectUrl (string, optional, default: “”): The URL to which users will be redirected after successful authentication. If left empty, users will be redirected to the page where the authentication is initiated.

From your configuration interface, you can provide a list of **scopes separated by a comma**:

  - Mandatory scope: Openid
  - Mandatory scope: https://auth.onekeyconnect.com/x/profile.basic is configured by default.

- https://auth.onekeyconnect.com/x/profile.basic will give you access to:
  - UserID
  - TrustLevel
- https://auth.onekeyconnect.com/x/profile.extended will give you access to:
  - UserID
  - Onekey ID
  - TrustLevel
  - Title
  - Firstname
  - Lastname
  - ProfessionalType
  - WorkplaceName
  - Address
  - PostalCode
  - City
  - County
  - Country
  - PhoneNumber
  - ProfessionalCode
  - Specialities

Note that:

- Using the scope “profile.extended” requires an existing OneKey subscription or a specific OneKey contract.
- You have to replace the scope “profile.extended” by “profile.basic”. You can’t have both 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>
```

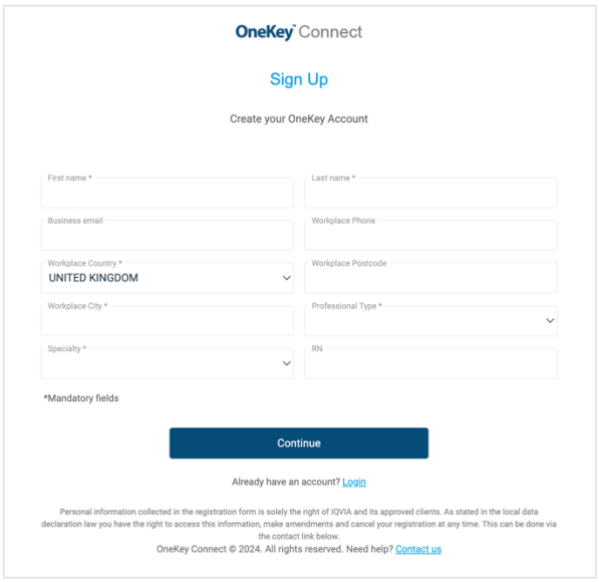
JavaScript SDK within your Angular app

```
// 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

When a user clicks on the Sign-up button, he accesses to the Sign-up form.
The Sign-up form contains fields to information on user’s profile:

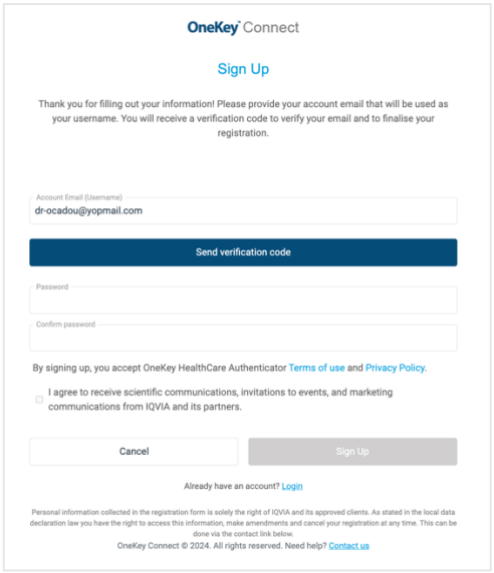
- First Name
- Last Name
- Email
- Phone (Workplace)
- Country (List of countries)
- Postal code (Workplace)
- City (Workplace)
- Professional type (List of OneKey professional types)
- Specialty (List of OneKey specialties)
- Professional Code



Note that: First Name, Last Name, Country, City, Professional type and Specialty are mandatory for identity verification purpose (automatic and manual).

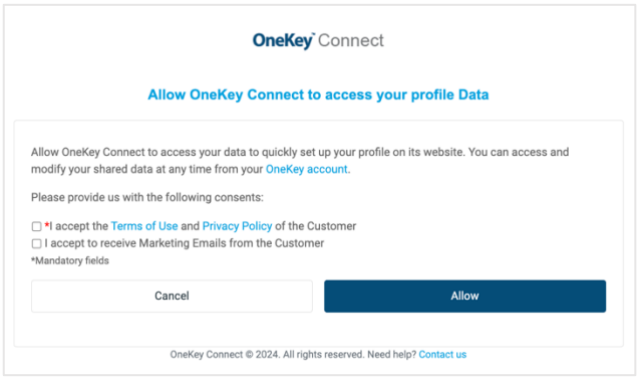
2.3.2.3 MFA form

During the Sign-up process, Multi-Factor Authentication (MFA) is activated and mandatory. The user has to enter his email address to receive a verification code, to enter in the form to verify his email. Then he can create his password.



2.3.2.4 Consent form

At the last step of the sign-up process, the user will have to accept the Data share consent. The Data share consent allows HealthCare Authenticator to share the user's sign-up details with your company and allows you to collect your users' consents.



Note that: You can create and collect up to 10 mandatory and 10 optional consents displayed in the Data share consent page.

2.3.3 Sign-in

2.3.3.1 Integration of the Sign-in Button

Display the "Sign-In" button UI on your app or web site.

To display a Sign-in button (that will turn to a logout after a user successfully logged in / Name of logged user + icon) 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-In button -->
        <div id="hca_signin">
        </div>
     </div>
 </body>
```

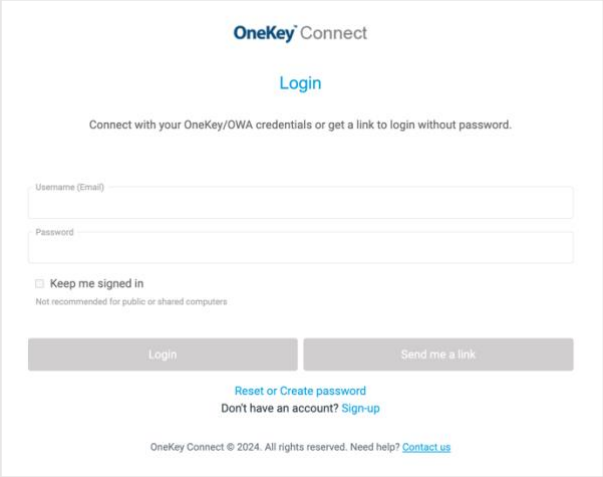
JavaScript SDK within your Angular app

```
// 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 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:

- The user can also log in with his OWA credentials (previous version of HCA).
- From the sign-in form, the 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 is AccountLogged 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

To allow 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 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 from HCA:

Me/account fields	Description
Id, userId	ID of the user
trustLevel	User's trust level (level of identity verification)
firstName	User's first name
lastName	User's last name
email	Email and username
businessEmail	User's business email
phone	Workplace phone
country	User's country
zipCode	Workplace postal code
city	Workplace city
profession	Professional type code
specialty	Specialty code
uci	Professional code

- /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:

Me/profile fields	Description
userId	ID of the user
trustLevel	User's trust level

  For a "profile.extended" scope, you will have access to:

Me/profile fields	Description
userId	HCA internal user ID
ucis	User's local professional code
trustLevel	User's Trust level (level of identity verification validation) which comes from HCA database or ID verification process
title (code, label, locale)	Code of Professional Title of the Healthcare Professional (HCP). Ex. Doctor, Professor… or Courtesy title (Mr., Mrs., Miss)
firstName	HCP's First Name in local Alphabet
lastName	HCP's Last Name in local Alphabet
suffixName	HCP's Suffix Name in local Alphabet
professionalType (code, label, locale)	HCP's profession label in sign-up's form's locale

specialties	HCP's profession local OneKey code
intlPhone	International Phone number of the healthcare professional at a workplace.
workplaceName	Official name for the Workplace
ZipCode	Workplace Postal code or zip code
City (code, label, locale)	City Name
Country	ISO-2 country code like FR, GB, DE…
localPhone	workplace international phone
IntlFax	workplace international fax
business name	workplace name
businessAddress	long label of workplace address (i.e. full readable address)
npiNumber	map of user uci (ex: {“adeli":1234, "rpps": 123})

2.3.3.8 First Sign-in & Consent

If a user signs-in for the first time on your web site or mobile app using his OneKey/OWA credentials (and didn’t Sign-up on your web site), then HCA will proceed to an automatic Sign-in.

In that case, user will have to accept the Data share consent page for us to provide you with data coming from HCA sign-up and OneKey details.

Note that:

At the end of the sign in process, our authentication server calls back the redirect URI that was given in the parameters. This will trigger a reload of the webpage, the SDK will 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 (setLoginCallback) 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);
```

Example: 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);
```

2.3.5 Custom locale

The **hcaSdk.setLocaleParams(locale)** method allows you to specify the language or regional settings (locale) for the HCA's sign-in / sign-up page.

```
// Set locale params for HCA's signin / signup pages
hcaSdk.setLocaleParams("fr-FR");
```

2.3.6 Custom button labels

To customize the text labels for the sign-in, sign-up, and sign-out buttons, you could use the **hcaSdk.setLabels(<signIn>, <signUp>, <signOut>)** methods.

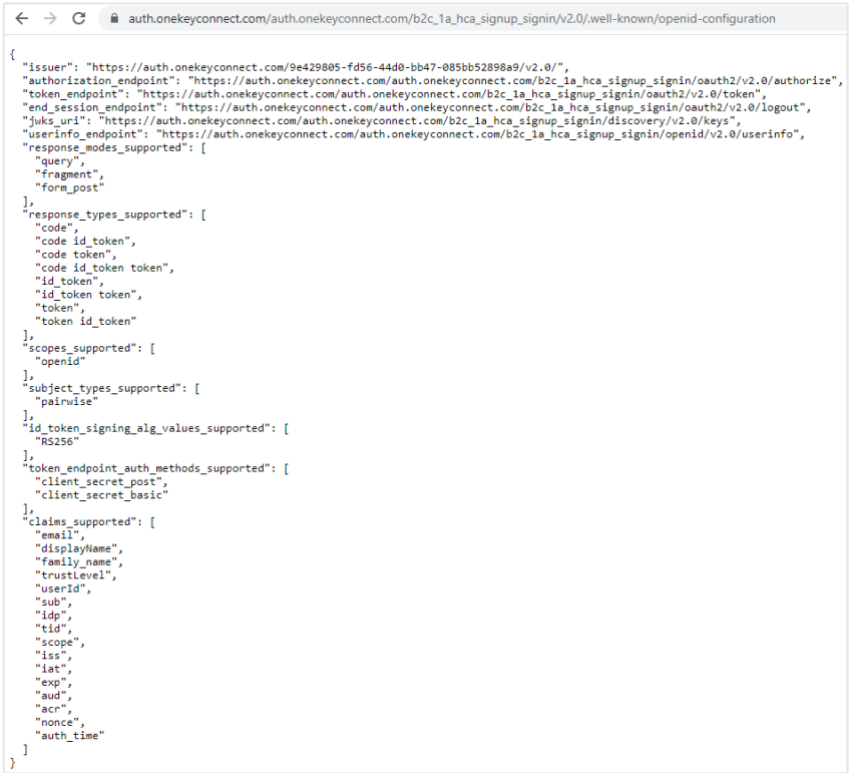
```
// Set text labels for the sign-in, sign-up, and sign-out buttons
hcaSdk.setLabels("Sign In with OneKey", "Sign Up with OneKey", "Sign Out from OneKey");
```

2.4 HCA Discovery end point

2.4.1 Using HCA Discovery end point

The HealthCare Authenticator Discovery end point 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…).



(More details on: 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

METHOD: GET

URL:
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 sign-up/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.

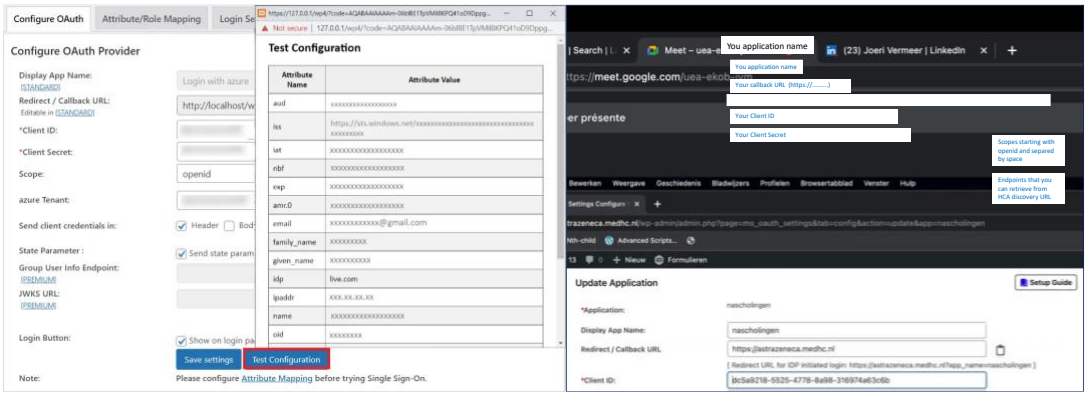
Note that: On our website, from the Developers page, Videos section, you will find tutorial videos for WordPress/Drupal integration.

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.



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

From your configuration interface, you can provide a list of **scopes separated by a space**:

- Mandatory scope: Openid
- Mandatory scope: https://auth.onekeyconnect.com/x/profile.basic is configured by default.

- https://auth.onekeyconnect.com/x/profile.basic will give you access to:
  - UserID
  - TrustLevel
- https://auth.onekeyconnect.com/x/profile.extended will give you access to:
  - UserID
  - Onekey ID
  - TrustLevel
  - Title
  - Firstname
  - Lastname
  - ProfessionalType
  - WorkplaceName
  - Address
  - PostalCode
  - City
  - County
  - Country
  - PhoneNumber
  - ProfessionalCode
  - Specialities

Note that:

- Using the scope “profile.extended” requires an existing OneKey subscription or a specific OneKey contract.
- You have to replace the scope “profile.extended” by “profile.basic”. You can’t have both scopes in the same configuration.

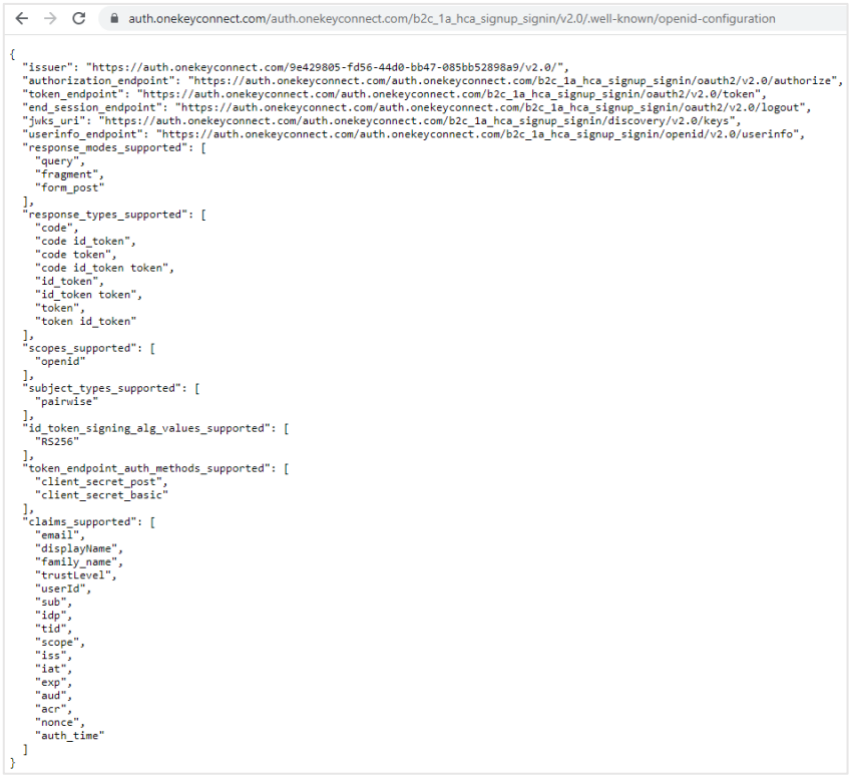
Generally, the solution you use will allow you to test the configuration. In that case, you will see all the values returned by HCA (First Name, Last Name, Email) to CMS in a table (Cf. below).
Once you see all the required values, then you will have to proceed User Attribute / Role Mapping which is mandatory for enabling users to successfully login into CMS.

Test configuration example

3.2.2 Using HCA Discovery end point

The HealthCare Authenticator Discovery end point 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…).



(More details on: 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

METHOD: GET

URL: 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 sign-up/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

3.2.4 Using HCA Me end points

Me API has 2 endpoints to allow you to retrieve user's Sign-up data and user's OneKey data. You need to call the Me endpoints using signed JWT (JSON Web Token) that you obtain during authentication process.

3.2.4.1 me/account

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

/api/hca/user/me/account: user's **Sign-up data**, including validated email in MFA process from HCA:

Me/account fields	Description
Id, userId	ID of the user
trustLevel	User's trust level (level of identity verification)
firstName	User's first name
lastName	User's last name
email	Email and username
businessEmail	User's business email

phone	Workplace phone
country	User's country
zipCode	Workplace postal code
city	Workplace city
profession	Professional type code
specialty	Specialty code
uci	Professional code

3.2.4.2 me/profile

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

/api/hca/user/me/profile: user's **OneKey data**. Available OneKey data depends on scopes defined when configuring HCA (Cf. 3.2.1) and on existing OneKey subscription / contract and identity verification process' status or result:

For a "profile.basic" scope, you will have access to:

Me/profile fields	Description
userId	ID of the user
trustLevel	User's trust level

For a "profile.extended" scope, you will have access to:

Me/profile fields	Description
userId	HCA internal user ID
ucis	User's local professional code
trustLevel	User's Trust level (level of identity verification validation) which comes from HCA database or ID verification process
title (code, label, locale)	Code of Professional Title of the Healthcare Professional (HCP). Ex. Doctor, Professor… or Courtesy title (Mr., Mrs., Miss)
firstName	HCP's First Name in local Alphabet
lastName	HCP's Last Name in local Alphabet
suffixName	HCP's Suffix Name in local Alphabet
professionalType (code, label, locale)	HCP's profession label in sign-up's form's locale
specialties	HCP's profession local OneKey code
intlPhone	International Phone number of the healthcare professional at a workplace.
workplaceName	Official name for the Workplace
ZipCode	Workplace Postal code or zip code
City (code, label, locale)	City Name

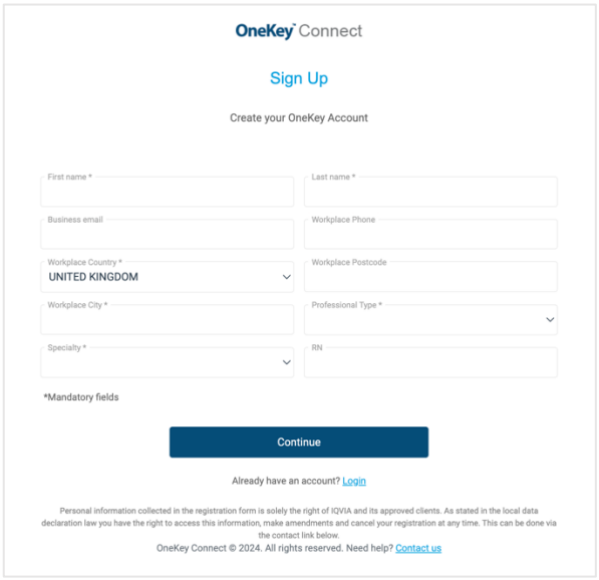
Country	ISO-2 country code like FR, GB, DE…
localPhone	workplace international phone
IntlFax	workplace international fax
business name	workplace name
businessAddress	long label of workplace address (ie full readable address)
npiNumber	map of user uci (ex : { "adeli":1234, "rpps": 123 })

3.3 Sign-up

3.3.1 Sign-up form

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

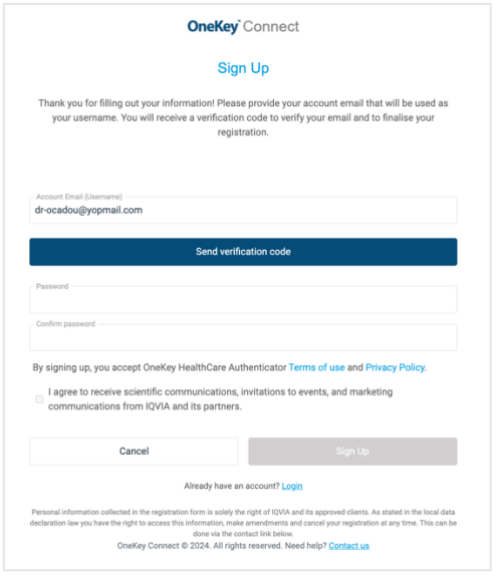
- First Name
- Last Name
- Email
- Phone (Workplace)
- Country (List of countries)
- Postal code (Workplace)
- City (Workplace)
- Professional type (List of OneKey professional types)
- Specialty (List of OneKey specialties)
- Professional Code



Note that: First Name, Last Name, Country, City, Professional type and Specialty are mandatory for identity verification purpose (automatic and manual).

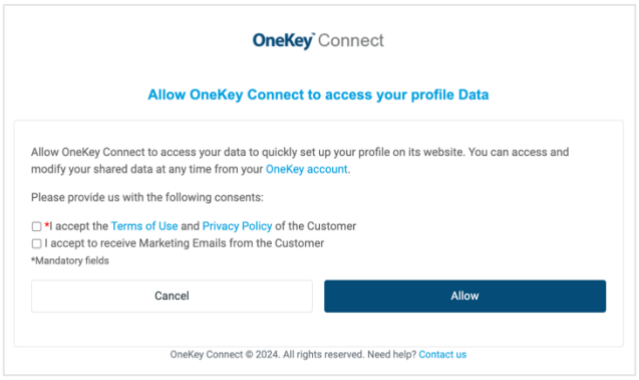
3.3.2 MFA form

During the Sign-up process, Multi-Factor Authentication (MFA) is activated and mandatory. The user has to enter his email address to receive a verification code, to enter in the form to verify his email. Then he can create his password.



3.3.3 Consent form

At the last step of the sign-up process, the user will have to accept the Data share consent. The Data share consent allows HealthCare Authenticator to share the user's sign-up details with your company and allows you to collect your users' consents.

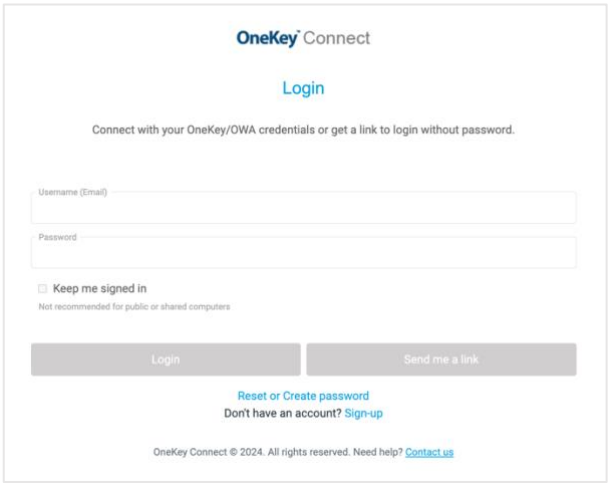


Note that: You can create and collect up to 10 mandatory and 10 optional consents displayed in the Data share consent page.

3.4 Sign-in

When a user clicks 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:

- The user can also log in with his OWA credentials (previous version of HCA).
- From the sign-in form, the user can also access to Forget Password form and to Sign-up form.

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 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.

From your configuration interface, you can provide a list of **scopes separated by a space**:

- Mandatory scope: Openid
- Mandatory scope: https://auth.onekeyconnect.com/x/profile.basic is configured by default.

- https://auth.onekeyconnect.com/x/profile.basic will give you access to:
  - UserID
  - TrustLevel
- https://auth.onekeyconnect.com/x/profile.extended will give you access to:
  - UserID
  - Onekey ID
  - TrustLevel
  - Title
  - Firstname
  - Lastname
  - ProfessionalType
  - WorkplaceName
  - Address
  - PostalCode
  - City
  - County
  - Country
  - PhoneNumber
  - ProfessionalCode
  - Specialities

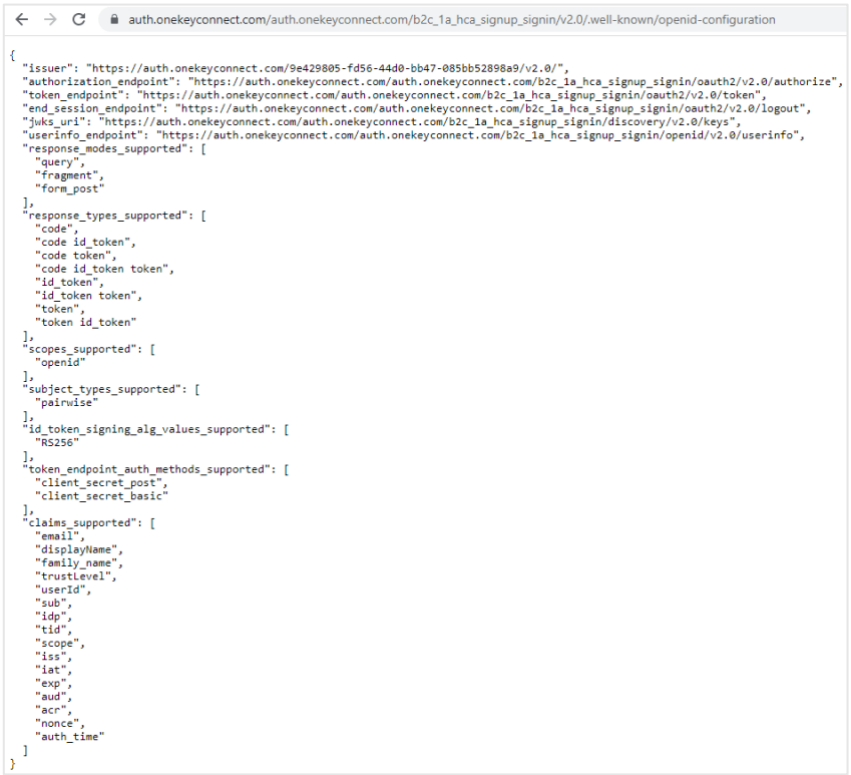
Note that:

- Using the scope “profile.extended” requires an existing OneKey subscription or a specific OneKey contract.
- You have to replace the scope “profile.extended” by “profile.basic”. You can’t have both scopes in the same configuration.
- 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 HealthCare Authenticator Discovery end point 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…).



(More details on: 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

METHOD: GET

URL:
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 sign-up/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

4.2.4 Using HCA Me end points

Me API has 2 endpoints to allow you to retrieve user's Sign-up data and user's OneKey data. You need to call the Me endpoints using signed JWT (JSON Web Token) that you obtain during authentication process.

4.2.4.1 me/account

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

/api/hca/user/me/account: user's **Sign-up data**, including validated email in MFA process from HCA:

Me/account fields	Description
Id, userId	ID of the user
trustLevel	User's trust level (level of identity verification)
firstName	User's first name
lastName	User's last name
email	Email and username
businessEmail	User's business email
phone	Workplace phone
country	User's country
zipCode	Workplace postal code
city	Workplace city
profession	Professional type code
specialty	Specialty code
uci	Professional code

4.2.4.2 me/profile

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

/api/hca/user/me/profile: user's **OneKey data**. Available OneKey data depends on scopes defined when configuring HCA (Cf. 3.2.1) and on existing OneKey subscription / contract and identity verification process' status or result:

For a "profile.basic" scope, you will have access to:

Me/profile fields	Description
userId	ID of the user
trustLevel	User's trust level

For a “profile.extended” scope, you will have access to:

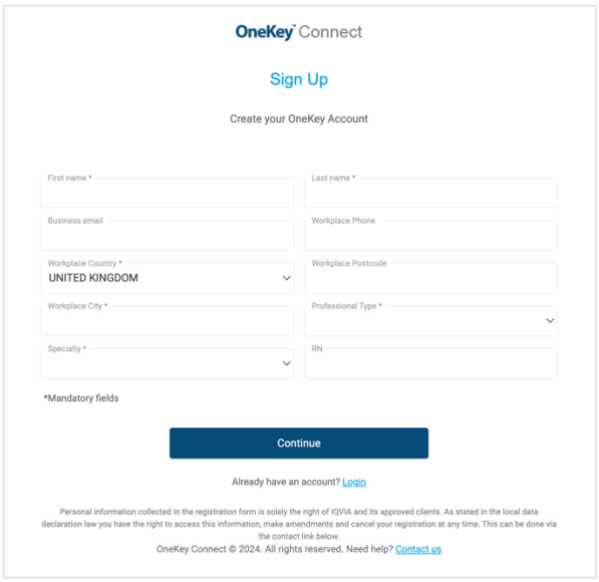
Me/profile fields	Description
userId	HCA internal user ID
ucis	User's local professional code
trustLevel	User's Trust level (level of identity verification validation) which comes from HCA database or ID verification process
title (code, label, locale)	Code of Professional Title of the Healthcare Professional (HCP). Ex. Doctor, Professor… or Courtesy title (Mr., Mrs., Miss)
firstName	HCP's First Name in local Alphabet
lastName	HCP's Last Name in local Alphabet
suffixName	HCP's Suffix Name in local Alphabet
professionalType (code, label, locale)	HCP's profession label in sign-up's form's locale
specialties	HCP's profession local OneKey code
intlPhone	International Phone number of the healthcare professional at a workplace.
workplaceName	Official name for the Workplace
ZipCode	Workplace Postal code or zip code
City (code, label, locale)	City Name
Country	ISO-2 country code like FR, GB, DE…
localPhone	workplace international phone
IntlFax	workplace international fax
business name	workplace name
businessAddress	long label of workplace address (ie full readable address)
npiNumber	map of user uci (ex : { "adeli":1234, "rpps": 123 })

4.3 Sign-up

4.3.1 Sign-up form

When a user clicks on the Sign-up button, he accesses to the Sign-up form.
The Sign-up form contains fields to information on user’s profile:

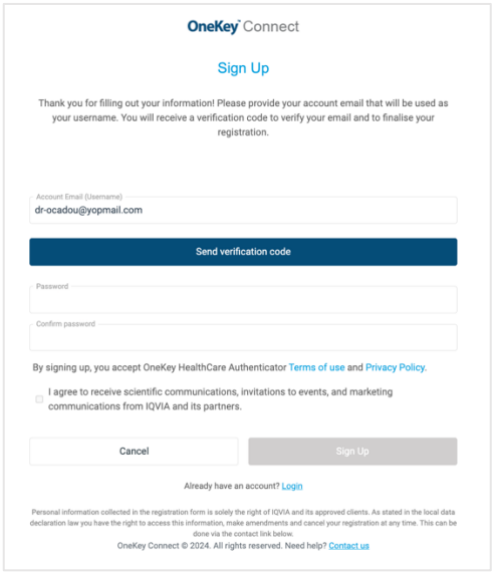
- First Name
- Last Name
- Email
- Phone (Workplace)
- Country (List of countries)
- Postal code (Workplace)
- City (Workplace)
- Professional type (List of OneKey professional types)
- Specialty (List of OneKey specialties)
- Professional Code



Note that: First Name, Last Name, Country, City, Professional type and Specialty are mandatory for identity verification purpose (automatic and manual).

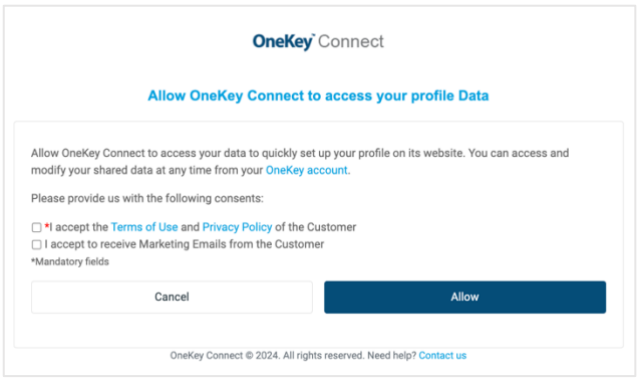
4.3.2 MFA form

During the Sign-up process, Multi-Factor Authentication (MFA) is activated and mandatory. The user has to enter his email address to receive a verification code, to enter in the form to verify his email. Then he can create his password.



4.3.3 Consent form

At the last step of the sign-up process, the user will have to accept the Data share consent. The Data share consent allows HealthCare Authenticator to share the user's sign-up details with your company and allows you to collect your users' consents.

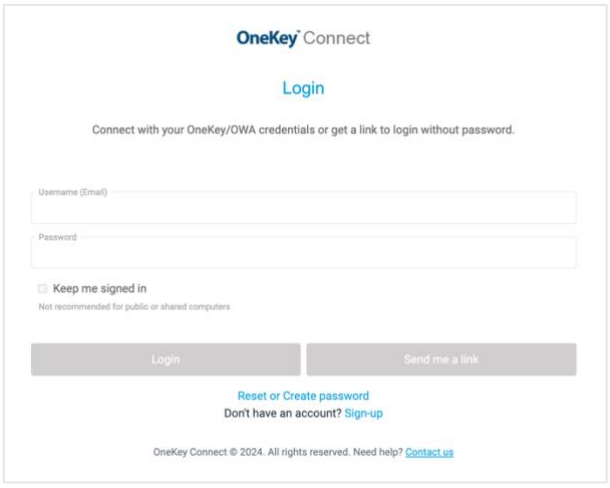


Note that: You can create and collect up to 10 mandatory and 10 optional consents displayed in the Data share consent page.

4.4 Sign-in

When a user clicks 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:

- The user can also log in with his OWA credentials (previous version of HCA).
- From the sign-in form, the user can also access to Forget Password form and to Sign-up form.

4.5 Others

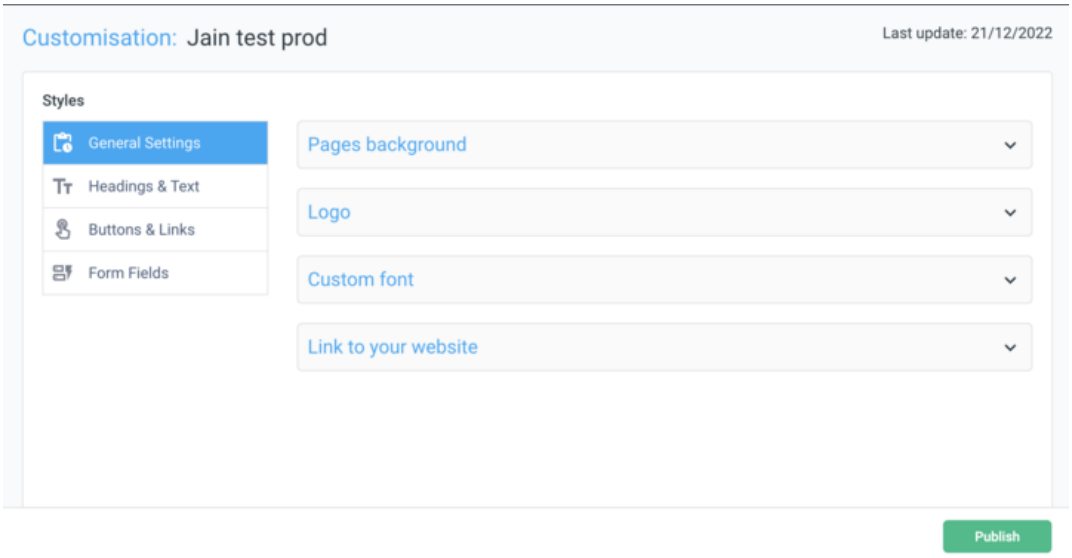
Data from Sign-up process
After Sign-up process HCA will give access to data from Sign-up and MFA form and from identity verification if purchased (OneKey record) using endpoints described above.

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 Pro version & Go live!

Once you tested your website or mobile app with HCA free trial solution, we will upgrade your free trial version in Pro version. With the Pro version, you will be able to:

- Select the countries needed for your website or mobile app.
- Customize your HCA pages:
  - **Graphical:** from your account on https://www.healthcaresdks.com/en/, in the tab "Customisation" > "Graphical" (see screenshot below), you can customize the logo, the fonts, the colors, etc. à Link to the graphical customisation guide HERE which will help you use the customization tool.



  - **Consents:** we will send you an Excel file to complete with the consents you will display to the users in the data share consent page in all languages needed. Once we receive all the information needed, we will implement the consent customization.

After this last step, you can do the final sanity check, and you are ready to go live!