FranceConnect Façade (FCF)

Zzz

Addendum to the technical-functional specifications intended for Microsoft Dynamics 365 BizApps Portals, Power App Portals/Power Pages applications, and Azure AD B2C support

Version Beta - April 2023

This page is intentionally left blank.

Notice

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user.

A drawing of a face

Description automatically generated

Attribution 4.0 International (CC BY 4.0)

Microsoft and any contributors grant you a license to this document under the [Creative Commons Attribution 4.0 International Public License](https://creativecommons.org/licenses/by/4.0/legalcode), see the [LICENSE](https://github.com/microsoft/franceconnect-facade-dotnet-webapp-aspnetcore/blob/main/LICENSE-CODE) file, and grant you a license to any code in the repository under the MIT License, see the [LICENSE-CODE](https://github.com/microsoft/franceconnect-facade-dotnet-webapp-aspnetcore/blob/main/LICENSE-CODE) file.

Microsoft, Windows, Microsoft Azure and/or other Microsoft products and services referenced in the document may be either trademarks or registered trademarks of Microsoft in the United States and/or other countries. The license for this document does not grant you rights to use any Microsoft names, logos, or trademarks. Microsoft's general trademark guidelines can be found at <http://go.microsoft.com/fwlink/?LinkID=254653>.

Privacy information can be found at <https://privacy.microsoft.com/en-us/>

Microsoft and any contributors reserve all other rights, whether under their respective copyrights, patents, or trademarks, whether by implication, estoppel or otherwise.

Content

[Introduction 5](#_Toc133563907)

[FCF endpoints 5](#_Toc133563908)

[FCF core endpoints 5](#_Toc133563909)

[FCF metadata endpoints 6](#_Toc133563910)

[FCF callback endpoint 11](#_Toc133563911)

[Issuance of tokens through FCF 11](#_Toc133563912)

[Code and tokens issued by FCP 11](#_Toc133563913)

[Expiry of FCP data 14](#_Toc133563914)

[Code and tokens transmitted or issued by FCF 14](#_Toc133563915)

[Expiry of FCF data 16](#_Toc133563916)

[Attributes (claims) available when signing in with FCP 16](#_Toc133563917)

[FCP errors’ management 18](#_Toc133563918)

[FCP error code system 18](#_Toc133563919)

[Support of the FranceConnect UX requirements 20](#_Toc133563920)

[Support for D365 Biz Apps portal 20](#_Toc133563921)

[Support for Azure AD B2C 24](#_Toc133563922)

[Additional references 26](#_Toc133563923)

# Introduction

This Addendum complements the FranceConnect Façade (FCF) technical-functional specifications as defined in the document *FranceConnect Façade (FCF) Specification.docx*.

As of this writing, Dynamics 365 Biz Apps portals, the Power Apps Portals/Power Pages websites[[1]](#footnote-2),[[2]](#footnote-3), as well as Azure AD B2C, cannot integrate with the FranceConnect platform (FCP), while both these offerings and FCP are based on industry standard protocol, namely OpenID Connect (OIDC) - The full specification for OIDC is available on the OpenID Foundation's website at <https://openid.net/specs/openid-connect-core-1_0.html> -.

More specifically, the w/ authorization code flow (without Proof Key for Code Exchange (PKCE)) is being used here, but, as always, the devil resides in detail.

The so-called FranceConnect Façade (FCF), i.e., a dedicated lightweight façade, aims at providing/serving as an adaptation layer between the FranceConnect platform (FCP) and either a service provider such as a D365 Biz Apps portal or an Azure AD B2C user flow with:

* Implementation choices that differ slightly from the [OpenID Connect Core 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-core-1_0.html) standard at the level of the authorization code flow, in particular but not limited to the token signature, see section [3.1 Authentication using the Authorization Code Flow](https://openid.net/specs/openid-connect-core-1_0.html#CodeFlowAuth).
* The absence of a Discovery endpoint in the FranceConnect platform. [Final: OpenID Connect Discovery 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-discovery-1_0.html) standard is not (yet) implemented. One should note that this endpoint is optional in the OpenID Connect (ODIC) standard.

The above hinders the possibility of native interoperability with the Microsoft solutions targeted by the FranceConnect Façade (FCF) specifications and this Addendum.

**This Addendum provides additional details regarding the lightweight façade to implement and can be viewed as detailed specifications.**

# FCF endpoints

## FCF core endpoints

FCF must implement the following endpoints for the OpenID Connect protocol support and the adaptation layer between the client application, i.e., a D365 Biz Apps portal or an Azure AD B2C user flow, calling (FCF, and in turn) FCP:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Method | Endpoint | Comments |
| 1 | GET | /api/<*version*>/authorize | Carried across from the OAuth 2.0 [[RFC6749](https://www.rfc-editor.org/rfc/rfc6749.html)] standard, this endpoint authorizes access a protected resource. Its corollary on FCP will require the user to first authenticate (sign in) and then give their consent. |
| 2 | POST | /api/<*version*>/token | Also carried over from the OAuth 2.0 standard, this endpoint allows the requester to directly retrieve JSON Web Token (JWT) [[RFC 7519](https://www.rfc-editor.org/rfc/rfc7519)] tokens, here an application calling this endpoint, and in turn the façade calling its corollary on FCP. |
| 3 | GET | /api/<*version*>/logout | This optional endpoint as per [Final: OpenID Connect Session Management 1.0](https://openid.net/specs/openid-connect-session-1_0.html) standard allows the user to initiate a (single) logout with FCP. |

In accordance to the above, the structure of the routes also adheres to both Microsoft practices in [Azure Active Directory (Azure AD)](https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis) and the [FranceConnect+ specification](https://github.com/france-connect/Documentation-FranceConnect-Plus/blob/main/fs/docs-fs.md) for versioning purposes to allow later improvements and evolutions of the façade.

/api/<*version*>/xx, where version is globally managed as the interfaces’ version for the endpoint to expose. Current version is beta.

## FCF metadata endpoints

Like modern OpenID Connect identity providers like Azure AD and the [Microsoft identity platform](https://learn.microsoft.com/en-us/azure/active-directory/develop/v2-overview), as well as the [FranceConnect+ platform](https://franceconnect.gouv.fr/france-connect-plus), FCF MUST provide the so-called OpenID Connect Discovery Document (a.k.a. the disco doc) at a publicly accessible endpoint to ease and smooth the integration from a client application’s perspective. Modern client applications even require such an endpoint to be available.

This document contain the FCF's metadata allowing client applications such as a D365 Biz Apps portal or an Azure AD B2C user flow to automatically configure themselves to integrate with this façade acting as a provider.

This includes data such as the location of the above OIDC core endpoints, allowed scopes & claims, and public keys using for token signing, and other metadata. Client applications can use the metadata to discover the URLs to use for authentication and the authentication service's public signing keys, among other things.

FCF must implement the following endpoints for the OpenID Connect Discovery:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Method | Endpoint | Comments |
| 1 | GET | /common/.well-known/openid-configuration | Well-known configuration document path to append to the authority FCF\_URL |
| 2 | GET | /common/discovery/keys | JSON Web Key Set (JWKS) [[RFC7517](https://www.rfc-editor.org/rfc/rfc7517)] path to append to the authority FCF\_URL |

Unlike to the core endpoints, the structure of the routes adheres to a common structure, i.e., /common/xx, regardless of the version globally managed as the interfaces’ version for the above endpoints to expose. Current version is beta.

The next sections depict in order the above listed endpoints.

### /common/.well-known/openid-configuration

A request to get the OpenID configuration metadata from the common authority's OpenID configuration document endpoint on FCF return the following metadata listed in this table:

|  |  |  |
| --- | --- | --- |
| Value | Field\* | Description |
| issuer | REQUIRED | URL using the https scheme with no query or fragment component that FCP asserts as its issuer identifier. This also MUST be identical to the iss claim value in id tokens issued from this façade. |
| authorization\_endpoint | REQUIRED | URL of the FCF's OAuth 2.0 authorization endpoint for authorizing the user and getting in return the code for the id\_token token. The URL is FCF\_URL/api/<*version*>/authorize. |
| token\_endpoint | REQUIRED | URL of the FCF's OAuth 2.0 token endpoint for obtaining the id\_token token. The URL is FCF\_URL/api/<*version*>/token. |
| token\_endpoint\_auth\_methods\_supported | OPTIONAL | JSON array containing a list of client authentication methods as described in section 9. Client Authentication of the [OpenID Connect Core 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-core-1_0.html) standard that FCF supports, currently ["client\_secret\_basic"], i.e., the HTTP basic authentication scheme specified in section 2.3.1. Client Password of OAuth2.0 [[RFC6749](https://www.rfc-editor.org/rfc/rfc6749.html)]. If omitted, the standard specified that it defaults to this method. |
| end\_session\_endpoint | REQUIRED | URL at FCF to which a client application can perform a redirect to request that the end-user be logged out at the FCP. The URL is FCF\_URL/api/<*version*>/logout. |
| jwks\_uri | REQUIRED | URL of the FCF's JSON Web Key Set (JWKS) [RFC] document. This contains the signing key(s) the client application uses to validate signatures from the FCF. The URL is FCF\_URL/api/<*version*>/discovery/keys. See eponym section /api/<*version*>/discovery/keys below. |
| scopes\_supported | RECOMMENDED | JSON array containing a list of the OAuth 2.0 scope values that FCF supports. FCF MUST support the opendid scope value. As per specification, an FCF instance MAY choose not to advertise some supported scope values even when this parameter is used. See section Attributes (claims) available when signing in with FCP below. |
| response\_types\_supported | REQUIRED | JSON array containing a list of the OAuth 2.0 response\_type values that FCF supports, currently ["code"]. |
| response\_modes\_supported | REQUIRED | JSON array containing a list of the OAuth 2.0 response\_mode values that FCF supports, as specified in [Final: OAuth 2.0 Multiple Response Type Encoding Practices](https://openid.net/specs/oauth-v2-multiple-response-types-1_0.html), currently ["query"] to align with [FranceConnect Service Provider Documentation](https://partenaires.franceconnect.gouv.fr/fcp/fournisseur-service). The code is provided as a query string parameter on the client application redirect URI. |
| acr\_values\_supported | OPTIONAL | JSON array containing a list of the authentication context class references (ACR) that FCF supports in respect to the [eIDAS regulation](https://www.ssi.gouv.fr/en/regulation/digital-confidence/the-eidas-regulation/):   * "eidas1": Standard level, e.g., authentication by identifier / password. * "eidas2": Substantial level, e.g., eIDAS approved second factor. * "eidas3": Strong level, e.g., use of eIDAS approved X.509 certificates, smartcard readers, etc.).   If omitted, the default is "eidas3". |
| subject\_types\_supported | REQUIRED | JSON array containing a list of the subject identifier types as described in section 8. Subject Identifier Types of the [OpenID Connect Core 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-core-1_0.html) standard that FCF supports, currently as ["pairwise"] to align with [FranceConnect Service Provider Documentation](https://partenaires.franceconnect.gouv.fr/fcp/fournisseur-service).  FranceConnect provides a different specific sub value to each client application - A end-user will always have the same sub for a given client application, but will have a different sub for each client application he/she uses. -, so as not to enable clients to correlate the end-user's activities without permission. |
| id\_token\_signing\_alg\_values\_supported | REQUIRED | JSON array containing a list of the JSON Web Signature (JWS) [[RFC7515](https://www.rfc-editor.org/rfc/rfc7515)] signing algorithms (alg values) supported by the FCF for the id token to encode the claims in a JWT [[RFC 7519](https://www.rfc-editor.org/rfc/rfc7519)]. The algorithm RS256 is the only one supported. |
| claim\_types\_supported | REQUIRED | JSON array containing a list of the claim types that FCF supports, currently ["normal"]. The claim types are described in section 5.6 Claims Types of the [OpenID Connect Core 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-core-1_0.html) standard.  Values defined by this Addendum are normal and aggregated. Strictly speaking, should be aggregated: claims that are asserted by FCP, a claims provider other than FCF but are returned by FCF. But as per section 5.6.2 Aggregated and Distributed Claims of the standard, aggregated claims are represented by using special \_claim\_names and \_claim\_sources members of the JSON object containing the claims. As the façade SHOULD be as transparent as possible, normal claims are used instead. |
| claims\_supported | RECOMMENDED | See section Attributes (claims) available when signing in with FCP below. Also note that for privacy or other reasons, this might not be an exhaustive list. |
| claims\_locales\_supported | OPTIONAL | Languages and scripts supported for values in claims being returned, represented as a JSON array of BCP47 [[RFC5646](https://www.rfc-editor.org/rfc/rfc5646)] language tag values. Not all languages and scripts are necessarily supported for all claim values. |

\* The Field status doesn’t necessarily reflect the one defined in the [Final: OpenID Connect Discovery 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-discovery-1_0.html) standard.

The following is an example of such a metadata document:

{

"issuer":"https://b8d7-86-238-58-229.ngrok.io",

"authorization\_endpoint":" b8d7-86-238-58-229.ngrok.io/api/beta/authorize",

"token\_endpoint":"https://b8d7-86-238-58-229.ngrok.io/api/beta/token",

"end\_session\_endpoint":" https://b8d7-86-238-58-229.ngrok.io/api/beta/logout",

"jwks\_uri":"https://b8d7-86-238-58-229.ngrok.io/api/beta/discovery/keys",

"scopes\_supported": [

"openid",

"email",

"profile",

"identite\_pivot"

],

"response\_types\_supported": [

"code"

],

response\_modes\_supported": [

"query"

],

"acr\_values\_supported": [

"eidas1",

"eidas2",

"eidas3"

],

"subject\_types\_supported": [

"pairwise"

],

"id\_token\_signing\_alg\_values\_supported": [

"RS256"

],

"claim\_types\_supported": [

"normal"

],

"claims\_supported": [

"sub",

"amr",

"gender",

"birthdate",

"birthcountry",

"birthplace",

"given\_name",

"family\_name",

"email",

"preferred\_username",

"acr",

"sid",

"auth\_time",

"iss"

],

"claims\_locales\_supported":

"fr-fr"

]

}

### /common/discovery/keys

A request to this endpoint on FCF returns in turn a JSON Web Key (JWK) Set [[RFC7517](https://www.rfc-editor.org/rfc/rfc7517)], i.e., a JSON object that represents a set of JWKs. This JSON object MUST have a "keys" member, with its value being an array of JWKs. This JSON object MAY contain whitespace and/or line breaks.

A JSON Web Key (JWK) [[RFC7517](https://www.rfc-editor.org/rfc/rfc7517)] is a JSON object that represents a cryptographic key with the following keys.

|  |  |  |
| --- | --- | --- |
| Value | Field\* | Description |
| kty | REQUIRED | Key type parameter that identifies the cryptographic algorithm family used with the key. "RSA" is the only value supported by FCF. |
| use | REQUIRED | Public key use parameter that identifies the intended use of the public key. Value supported by FCF is "sig" (signature). This value is a case-sensitive ASCII string. |
| alg | OPTIONAL | Algorithm parameter that identifies the algorithm intended for use with the key as per JSON Web Algorithm (JWA) [[RFC7518](https://www.rfc-editor.org/info/rfc7518)]. Value supported by FCF is “RS256". This value is a case-sensitive ASCII string. |
| kid | REQUIRED | Key ID parameter is used to match a specific key. This is used, for example, to choose among a set of keys within a JWK Set during key rollover. In [RFC7517](https://www.rfc-editor.org/rfc/rfc7517), the structure of the "kid" value is unspecified. When "kid" values are used within a JWK Set, different keys within the JWK Set SHOULD use distinct "kid" values. So, when defined, the "kid" values SHOULD correspond to the "x5t" values, see below. |
| x5t | OPTIONAL | X.509 certificate SHA-1 thumbprint parameter that is a base64url-encoded (see section [5. Base 64 Encoding with URL and Filename Safe Alphabet](https://www.rfc-editor.org/rfc/rfc4648#section-5) of [RFC4648](https://www.rfc-editor.org/rfc/rfc4648)) SHA-1 thumbprint (a.k.a. digest) of the DER encoding of an X.509 certificate [[RFC5280](https://www.rfc-editor.org/rfc/rfc5280)]. |
| n | REQUIRED | Modulus n parameter that contains the modulus value for the RSA public key as per section [6.3.1.1. "n" (Modulus) Parameter](https://www.rfc-editor.org/rfc/rfc7518#section-6.3.1.1) of [RFC7518](https://www.rfc-editor.org/rfc/rfc7518). It is represented as a Base64urlUInt-encoded value (see section [5. Base 64 Encoding with URL and Filename Safe Alphabet](https://www.rfc-editor.org/rfc/rfc4648#section-5) of [RFC4648](https://www.rfc-editor.org/rfc/rfc4648)). |
| e | REQUIRED | Exponent e parameter that contains the exponent value for the RSA public key as per section [6.3.1.2. "e" (Exponent) Parameter](https://www.rfc-editor.org/rfc/rfc7518#section-6.3.1.1) of [RFC7518](https://www.rfc-editor.org/rfc/rfc7518). It is represented as a Base64urlUInt-encoded value. For example, when representing the value 65537, the octet sequence to be base64url-encoded MUST consist of the three octets [1, 0, 1]; the resulting representation for this value is "AQAB". |
| x5c[[3]](#footnote-4) | REQUIRED | X.509 certificate chain parameter that contains a chain of one or more X.509 certificates [[RFC5280](https://www.rfc-editor.org/rfc/rfc5280)]. The certificate chain is represented as a JSON array of certificate value strings. Each string in the array is a base64-encoded (see section [4. Base 64 Encoding](https://www.rfc-editor.org/rfc/rfc4648#section-4) of [RFC4648](https://www.rfc-editor.org/rfc/rfc4648)) - not base64url-encoded - DER X.509 certificate value. See [PEM, DER, CRT, and CER: X.509 Encodings and Conversions](https://www.ssl.com/guide/pem-der-crt-and-cer-x-509-encodings-and-conversions/) for information on how to convert formats and encoding from one to another.  The X.509 certificate containing the key value MUST be the first certificate. This MAY be followed by additional certificates, with each subsequent certificate being the one used to certify the previous one. The key in the first certificate MUST match the public key represented by other members of the JWK. |

\* The Field status doesn’t necessarily reflect the one defined in the [RFC7517](https://www.rfc-editor.org/rfc/rfc7517) or the [RFC7518](https://www.rfc-editor.org/rfc/rfc7518).

The following is an example of a JWK Set (JKWS):

{

keys: [

    {

      "kty": "RSA",

      "alg": "RSA256",

      "use": "sig",

      "kid": "CCD9D193A73570B741CB341EB26FE648C743DB69",

      "n": "7FS4NnTgTglUTPapugS9XTIGRvBrY3y6Hx1q799HTkxx6r8PGyhrl2riCQEk8oUmXalvTVj5Pz3PW14fw0R5s/shJOxhRn2E6Tc5T2AHXQKyCh3I6o7hBIs/PLnKpk9o+4xO+r8XIkex1iCtOMtRR/T2LBp8GLcM23nj6t5ni9HPIWqNC5Kwjrt2HWllETCHILt7wWawuVdcjkaF4fEbC8IM0uLGbCVSgYBKdRbF1Rjgbc12C32cKWUQ43HFrrpn1H7ZhpfeIAJ4o7rtoT3W2rgNLN0NHZCi+q+PU90WaCX2EAsSiyUt2NTlgpYkiQc+G06C4RWWybs5yAkKgk3aVQ==",

      "e": "AQAB",

      "x5c": [

""

]

}

]

}

## FCF callback endpoint

When FCF builds the query for the authorize request it must fill in a key to stipulate the redirect Uri. The endpoint */login-callback* is given as the value of the redirect Uri. Therefore, as soon as the credentials are sent to FranceConnect by the user browser, FranceConnect will send a token and a state to the FCF. The façade will then in turn send a redirection with the code and state granted by FranceConnect to the user browser.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Method | Endpoint | Comments |
| 1 | GET | /login-callback | Retrieves the parameters returned by FranceConnect and redirects to the Portal/Page application. |

# Issuance of tokens through FCF

As extensively covered in the aforementioned technical-functional specifications, this facade is therefore intended to receive the code and JSON Web Token (JWT) tokens, i.e., an open, industry standard [RFC 7519](https://www.rfc-editor.org/rfc/rfc7519) method for representing claims securely between two parties, issued by the FranceConnect platform to in turn issue JWT tokens to a client application: here either a D365 Biz Apps Portal or an Azure AD B2C user flow.

This section describes the related formats.

## Code and tokens issued by FCP

### Authorization code

This code is returned (in the URL) by FCP to the client application. The latter makes in turn a call on the *FCP\_URL/api/v1/authorize* endpoint through FCF. This code is then passed (in the body of the HTTP POST request) when calling *the FCP\_URL/api/v1/token* endpoint.

### Access token

The access token is issued by FCP to the client application: it is returned (in the HTTP body) by the call to the FCP\_*URL/api/v1/token* endpoint. The client passes it to FCP through FCF by a call to the FCP\_*URL/api/v1/token* endpoint.

This token contains the permissions the client has been granted by FCP based on its client ID, a.k.a. application ID. The client ID uniquely identifies the application in FCP and is included in the security tokens FCP issues. The same client ID (and secret ID) is being used by FCF as a passthrough mechanism. From the FCF perspective, FCF is the client application.

**FCF must treat access tokens issued by FCP as opaque strings because the contents of the token are intended for the intended recipient only:**

* For validation and debugging purposes only, developers can decode JWTs using a site like <https://jwt.ms>.
* For details on what's inside the access token, one should use the token response data that's returned with the access token to FCF/the client application.

### id\_token token

The client application uses id\_token tokens when signing in users and to get basic information about them. This token in JWT format, i.e. a formatted and signed JSON object, is returned when calling the *FCP\_URL/api/v1/token* endpoint.

JWTs are split into three pieces:

1. The header. Provides information about how to validate the token including information about the type of token and how it was signed.
2. The payload. Contains all of the important data about the user or application that's attempting to call the service.
3. The signature. Is the raw material used to validate the token.

Each piece is separated by a period (.) and separately Base64 encoded.

Graphical user interface, text, application

Description automatically generated

#### Header claims

The table below shows header claims present in FCP id\_token tokens.

|  |  |  |
| --- | --- | --- |
| Type | Format | Description |
| typ | String – always JWT. | Indicates that the token is a JWT token. |
| alg | String | Indicates the algorithm that was used to sign the token. Here HS256. See below |

#### Payload claims

The payload contains six fields:

* The mandatory aud, exp, iat, iss, sub fields as described in the [OpenID Connect Core 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-core-1_0.html) standard.
* The nonce is a mandatory parameter sent when calling *FCP\_URL/api/v1/authorization*. FCF (on behalf of the client application) must verify that the value corresponds to the one it sent, which must be linked to the user's session.

The table below shows the claims that are in FCP id\_token tokens.

|  |  |  |
| --- | --- | --- |
| Type | Format | Description |
| iss | String, an issuer URI. | Identifies the issuer that constructs and returns the token and correspond to the FCP\_URL. For example, <http://fcp.interg01.dev-franceconnect.fr> for the integration environment. |
| sub | String | Indicates the principal about which the token asserts information, i.e., the technical identifier of the user in FCP. This value is immutable and cannot be reassigned or reused. The subject is a pairwise identifier - it is unique to a particular client ID, a.k.a. application ID. If a single user signs into two different applications using two different client IDs, those apps will receive two different values for the subject claim. |
| aud | String | Identifies the intended recipient of the token. In id\_token tokens, the audience is the application's client ID. This value should be validated. The token should be rejected if it fails to match the application's client ID attributed by the [FranceConnect registration process](https://franceconnect.gouv.fr/partenaires). |
| exp | int, a Unix timestamp | Identifies the expiration time on or after which the JWT must not be accepted for processing. It's important to note that in certain circumstances, a resource may reject the token before this time. |
| iat | int, a Unix timestamp | Indicates when the authentication for this token occurred: "Issued At". |
| nonce | String | Should match the parameter included in the original /authorize request to FCP. If it does not match, client application/FCF should reject the token. |
| idp | String | Records the identity provider that authenticated the subject of the token. Marked as FC. |
| acr | String | Indicates the authentication context class reference value that identifies the authentication performed satisfied, namely eidas1, eidas2 or eidas3 |
| amr | String | Indicates the identifiers for authentication methods used in the authentication. The particular values to be used are beyond the scope of this Addendum. They aren’t specified in the [FranceConnect Service Provider Documentation](https://partenaires.franceconnect.gouv.fr/fcp/fournisseur-service). |

### id\_token\_hint token

This token in JWT format is passed as a parameter during the call to *FCP\_URL/api/v1/logout*. It is identical to the id\_token token that was received during the exchange with the call to *FCP\_URL/api/v1/token*.

### Token signature & encryption

All JWT token exchanges between FCP and FCF are signed using the symmetric HS256 algorithm ([HMAC](https://en.wikipedia.org/wiki/HMAC) avec [SHA-256](https://en.wikipedia.org/wiki/SHA-256)). The exchanged tokens are NOT encrypted.

## Expiry of FCP data

|  |  |  |  |
| --- | --- | --- | --- |
| Type | | Description | Lifetime (in seconds) |
| *Web session* | At each sign-in request (authentication) and to maintain the session on FCP. | | 1800 |
| Authorization code | Code provided at the beginning of the sign-in process and used to retrieve the access token. | | 30 |
| *Access token* | User attribute (a.k.a. claims) retrieval | | 60 |
| Consent | Consent given by the user for access to a protected resource (associated with a scope in the OAuth2 sense) | | 5 |

## Code and tokens transmitted or issued by FCF

### Access token

This code received from FCP is returned (in the URL) by FCF to the client application when the latter makes a call on the *FCF\_URL/api/v1/authorize endpoint*. This code is then passed (in the body of the HTTP POST request) when calling the *FCF\_URL/api/v1/token* endpoint, which in turn invokes the *FCP\_URL/api/v1/token endpoint*.

It is passed as such on these calls.

### Access token

This token must be considered as opaque. It serves to call the UserInfo endpoint to retrieve claims in addition to the ones conveys in the id\_token token as per section 5.5. Requesting Claims using the “claims” Request Parameter in the [OpenID Connect Core 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-core-1_0.html) standard.

### id\_token token

#### header claims

The table below shows header claims that must be present in id\_token tokens issued by FCF.

|  |  |  |
| --- | --- | --- |
| Type | Format | Description |
| typ | String – always JWT. | Indicates that the token is a JWT token. |
| alg | String | Indicates the algorithm that was used to sign the token. Here RS256. See below |
| kid | String | Specifies the thumbprint for the public key (certificate) that can be used to validate this token's signature. |

#### payload claims

The table below shows the claims that must be present in id\_token tokens issued by FCF.

|  |  |  |
| --- | --- | --- |
| Type | Format | Description |
| iss | String, an issuer URI. | Identifies the issuer that constructs and returns the token and correspond to the FCF\_URL. For example, <https://b8d7-86-236-58-229.ngrok.io> for a locally instantiated FCF. |
| sub | String | Same as the one received from FCP. Indicates the principal about which the token asserts information, i.e., the technical identifier of the user in FCP. |
| aud | String | Same as the one received from FCP. Identifies the intended recipient of the token. In id\_token tokens, the audience is your app's Client ID. This value should be validated. The token should be rejected if it fails to match your app's Client ID. |
| exp | int, a Unix timestamp | Same as the one received from FCP. Identifies the expiration time on or after which the JWT must not be accepted for processing. It's important to note that in certain circumstances, a resource may reject the token before this time.. |
| iat | int, a Unix timestamp | Same as the one received from FCP. Indicates when the authentication for this token occurred: "Issued At". |
| nonce | String | Same as the one received from FCP. Should match the parameter included in the original /authorize request to FCP. If it does not match, client application/FCF should reject the token. |
| nbf | String | Identifies the time before which the JWT MUST NOT be accepted for processing: “Not Before”. Same as iat. |
| given\_name | String | Optional attribute of the so-called pivot identity depending on the scopes and aliases passed to FCP. See section Attributes (claims) available when signing in with FCP below. |
| family\_name | String | Ibid |
| birthdate | String | Ibid |
| gender | String | Ibid |
| birthplace | String | Ibid |
| birthcountry | String | Ibid |
| email | String | Ibid |
| preferred\_username | String | Ibid |
| identity\_provider | String | Set to “franceconnect” by FCF to attest a successful sign-in at FCP through FCF. Can be later leveraged by a client application to customize the UX. |

Following is an example of such a token decoded with <https://jwt.ms>:

Une image contenant texte

Description générée automatiquement

### id\_token\_hint token

See eponym section above for FCP.

### Token signature & encryption

**All JWT token exchanges between FCF and a client application are signed using the RS256 asymmetric signature algorithm (RSA with SHA-256).**

**The public signature keys of FCF are available via the** [JSON Web Key Set (JWKS)](https://www.rfc-editor.org/rfc/rfc7517) **URL present in the above mentioned** OpenID Connect Discovery Document (a.k.a. the disco doc) exposed by FCF as per [Final: OpenID Connect Discovery 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-discovery-1_0.html) standard.

**The tokens are not encrypted.**

## Expiry of FCF data

Same as FCP. See section Expiry of FCP data above.

# Attributes (claims) available when signing in with FCP

The so-called “pivot identity” in the FranceConnect language identifies a particular user in FranceConnect. The user attributes (a.k.a. claims) returned by FCP for this pivot identity are based on the [OpenID Connect (OIDC) scopes mechanism](https://openid.net/specs/openid-connect-core-1_0.html#ScopeClaims).

To provide greater modularity in these attributes, FCP extends and adapts this mechanism as follows:

* Only one scope is mandatory: openid. This scope value is used to retrieve a unique technical user identifier, i.e., the sub, for each user.
* The standard OIDC scope profile is supported. It is optional. This scope value retrieves the following attributes (claims) related to the user profile: family\_name, given\_name, gender, birthdate, and if available preferred\_username. Not all attributes defined in the standard for this scope are returned. This concerns the following attributes: name, middle\_name, nickname, profile, picture, website, zoneinfo, locale, and updated\_at.
* It is possible to retrieve each attribute of the pivot identity individually using their dedicated scopes - the scopes are case sensitive, their value is string - :

|  |  |
| --- | --- |
| Scope | Description |
| openid | Technical identifier (sub) of the user (OIDC standard) |
| given\_name | First names separated by spaces ([details](https://xml.insee.fr/schema/cog.html#CodePaysOuTerritoireEtranger_stype)) |
| family\_name | Name of birth ([details](http://xml.insee.fr/schema/commun.html#ChaineFrancaisOfficielMajuscule_stype)) |
| birthdate | Date of birth ([details](http://xml.insee.fr/schema/commun.html#DateSouple_stype)) |
| idp\_birthdate | Date of birth as provided by the identity provider without transformation ([details](http://xml.insee.fr/schema/commun.html#DateSouple_stype)) |
| gender | male or female |
| birthplace | INSEE code of place of birth ([list of codes](https://www.insee.fr/fr/information/3363419), [detail](http://xml.insee.fr/schema/cog.html#CodeCommuneFrancaiseOuPaysOuTerritoireEtranger_stype)). Empty in case of foreign country |
| birthcountry | INSEE code of country of birth ([details](http://xml.insee.fr/schema/cog.html#CodePaysOuTerritoireEtranger_stype), for France: 99100) |
| email | Email address in [RFC 5322](https://www.rfc-editor.org/rfc/rfc5322) format ([details](http://xml.insee.fr/schema/cog.html#CodePaysOuTerritoireEtranger_stype)) |
| preferred\_username | Name of use ([details](http://xml.insee.fr/schema/commun.html#ChaineFrancaisOfficielMajuscule_stype)) |

* In accordance with the OIDC standard, it is possible to combine multiple scopes of one's choice to retrieve only those attributes that are needed for user information, by creating a space-delimited list of scope values.

The following is an example of an unencoded scope request:

scope=openid email

As such:

* + The profile alias corresponds to the OIDC standard profile scope as described above. It retrieves the following attributes (claims): openid, given\_name, family\_name, gender, preferred\_username, and birthdate.
  + The birth alias groups the attributes/scopes birthplace and birthcountry which allows to retrieve the city and the department of birth of the user.
  + The identite\_pivot alias combines the profile and birth scopes and retrieves the complete pivot identity plus the user name if available: openid, given\_name, family\_name, gender, preferred\_username, birthdate, birthcountry, and birthplace.

The claims requested by the profile and  email scope values are returned from the UserInfo endpoint, as described in the [OpenID Connect Core 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-core-1_0.html) standard  in section [5.3.2 Successful UserInfo Response](https://openid.net/specs/openid-connect-core-1_0.html#UserInfoResponse) when a response\_type value is used that results in an access token being issued.

On that basis, FCF supports the following scopes and/or alias: openid, email, profile, and identite\_pivot. See section /api/<*version*>/.well-know/openid-configuration above.

# FCP errors’ management

Acting as an OpenID Connect (OIDC) identity provider from FCF perspective, FCP can potentially return all kinds of errors to a client application, here FCF.

To do so, FCP relies on the errors’ return mechanism as described in the [OpenID Connect Core 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-core-1_0.html) standard for an identity provider, see the following sections, and proposes in this context an error code system to make the detection and management of bugs easier:

* [3.1.2.6 Authentication Error Response](http://openid.net/specs/openid-connect-core-1_0.html#AuthError): The following is a non-normative example error response :

HTTP/1.1 302 Found

Location: https://fcp.integ01.dev-franceconnect.fr/api/v1/authorize?

error=invalid\_request

&error\_description=FranceConnectErrorCode

&state=some\_state\_value

* [3.1.3.4 Token Error Response](http://openid.net/specs/openid-connect-core-1_0.html#TokenErrorResponse): The following is a non-normative example error response :

HTTP/1.1 400 Bad Request

Content-Type: application/json

Cache-Control: no-store

Pragma: no-cache

{

"error": "FranceConnectErrorCode"

}

* [5.3.3 UserInfo Error Response](https://openid.net/specs/openid-connect-core-1_0.html#UserInfoError): The following is a non-normative example error response :

HTTP/1.1 401 Unauthorized

WWW-Authenticate: error=FranceConnectErrorCode

## FCP error code system

As such, the FCP error code system is broken down into four areas:

1. Core,
2. [RNIPP (stands for “Répertoire national d’identification des personnes physiques” in French)](https://www.cnil.fr/fr/rnipp-repertoire-national-didentification-des-personnes-physiques-0), i.e., relating to the national directory of identification of physical persons, a French directory, held by INSEE since 1946, which lists all living and deceased persons, born in France or born abroad and coming to work in France.
3. Identity Provider,
4. [electronic IDentification, Authentication and trust Services(eIDAS)](https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation) interoperability.

**As indicated in the FCP technical-functional specification, all the error codes supported by the above system are by design passed to FCF, which in turn will be passed to the calling application.**

**These error codes are listed in the following section on a per area basis.**

### Core

|  |  |  |
| --- | --- | --- |
| Code | | Description |
| *E000000* | Exception mal ou non gérée. A remonter impérativement. | |
| E000001 | Compte désactivé, connexon impossible via FranceConnect à moins de réactiver le compte. | |
| *E000009* | Compte désactivé, connexon impossible via FranceConnect à moins de réactiver le compte. | |
| E000035 | Connexion désactivée par l'usager pour le fournisseur d'identité actuel. | |

### RNIPP

|  |  |
| --- | --- |
| Code | Description |
| E010000 | Exception mal ou non gérée. A remonter impérativement. |
| E010004 | Demande non identifiée mais existence d'un seul écho. |
| E010006 | Demande non identifiée mais existence de plus d'un écho. |
| E010007 | Demande identifiée avec le nom d'usage uniquement. |
| E010008 | Demande non identifiée sans écho. |
| E010009 | Demande rejetée au contrôle en raison d'erreurs de syntaxe. |
| E010011 | Délai de réponse dépassé pour le RNIPP |
| E010012 | Mauvais format retourné, qui n'est pas du XML, mais une généralement une page HTML renvoyée avec le [code http](https://fr.wikipedia.org/wiki/Liste_des_codes_HTTP) correspondant. |
| E010013 | XML bien formaté mais avec des informations manquantes dans l'identité renvoyée. P. ex. lieu de naissance vide pour les personnes nées à l'étranger. |
| E010014 | Différence de la date de naissance envoyée par le FI et celle renvoyée par le RNIPP. |
| E010015 | Information "décédée" retournée par le RNIPP pour la personne/ |

### Identity Provider

|  |  |
| --- | --- |
| Code | Description |
| E020000 | Exception mal ou non gérée. A remonter impérativement. |
| E020001 | Code http 404 (ressource non trouvée) retourné par le FI |
| E020002 | Informations personnelles fournies incomplètes ne permettant pas de certifier l'identité. |
| E020003 | Problème de format pour un des champs reçu du FI |
| E020004 | Adresse mèl de l'usager qui se connecte non envoyée par le FI. Dans le cadre de l'interopérabilité eIDAS, l'e-mail n’est pas envoyé la majorité du temps car non obligatoire. |
| E020005 | champ sub non présent dans l'objet renvoyé par le FI |
| E020006 | Enregistrement en base du « compte » de l'usager impossible |
| E020007 | Contenu retourné par le FI non valide |
| E020008 | Code http 401 (utilisateur non authentifié) retourné par le FI |
| E020009 | Code http 500 (erreurs serveur) retourné par le FI |
| E020010 | Code http 502 (erreurs serveur) retourné par le FI |
| E020011 | Code http 503 (erreurs serveur) retourné par le FI |
| E020012 | Niveau eIDAS fourni par le FI supérieur à celui déclaré dans la plateforme FranceConnect |
| E020017 | Appel d’un FI non activé. |
| E020018 | Absence de réponse du FI |
| E020019 | FI non déclaré dans la base de données de la plateforme FranceConnect |
| E020020 | Absence de contexte lors de l'appel à la route */oidc\_callback* (comprendre hors cinématique de connexion). |
| E020021 | Paramètre manquant à la route */oidc\_callback* (code et/ou state). Due en général à une mauvaise implémentation du FI. |
| E020022 | Paramètre state non valide. |

### eIDAS interoperability

|  |  |  |
| --- | --- | --- |
| Code | | Description |
| *E050000* | Exception mal ou non gérée. A remonter impérativement. | |
| E050001 | Lien réutilisé ou rafraîchi trop tard sur la page "choix du pays" ou cookies désactivés sur le navigateur de l'usager | |
| *E050002* | Erreur de traitement de la réponse SAML du FS (ou du FI ?) | |

# Support of the FranceConnect UX requirements

## Support for D365 Biz Apps portal

### Support of the FranceConnect button in the UI

As per FranceConnect specification, **the authentication flow is initiated by clicking the so-called FranceConnect button.**

The support of the FranceConnect button is MANDATORY. As outlined in the [FranceConnect Service Provider Documentation](https://partenaires.franceconnect.gouv.fr/fcp/fournisseur-service): “The FranceConnect action buttons are essential in the use of the service. It is mandatory to use one of the proposed buttons and no other visual for the connection of users.”

 

The .svg and .png files to use for the button can be download here: <https://partenaires.franceconnect.gouv.fr/files/fc_boutons.zip>.

The expected user experience (UX) is fully covered on the FranceConnect partner’s portal at <https://partenaires.franceconnect.gouv.fr>, under in your project page, once your project’s submission for accreditation is validated. Please refer to the so-called section [Recetter](https://partenaires.franceconnect.gouv.fr/monprojet/recetter/).

All the below implementation choices regarding Dynamics 365 Biz Apps portals, the Power Apps Portals/Power Pages websites[[4]](#footnote-5),[[5]](#footnote-6) have been made with the goal of easing the fulfillment of such requirements.

These choices are illustrated hereafter from the perspectives of [Microsoft Power Pages](https://powerpages.microsoft.com/), which should (hopefully) encompasses as of this writing the configuration that pertains to the notion of D365 Biz Apps portal.

As such, Power Pages offers the possibility to save images and documents through [Web Files](https://learn.microsoft.com/en-us/power-apps/maker/portals/configure/web-files). This feature will be leveraged to store the FranceConnect button’s images in .svg format. These images will thus be available at any time at an publicly available address in the following format: <*portal\_url/partial\_url\_of\_the\_button*>.

By default, Power Pages groups all external identity providers in the same section. However, the FranceConnect UX clearly requires that this FranceConnect button must be distinguishable from other external sign-in methods:

“*FranceConnect n’est pas un réseau social et ne doit pas être présenté ou susceptible d’être perçu comme tel par l’utilisateur. Pour éviter toute confusion, FranceConnect ne doit pas figurer près de liens, d'icônes ou de services d'identification proposés par des réseaux sociaux et/ou autres services similaires*.”

In plain English: FranceConnect is not a social network and shall not be presented or likely to be perceived as such by the user. To avoid confusion, FranceConnect should not appear near links, icons or identification services offered by social networks and/or other similar services.

Consequently, the layout must be rearranged in an appropriate manner, and the default HTML code touched. This will be specifically the purpose of two scripts named *signin.ejs* and *register.ejs* in order to rearrange the buttons, so that the FranceConnect button is well isolated from the rest and discernible.

To this aim, [jQuery](https://jquery.com/) is a JavaScript library which offers the possibility to easily access and modify HTML elements. The front-end UI that displays the login and registration pages will be overridden by the content snippets *Account/SignIn/PageCopy* and *Account/Register/PageCopy*.

Here is an example of such a code to later copy/paste into *Account/SignIn/PageCopy* on a Power Pages site where [*FRANCECONNECTFACADE URL*] will be replaced by the effective FranceConnect Façade (URL):

<script>

$(document).ready(function(){

// Personalisation du bouton FranceConnect

$("#[FRANCECONNECTFACADE URL]").addClass("fc-button");

$(".fc-button").addClass("fc-button-background");

$(".fc-button").removeClass("btn btn-primary btn-line");

$(".fc-button").text("");

// La notion de compte FranceConnect ne doit pas apparaître

$(".fc-button").prop("title", "");

$("#mainContent").prepend('<div class="fc-signin"></div>');

$(".fc-signin").append('<form id="fc-form" action="/Account/Login/ExternalLogin?ReturnUrl=%2F" method="post"><input name="\_\_RequestVerificationToken" type="hidden" value="ssx\_iSw4ax0ZlLr1vCBABRmyBSqC9OV\_vTCBIz9r9uaxfkWesrmCixIU0J\_ZSoQ4P-uT6fwRHmWGNrvlTASewYbLXI9x9AkRgWXH1a9ZZik1">');

$("#fc-form").append('<div class="form-group fc-container form-horizontal"></div>');

$("#fc-form").append('<div class="form-group span-container"></div>');

// Ajout des éléments dans les nouvelles div

$(".fc-container").append('<p class="fc-text-info">FranceConnect est la solution proposée par l\'État pour sécuriser et simplifier la connexion à vos services en ligne.</p>')

$(".fc-button").appendTo(".fc-container");

$(".fc-container").append('<br>');

$(".fc-container").append('<a class="fc-redirect" href="https://franceconnect.gouv.fr/">Qu’est-ce que FranceConnect ?</a>')

$(".span-container").append('<span class="xrm-editable-text xrm-attribute" data-languagecontext="français"><span class="xrm-attribute-value-encoded xrm-attribute-value">OU</span></span>');

});

</script>

This JavaScript code - You can find all the methods used in the [jQuery API documentation](https://api.jquery.com/)- takes into account the following considerations:

* When creating buttons for external identity providers, Power Apps assigns a unique identifier to each button. These identifiers are not configurable and correspond to the Authority field, i.e., in other words, the FCF URL that will be specified when creating/configuring an Identity Provider for the Façade. So, the id is therefore a URL. In the case of a local deployment with ngrok – see guide Getting Started with the FranceConnect Facade (FCF) - A developer guide to configure and test the facade -, this URL will be very long. Moreover, it is necessary to take care to escape each special character as described in [jQuery selectors](https://api.jquery.com/category/selectors/).
* The text button is set to an empty string since the default text set by Power Pages is “*Connectez-vous avec votre compte FranceConnect*” (Sign in with your FranceConnect account).
* The notion of FranceConnect account should not appear according to the specifications “*Merci de ne pas faire apparaître la notion de "compte FranceConnect". Il n'existe pas de compte FranceConnect*” (Please do not show the notion of "FranceConnect account". There is no such FranceConnect account).

Thus, for a local deployment of the FranceConnect Facade, the line of code

$("#[FRANCECONNECTFACADE URL]").addClass("fc-button");

will for instance become:

$("#https\\:\\/\\/ceb3\\-2a01\\-cb04\\-b87\\-6600\\-99ca\\-979b\\-18cb\\-b20f\\.eu\\.ngrok\\.io\\/").addClass("fc-button");

And will allow i) a reference to the so-called FranceConnect button defined by Power Pages, and ii) add a class fc-button, which will be an easier way to access this button in the rest of the html code file, a well as in the css file. Please note that jQuery needs the selector “#” to access the element containing the correspond id, this is why the symbol “#” is added in front of the URL, see [jQuery ID Selector](https://api.jquery.com/id-selector/).

In the rest of the code, a new html div will be created to contain the FranceConnect button, its description and the link « *Qu’est-ce que FranceConnect ?* » (What is FranceConnect?).

Likewise, the file *Account/Register/PageCopy* will contain the same kind of changes, with one noticeable slight update here: Power Pages does not put the FCF URL in the button id but in its attribute value. So, the code must be slightly different to this end.

<script>

…

$('[value="[FRANCECONNECTFACADE URL]"]').addClass("fc-button");

…

</script>

For example, for a local deployments with ngrok:

<script>

…

$('[value="https\\:\\/\\/ceb3\\-2a01\\-cb04\\-b87\\-6600\\-99ca\\-979b\\-18cb\\-

b20f\\.eu\\.ngrok\\.io\\/"]').addClass("fc-button");

…

</script>

Adding the class fc-button to both buttons allows to factorize the code in *.css* file, namely the file *franceconnect.css*, and the same style will be ultimately applied to both buttons.

Once the html code files are modified, the custom file *franceconnect.css* will be prioritized over the Power Pages default file *theme.css*, which manages among others the style of the buttons.

The following code written at the beginning of the file *franceconnect.css* will be used to preload the .svg images of the FranceConnect buttons to avoid display issues of any kind:

body::after {

position: absolute;

width: 0;

height: 0;

overflow: hidden;

z-index: -1;

content: url('[FRANCECONNECT BUTTON URL]') url('[FRANCECONNECT BUTTON HOVER URL]');

}

### Support of the FranceConnect profile in the UI

The FranceConnect profile SHOULD be also optionally supported in the UI. To achieve this, Power Pages must know which login method was used by the user.

To this extend, an imported solution *FranceConnect\_1\_0\_0\_1\_managed.zip* on Power Apps will enable to add a new column Identity Provider to the Contact table.

The Microsoft Dataverse contact record contains all the information related to the users and once a user is signed in, the Power Pages portal will have access to the user's data through the [Liquid Object user](https://learn.microsoft.com/en-us/power-pages/configure/liquid/liquid-objects#user).

This will allow the conditional display of the FranceConnect profile in the HTML *header* document according to whether the user has connected via FranceConnect:

{% if user.msf\_identityprovider == "franceconnect" %}

<div id="fconnect-profile" data-fc-logout-url="/logout">

<a href="#" class="has-text-link">

{{ username | default: resx['Default\_Profile\_name'] }}

</a>

</div>

Only the FCF can fill the column Identity Provider when it issues and sends back the token with the claims to the portal. The Power Pages site will automatically add the value of the identity\_provider claim to the corresponding Dataverse Contact thanks to the following mapping:

msf\_identityprovider=identity\_provider

in Registration claims mapping and Login claims mapping when configuring the FranceConnect identity provider on Power Pages.

### Unsupported FranceConnect UI features

In the current configuration, the Identity Provider column dedicated to knowing if the user has signed in using FranceConnect is immutable. This means that in the scenario, where a user creates a local account, and then signs in with FranceConnect, all subsequent sign-ins using the local account will have the FranceConnect profile displayed in so far as the Identity Provider column that binds to the user will have been set to “*franceconnect”*.

One way to overcome this slight issue may consist in defining an Automated Cloud Flow on [Power Automate](https://make.powerautomate.com/), which will empty the *Identity Provider* column of the user at his logout.

## Support for Azure AD B2C

### Support of the FranceConnect button in Azure AD B2C

By default, Azure AD B2C provides templates for branding and customizing the user interface that Azure AD B2C displays to your customers in order to help providing a seamless user experience (UX). These experiences include signing up, signing in, profile editing, and password resetting through predefined [user flows](https://learn.microsoft.com/en-us/azure/active-directory-b2c/user-flow-overview) or through fully configurable [custom policies](https://learn.microsoft.com/en-us/azure/active-directory-b2c/custom-policy-overview).

The templates for signing up and signing in, and/or signing in can be leveraged to include the so-called FranceConnect button and the related UX that pertains to it as per signing up, signing in as per [Customize the user interface with HTML templates in Azure Active Directory B2C](https://learn.microsoft.com/en-us/azure/active-directory-b2c/customize-ui-with-html?pivots=b2c-user-flow).

However, in the template, the button for FranceConnect is displayed among the social accounts, which does not fulfill the FranceConnect specifications in terms of UX, which is fully covered on the FranceConnect partner’s portal at <https://partenaires.franceconnect.gouv.fr>, under in your project page, once your project’s submission for accreditation is validated. Please refer to the so-called section [Recetter](https://partenaires.franceconnect.gouv.fr/monprojet/recetter/).

To solve this issue, Azure AD B2C offers the possibility to display a custom HTML page content, that is merged with an HTML fragment inserted by Azure AD B2C to reflect the considered user flow or custom policy configuration, and then the complete page is displayed to your customer.

This custom HTML page content can be hosted on any publicly available HTTPS endpoint that supports [Cross-Origin Resource Sharing (CORS)](https://www.w3.org/TR/cors/). An [Azure Blob storage](https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction) can be leveraged to host the custom HTML page and images such as identity provider logos and FranceConnect buttons.

 

The .svg and .png files to use for the button can be download here: <https://partenaires.franceconnect.gouv.fr/files/fc_boutons.zip>.

At runtime, the custom HTML page content is loaded from this HTTPS endpoint that you have to specify in your user flow or custom policy. This custom HTML page content can contain any HTML elements, including CSS and JavaScript, but can't include insecure elements like iframes. The only required element is a div element with id set to api, such as this one <div id="api"></div> within the HTML page.

For security reasons, Azure AD B2C does NOT offer the possibility of modifying the HTML code it injects in the above div element using JavaScript. It is therefore not possible to move HTML elements once the DOM is loaded in the same way as above-specified for Microsoft Power Pages.

<button class="fc-button fc-button-background" id="FranceConnectButton"></button>

...

<script>

$(document).ready(function () {

$("#FranceConnectButton").click(function () {

$("#FranceConnect").click();

});

});

</script>

One possible approach to solve this issue aims at creating a button that fulfill the requirements of the above-mentioned FranceConnect specifications and, which, via a JavaScript script, will trigger the button generated by Azure AD B2C in the inserted HTML fragment when it is selected.

### User flows in Azure AD B2C

The integration of the custom HTML page content into a specific user flow is configured directly on the Azure portal. It simply requires activating the use of JavaScript and specify the endpoint URL to the custom HTML page content.

### Custom policies in Azure AD B2C

The integration of the custom HTML page content with the custom policies works in the same way but the configuration must be defined directly in the XML-formatted custom policy files via a [ContentDefinition](https://learn.microsoft.com/en-us/azure/active-directory-b2c/contentdefinitions) XML element.

<ContentDefinition Id="api.signuporsignin">

<!-- Action required -->

<LoadUri>https://yourstorageaccount.blob.core.windows.net/customsignin/unified.html</LoadUri>

<RecoveryUri>~/common/default\_page\_error.html</RecoveryUri>

<DataUri>urn:com:microsoft:aad:b2c:elements:contract:unifiedssp:2.1.5</DataUri>

<Metadata>

<Item Key="DisplayName">Signin and Signup</Item>

</Metadata>

</ContentDefinition>

The URL pointing to the custom HTML page content must be specified between the LoadUri tags:

<LoadUri>

https://yourstorageaccount.blob.core.windows.net/customsignin/unified.html

</LoadUri>

To enable JavaScript client-side code, the elements on which the JavaScript is based MUST be immutable. If they're not immutable, any changes could cause unexpected behavior on your user pages.

To prevent these issues, one will need to enforce the use of a page layout and specify a page layout version to ensure the content definitions you’ve based your JavaScript on are immutable.

To specify a page layout version for the custom policy pages:

1. Select a page layout for the user interface elements of your application.
2. Define a page layout version with page contract version for all of the content definitions in your custom policy. The format of the value must contain the word contract: urn:com:microsoft:aad:b2c:elements:contract:page-name:version.

<RelyingParty>

<DefaultUserJourney ReferenceId="SignUpOrSignInWithFranceConnect" />

<UserJourneyBehaviors>

<ScriptExecution>Allow</ScriptExecution>

</UserJourneyBehaviors>

...

</RelyingParty>

The next step required is to allow the execution of scripts in the user journey by adding <ScriptExecution>Allow</ScriptExecution> in the [UserJourneyBehaviors](https://learn.microsoft.com/en-us/azure/active-directory-b2c/relyingparty#userjourneybehaviors) XML element.

For additional information on JavaScript customization, please refer to [Guidelines for using JavaScript](https://learn.microsoft.com/en-us/azure/active-directory-b2c/javascript-and-page-layout?pivots=b2c-custom-policy#guidelines-for-using-javascript).

# Additional references

*Use this section to provide links to any supporting documentation or related content. You may also embed attachments here.*

[FranceConnect Service Provider Documentation](https://partenaires.franceconnect.gouv.fr/fcp/fournisseur-service)

[OpenID Connect Core 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-core-1_0.html)

[Final: OpenID Connect Discovery 1.0 incorporating errata set 1](https://openid.net/specs/openid-connect-discovery-1_0.html)

[Final: OpenID Connect Session Management 1.0](https://openid.net/specs/openid-connect-session-1_0.html)

[Final: OAuth 2.0 Multiple Response Type Encoding Practices](https://openid.net/specs/oauth-v2-multiple-response-types-1_0.html)

[RFC4648 The Base16, Base32, and Base64 Data Encodings](https://www.rfc-editor.org/rfc/rfc4648)

[RFC5280 Internet X.509 Public Key Infrastructure Certificate Internet X.509 Public Key Infrastructure Certificate](https://www.rfc-editor.org/rfc/rfc5280)

[RFC 5322 Internet Message Format](https://www.rfc-editor.org/rfc/rfc5322)

[RFC646 Tags for Identifying Languages](https://www.rfc-editor.org/rfc/rfc5646)

[RFC6749 The OAuth 2.0 Authorization Framework](https://www.rfc-editor.org/rfc/rfc6749.html)

[RFC7515 JSON Web Signature (JWS)](https://www.rfc-editor.org/rfc/rfc7515)

[RFC7517 JSON Web Key (JWK)](https://www.rfc-editor.org/rfc/rfc7517)

[RFC7518 JSON Web Algorithms (JWA)](https://www.rfc-editor.org/info/rfc7518)

[RFC7519 JSON Web Token (JWT)](https://www.rfc-editor.org/rfc/rfc7519)

Zzz

1. After the launch of Power Apps portals on October 1, 2019, the full capabilities of Dynamics 365 Portals, previously offered only as an add-on to customer engagement apps (Dynamics 365 Sales, Dynamics 365 Customer Service, Dynamics 365 Field Service, Dynamics 365 Marketing, and Dynamics 365 Project Service Automation), are now available standalone in Power Apps. In other word, all Dynamics 365 Portals are now referred to as Power Apps portals. [↑](#footnote-ref-2)
2. Effective October 12, 2022, Power Apps portals are Power Pages. See [Microsoft Power Pages is now generally available](https://powerpages.microsoft.com/en-us/blog/microsoft-power-pages-is-now-generally-available/). [↑](#footnote-ref-3)
3. As far as the client application is concerned, and from the security point of view – one SHOULD NOT use the x5c certificate to validate the signature directly. In that case, anybody could just provide their own certificate and spoof any identity. The purpose of the x5t header is to identify the signer – One should check the trust in the certificate provided by x5c or x5t (or its issuer) under the specified iss value of the id\_token, only then one should validate the signature. See section id\_token token below. [↑](#footnote-ref-4)
4. After the launch of Power Apps portals on October 1, 2019, the full capabilities of Dynamics 365 Portals, previously offered only as an add-on to customer engagement apps (Dynamics 365 Sales, Dynamics 365 Customer Service, Dynamics 365 Field Service, Dynamics 365 Marketing, and Dynamics 365 Project Service Automation), are now available standalone in Power Apps. In other word, all Dynamics 365 Portals are now referred to as Power Apps portals. [↑](#footnote-ref-5)
5. Effective October 12, 2022, Power Apps portals are Power Pages. See [Microsoft Power Pages is now generally available](https://powerpages.microsoft.com/en-us/blog/microsoft-power-pages-is-now-generally-available/). [↑](#footnote-ref-6)