REFERENCE IMPLEMENTATION USER MANUAL

Index

List of Figures

1 INTRODUCTION

The purpose of this document is to describe the functionalities that a USER of the reference implementation of the Enterprise Wallet associated with deliverable 3.6 can perform from the administration interface.

This manual includes, on the one hand, a description of how to use the functionalities related to the management of verifiable credentials currently provided by the platform and the management of users.

1.1 Access to the administration interface

The backend of the reference implementation exposes an administration interface that is accessible from any browser. To access it, once an instance of the Wallet has been launched, simply go to the URL where it is hosted. If everything works correctly, a screen similar to the one below should appear, prompting you to enter an username and password. The first time you access the system, you can use the username and password: "Identfy". This will allow you to log in as a predefined user in the system with superuser permissions.

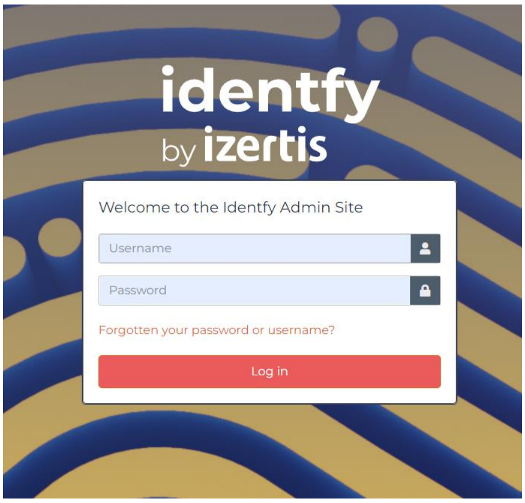

identfy
by izertis
Welcome to the Identfy Admin Site
Username
Password
Forgotten your password or username?
Log in

Figure 1:Django Admin Login

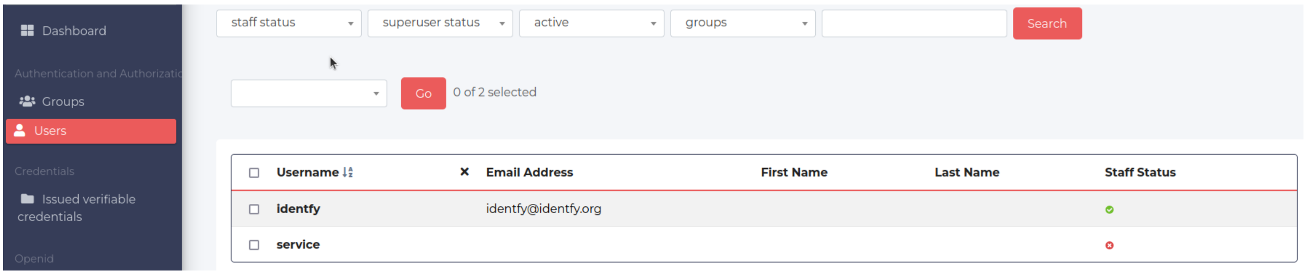
2 PLATFORM USER MANAGEMENT

The following sections explain how to view user information and manage the different models related to platform management.

2.1 Users

Users is a native Django model that provides the data for logging into the Administration Platform. In this model, user permissions are configured, as well as personal information such as name, surname, email, and user type.

To view user information, go to the "Users" section in the side menu.



□	Username	Email Address	First Name	Last Name	Staff Status
□	**identfy**	identfy@identfy.org			✓
□	**service**				✗

Figure 2: Users list

As you can see, in addition to the user used for logging in, there is also a "service" user. This user has a set of special permissions that only allow it to read certain specific data models related to credential issuance and verification. It is a user designed to be used by the "Entity Service" component. If desired, you can change both the name and password of the default users, or even delete them if you already have others.

If you interact with any of the user entries, you will see a menu like the one below with all the configurable categories.

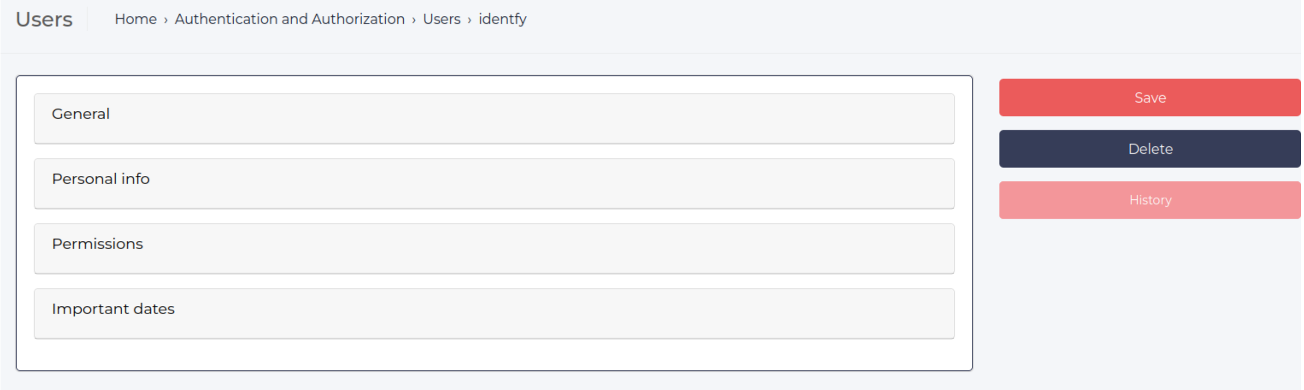
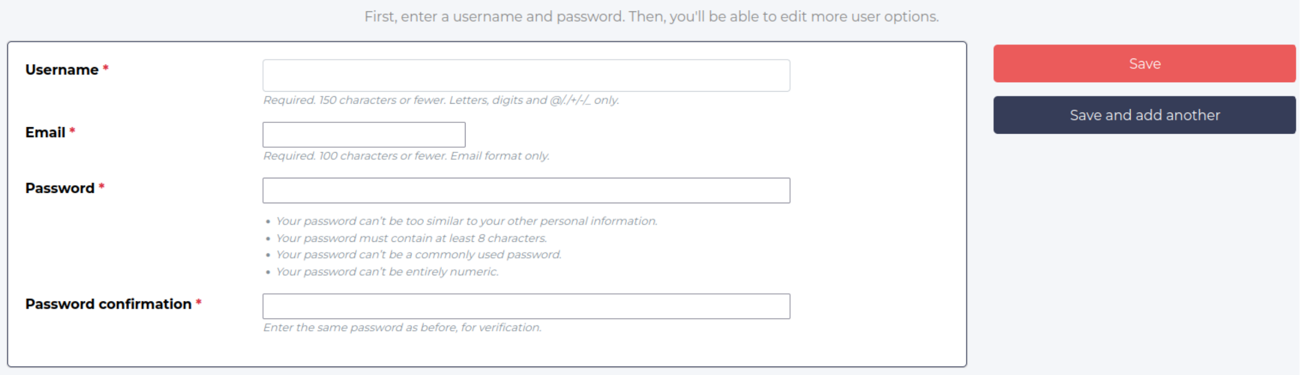


Figure 3: Users information

If, on the other hand, you want to create a new user, simply click the "add" button on the right-hand side. A form like the one below will appear on the screen:



First, enter a username and password. Then, you'll be able to edit more user options.

Username *

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email *

Required. 100 characters or fewer. Email format only.

Password *

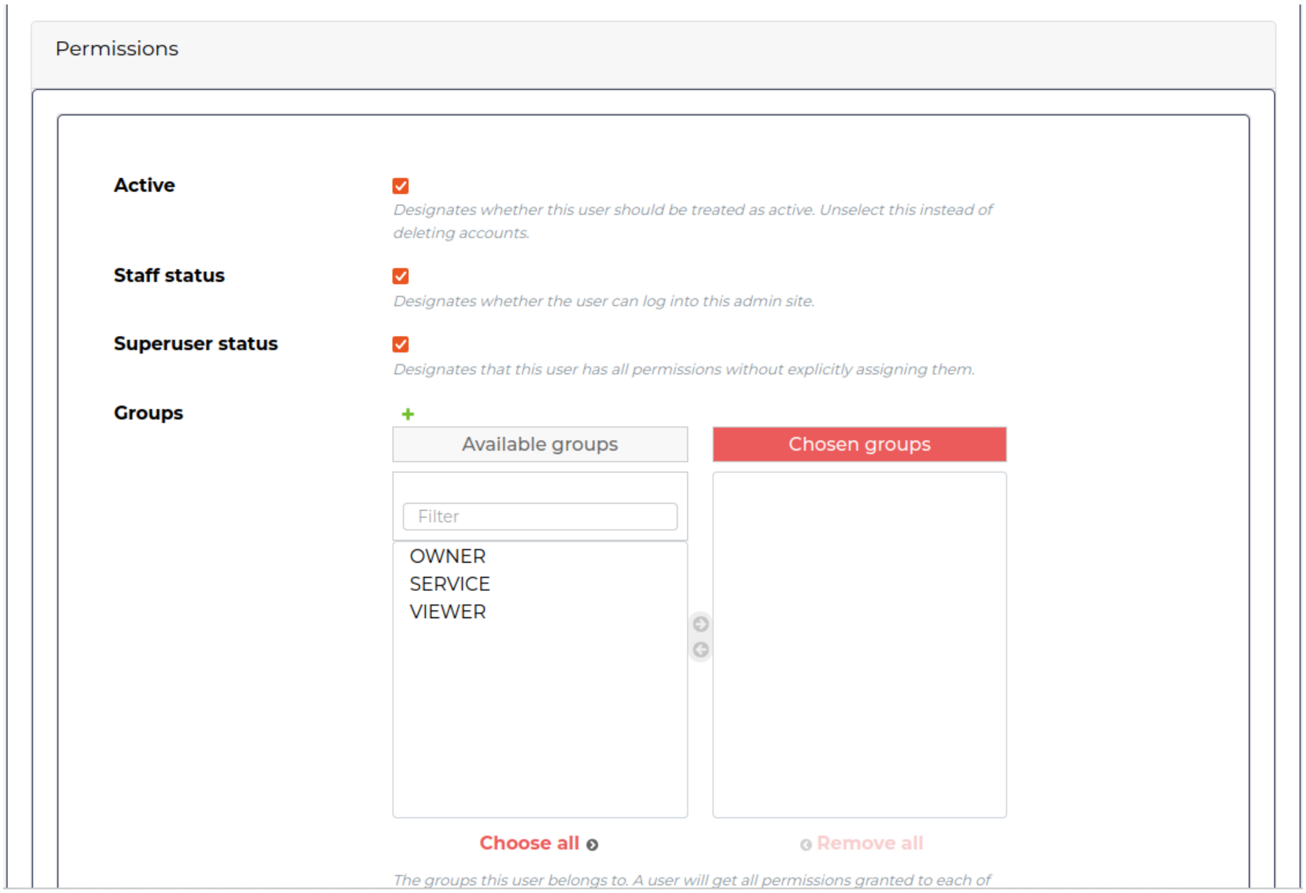
- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation *

Enter the same password as before, for verification.

Figure 4:User Add form

All fields must be filled in to create a user. If the Wallet is properly configured with an email service, the user in question should receive an email to change their password. Once a user is created, you must access their entry to set their permissions.



Permissions

Active ☑

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

Staff status ☑

Designates whether the user can log into this admin site.

Superuser status ☑

Designates that this user has all permissions without explicitly assigning them.

Groups

The groups this user belongs to. A user will get all permissions granted to each of

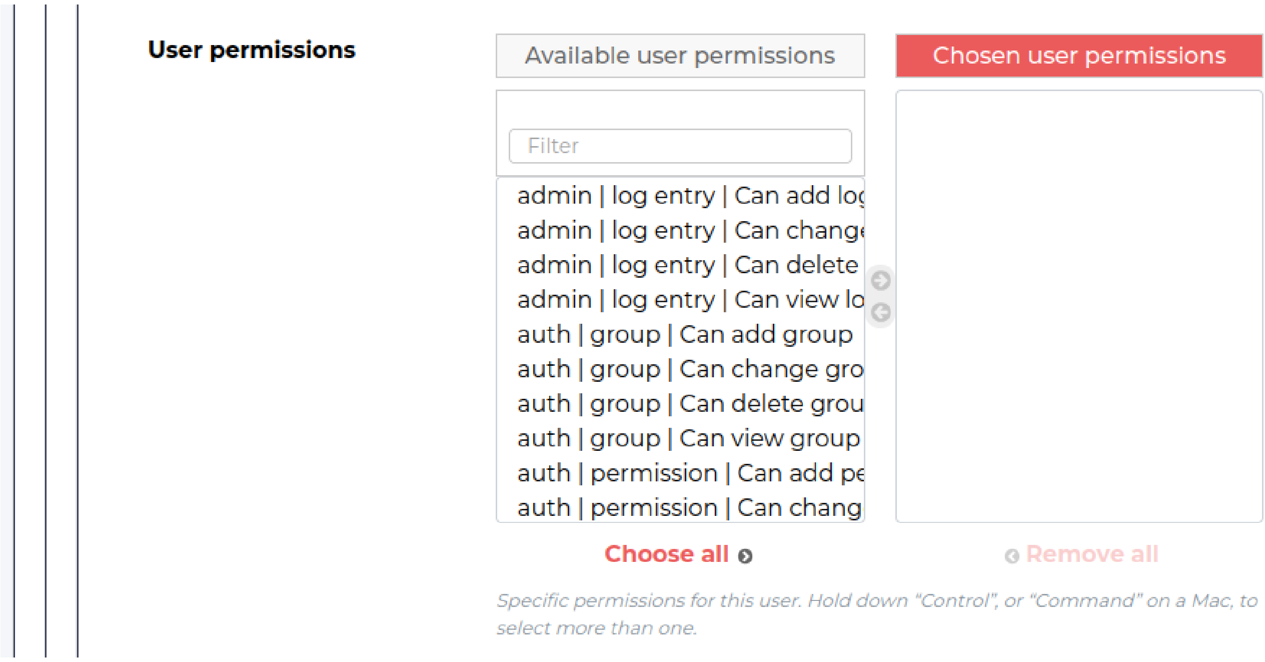
Figure 5: User Permissions

In the permissions section, you can indicate whether the user account is active or not, as well as their superuser status. The staff status is required to access the administration interface; otherwise, the user will only be able to interact with the APIs that require authentication. The "groups" section allows you to specify which group the user belongs to. Doing so will grant the user all the permissions associated with that group. By default, there are three permission groups:

- OWNER: This permission group allows the user to perform both read and write operations on the various data models of the Wallet.
- SERVICE: Only allows reading specific data models. This corresponds to the group intended for users employed by the "Entity Service" component.
- VIEWER: This permission group grants the user read permissions, but no write permissions.

The specific permissions of each group can be checked in the "Groups" section of the left-hand side menu. It is not necessary to assign a group to users if not desired.

After assigning the groups, we can indicate additional permissions for the user. These permissions are additive to the previous ones.



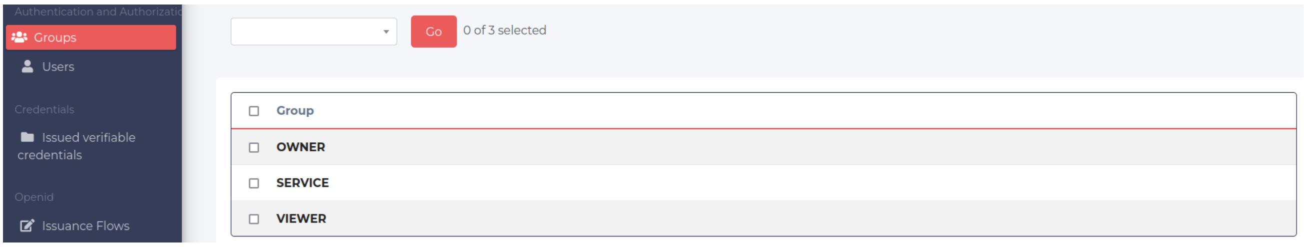
admin | log entry | Can add log
admin | log entry | Can change
admin | log entry | Can delete
admin | log entry | Can view lo
auth | group | Can add group
auth | group | Can change gro
auth | group | Can delete grou
auth | group | Can view group
auth | permission | Can add pe
auth | permission | Can chang

Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

Figure 6: User Permissions - 2

2.2 Groups

The Wallet allows defining additional permission groups or modifying existing ones. To do this, you need to access the "Groups" option from the side menu.

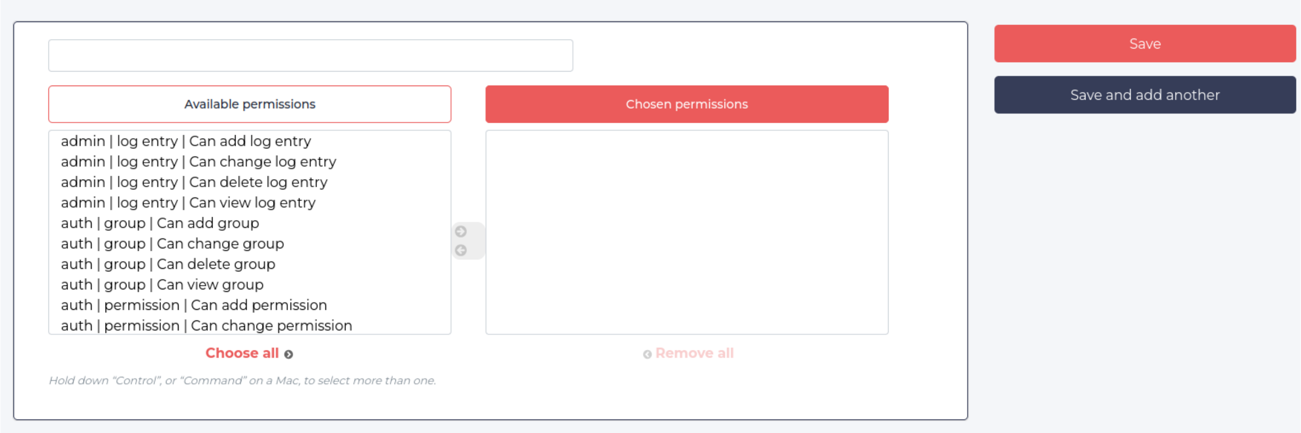


Authentication and Authorizatio
Groups
Users
Credentials
Issued verifiable credentials
Openid
Issuance Flows
Go
0 of 3 selected

□	Group
□	OWNER
□	SERVICE
□	VIEWER

Figure 7: Group list

To add a new group, just like with users, click the “Add” button on the right-hand side.



admin | log entry | Can add log entry
admin | log entry | Can change log entry
admin | log entry | Can delete log entry
admin | log entry | Can view log entry
auth | group | Can add group
auth | group | Can change group
auth | group | Can delete group
auth | group | Can view group
auth | permission | Can add permission
auth | permission | Can change permission

Hold down “Control”, or “Command” on a Mac, to select more than one.

Figure 8: Group Add Form

You will need to provide the group's name in the top section and assign the associated permissions.

3 KEY ASSIGNMENT

Within the "Organizations" section, there is an entry named "Organization Keys" that allows users to specify the keys they wish to use for their Enterprise Wallet. The keys must be provided in JWK format and must use either the "secp256k1" or "secp256r1" algorithm. Keys of the first type are used for operations involving the creation of transactions that modify the EBSI ledger. In contrast, the latter are used to sign Verifiable Credentials, as well as any other data involved in protocol-related processes.

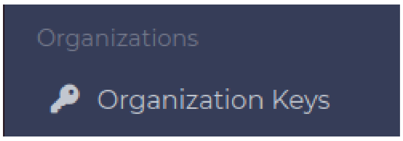
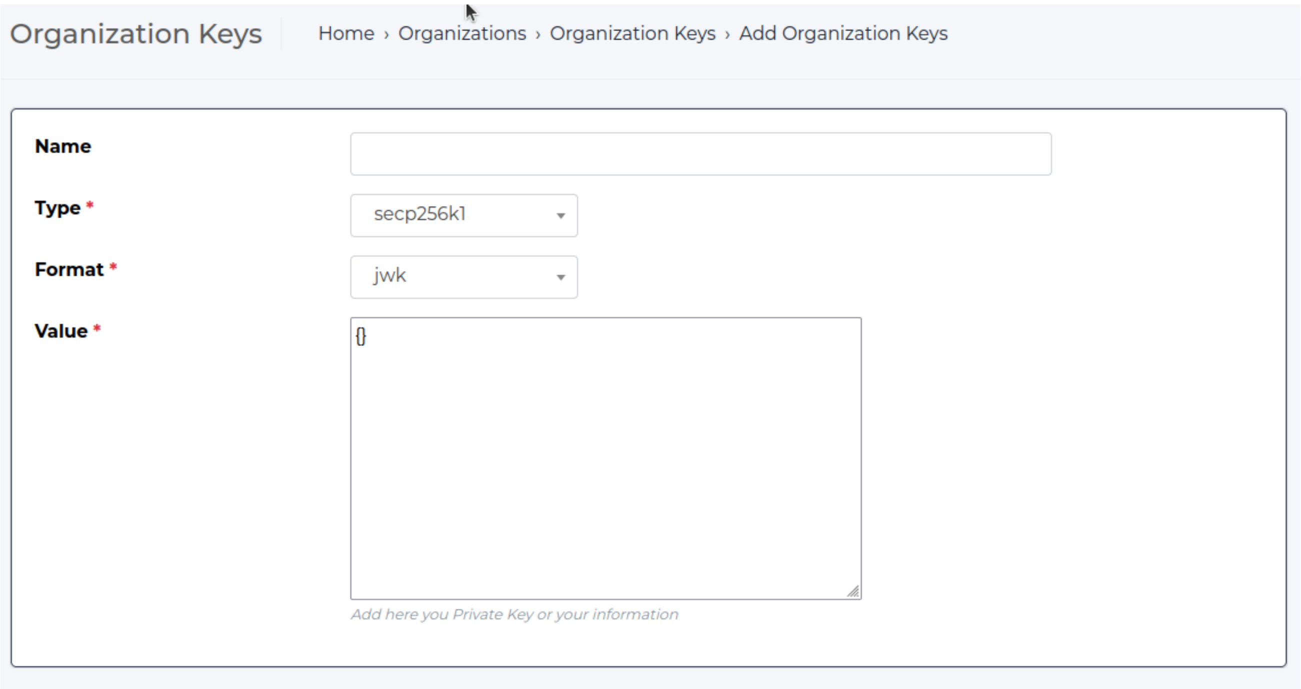


Figure 9: Organizations Section

If there is no need to modify the ledger, it is sufficient to define only "secp256r1" keys.


Organization Keys
Home › Organizations › Organization Keys › Add Organization Keys

Name	
Type *	secp256k1
Format *	jwk
Value *	{} Add here you Private Key or your information

Figure 10: Organization Keys Screen

4 ISSUANCE AND VERIFICATION OF VERIFIABLE CREDENTIALS

When we access the administration interface with a user, the available menu options will depend on the associated permissions. For simplicity, this manual assumes that the user has permissions to perform all the operations that will be described from now on.

4.1 OpenID

All menu options related to the issuance or verification of verifiable credentials have been grouped under the same category named “OpenID”.

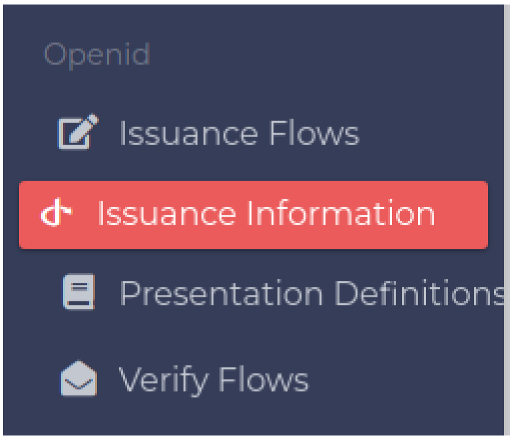


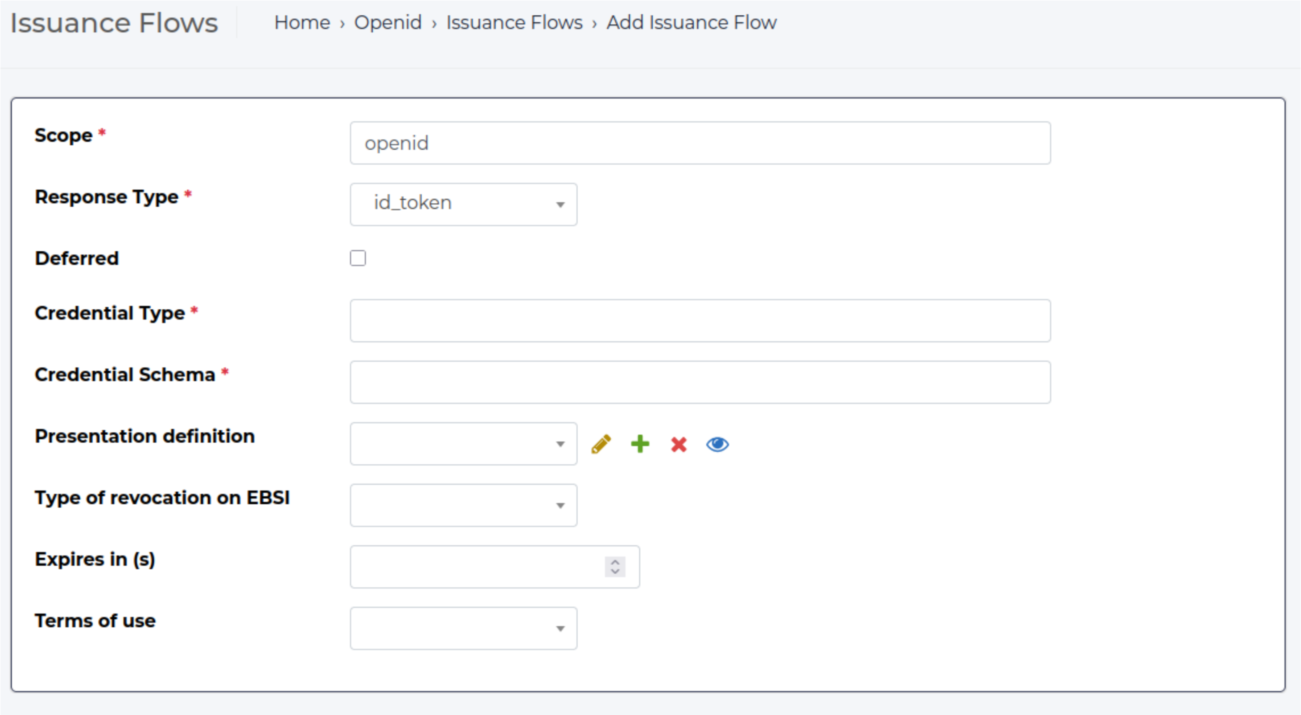
Figure 11: OpenID configurations

4.1.1 Defining an Issuance Flow

To define an issuance flow, we must access the option with the same name from the menu. Once inside, we add a new flow with the "add" button. The user will be asked to fill out a form with the following attributes:

- ***Scope***: Defines the identifier of the operation, which the Holder Wallet must use to indicate its interest in this process. By default, "openid".
- ***Response Type***: Specifies what type of token the user must provide to authenticate.
- ***Deferred***: A flag that determines whether the credential will use the deferred flow.
- ***Credential Type***: The name of the credential type, e.g., "MyAcademicId".
- ***Credential Schema***: The URL to the schema specification.
- ***Presentation Definition***: If a VP Token is requested, this section allows the specification of the presentation definition to be used.

- ***Type of Revocation on EBSI:*** Allow to select the type of revocation to be used. Currently, only StatusList is available.
- ***Expires In***: This parameter allows us to specify when we want the issued credentials to expire.
- ***Terms of use***: This parameter allow to link the entity accreditation to the credential to issue.



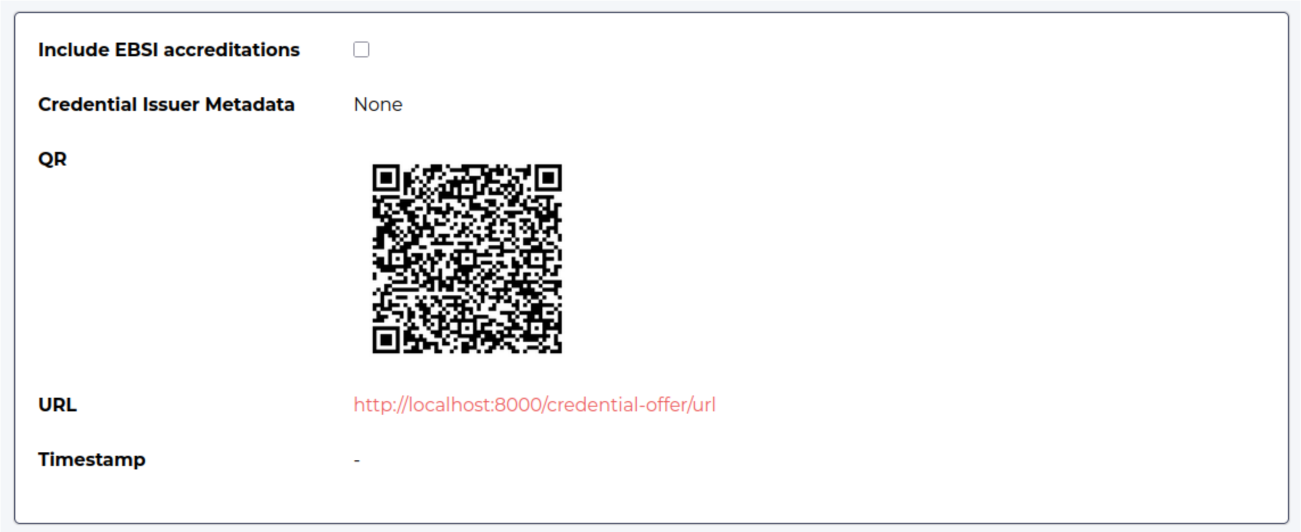
Issuance Flows — Home › Openid › Issuance Flows › Add Issuance Flow

Scope *	openid
Response Type *	id_token
Deferred	□
Credential Type *	
Credential Schema *	
Presentation definition	
Type of revocation on EBSI	
Expires in (s)	
Terms of use	

Figure 12: Issuance flow

4.1.2 Defining the Issuance Information

Once at least one issuance flow has been defined, the next step is to generate an "Issuance Information" entry. To do this, we must access the option with the same name in the menu and click "Add".

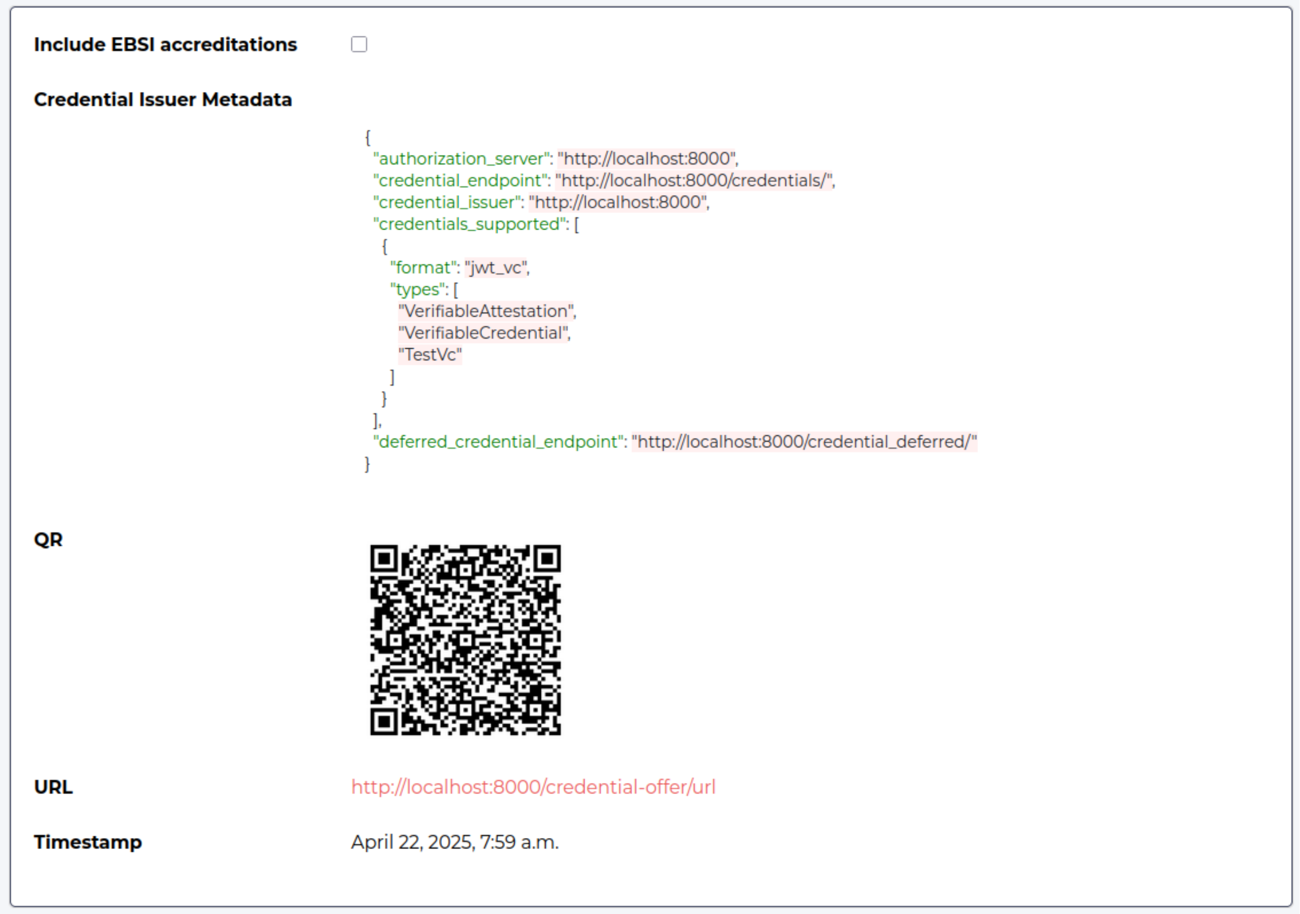


Include EBSI accreditations	□
Credential Issuer Metadata	None
QR	
URL	http://localhost:8000/credential-offer/url
Timestamp	-

Figure 13: Issuance information

As you can see, the form does not allow for modification. Simply click "save," and the system will automatically generate the relevant information. There is a checkbox at the top of the screen that allows the user to specify if they want to include to the issuance information the different accreditations this entity can issue. Accreditations are a special type of verifiable credentials linked to the EBSI governance model.

If you interact again with the newly created entry:



Include EBSI accreditations □

Credential Issuer Metadata

```
{
 "authorization_server": "http://localhost:8000",
 "credential_endpoint": "http://localhost:8000/credentials/",
 "credential_issuer": "http://localhost:8000",
 "credentials_supported": [
  {
   "format": "jwt_vc",
   "types": [
    "VerifiableAttestation",
    "VerifiableCredential",
    "TestVc"
   ]
  }
 ],
 "deferred_credential_endpoint": "http://localhost:8000/credential_deferred/"
}
```

QR

URL http://localhost:8000/credential-offer/url

Timestamp April 22, 2025, 7:59 a.m.

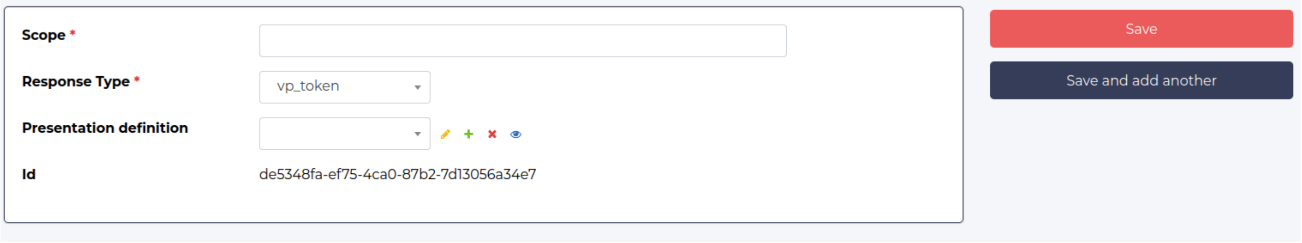
Figure 14: Issuance Information Entry

You will be able to view the information automatically generated by the Wallet. You do not need to create more instances of this model. Each time you update an issuance flow or add a new one, the system will automatically update the "Issuance Information" instance.

4.1.3 Defining a Verify Flow

To define a verification flow, go to the "Verify Flow" section of the menu. In this case, the form to be filled out is as follows:

- ***Scope***: Defines the scope of the operation, which also acts as the operation's identifier.
- ***Response Type***: Specifies what type of token the user must provide to authenticate. Generally, VP Token will be used, but the option of ID Token is allowed to complete EBSI conformance tests.
- ***Presentation Definition***: This section allows for the specification of the presentation definition to be used. Only necessary if a VP Token is requested.
- ***ID***: This is the unique identifier of the verification process, which is required to obtain the VP offer. It is automatically assigned.



Scope *

Response Type * vp_token

Presentation definition

Id de5348fa-ef75-4ca0-87b2-7d13056a34e7

Figure 15: Verify flow

4.1.4 Defining the Presentation Definition

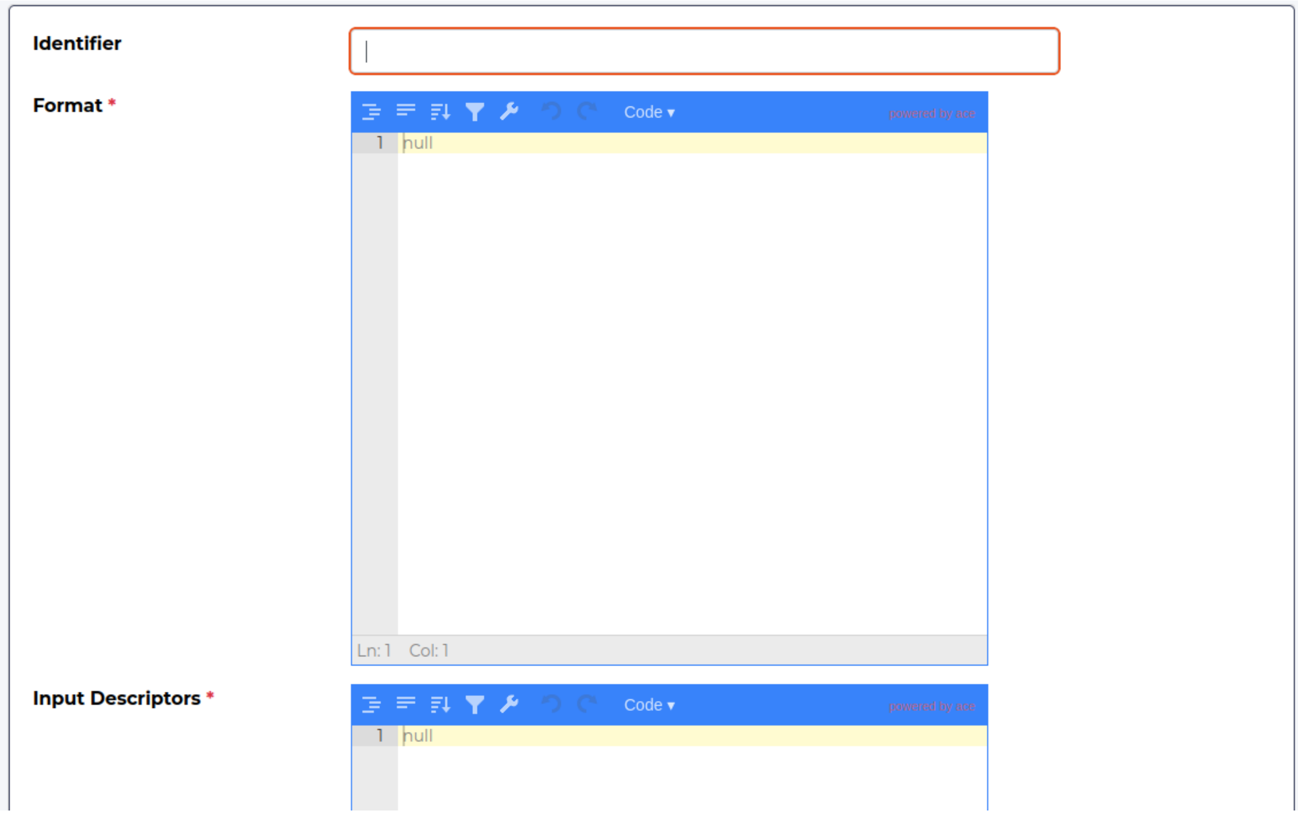
To use VP Tokens, whether in an issuance or verification process, it is necessary to define at least one Presentation Definition. To do this, you must access the menu option with the same name. Below is a description of what each of the fields shown when adding a new entry to the database corresponds to:

- ***Identifier***: Defines the unique identifier of the definition. The user can specify it, but if not defined, the Wallet will generate a default UUID.
- ***Format***: This is a JSON object where the format in which the credential must be delivered for validation is defined. For example:

```
{
  "jwt_vc": {
    "alg": [
      "ES256"
    ]
  },
  "jwt_vp": {
    "alg": [
      "ES256"
    ]
  }
}
```

- ***Input Descriptors***: This is a JSON object where the credential data to be presented is defined, as well as any restrictive data, meaning data that must be of a certain type. For example:

```
[
  {
    "id": "<any id, random or static>",
    "format": {
      "jwt_vc": {
        "alg": [
          "ES256"
        ]
      }
    },
    "constraints": {
      "fields": [
        {
          "path": [
            "$.vc.type"
          ],
          "filter": {
            "type": "array",
            "contains": {
              "const": "VerifiableAttestation"
            }
          }
        }
      ]
    }
  }
]
```



Identifier

Format *

1 null

Ln: 1 Col: 1

Input Descriptors *

1 null

Figure 16: Presentation Definition

5 MANAGEMENT OF VERIFIABLE CREDENTIALS

The Enterprise Wallet allows users to store Verifiable Credentials and review a history of previously issued credentials, including the option to request their revocation through the interface. All features related to the management of Verifiable Credentials can be found in the "Verifiable Credentials" section.

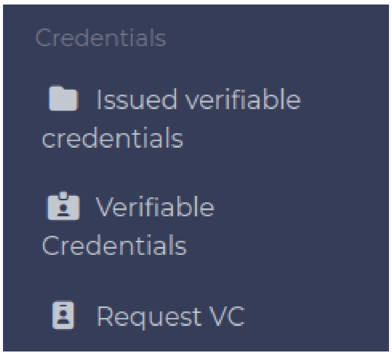
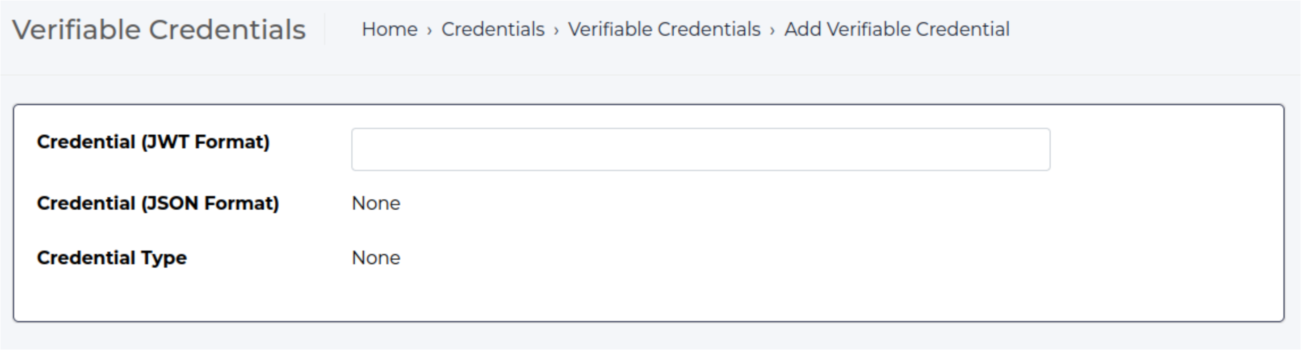


Figure 17: Credentials Section

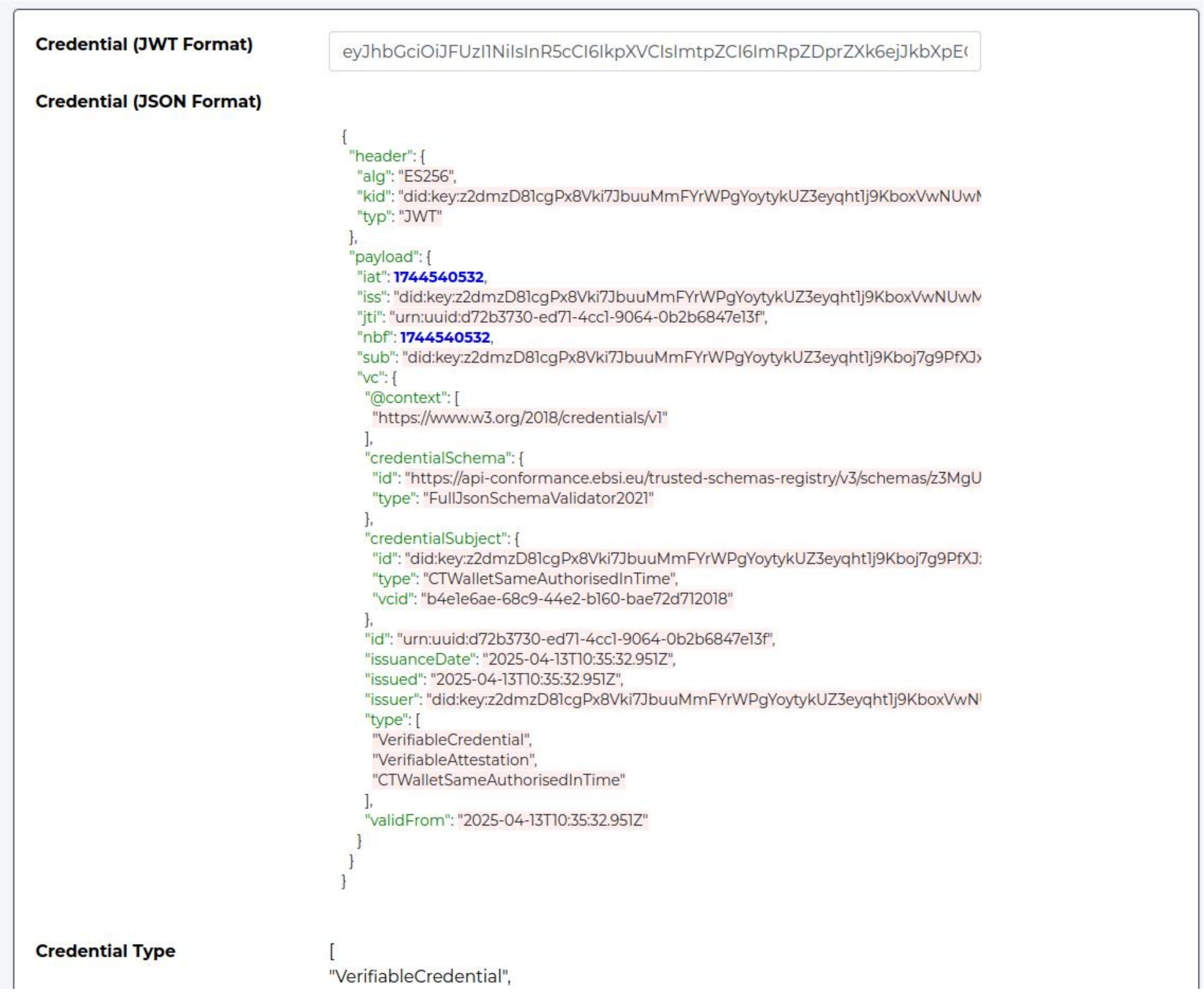
5.1 Verifiable Credentials

This subsection allows the Enterprise Wallet user to store Verifiable Credentials. These credentials may have been obtained through other means, and the user can manually upload them using the interface. In all cases, the credential must be in JWT JSON format. Once saved, an entry will be created in the collection, through which the credential can be viewed in its decoded form—allowing users to inspect its full content. Additionally, the Enterprise Wallet can also request credentials and automatically store them in this collection once successfully obtained.


Verifiable Credentials
Home › Credentials › Verifiable Credentials › Add Verifiable Credential

Credential (JWT Format)	
Credential (JSON Format)	None
Credential Type	None

Figure 18: Verifiable Credentials Storage



Credential (JWT Format) eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImRpZDprZXk6ejJkbXpE

Credential (JSON Format)

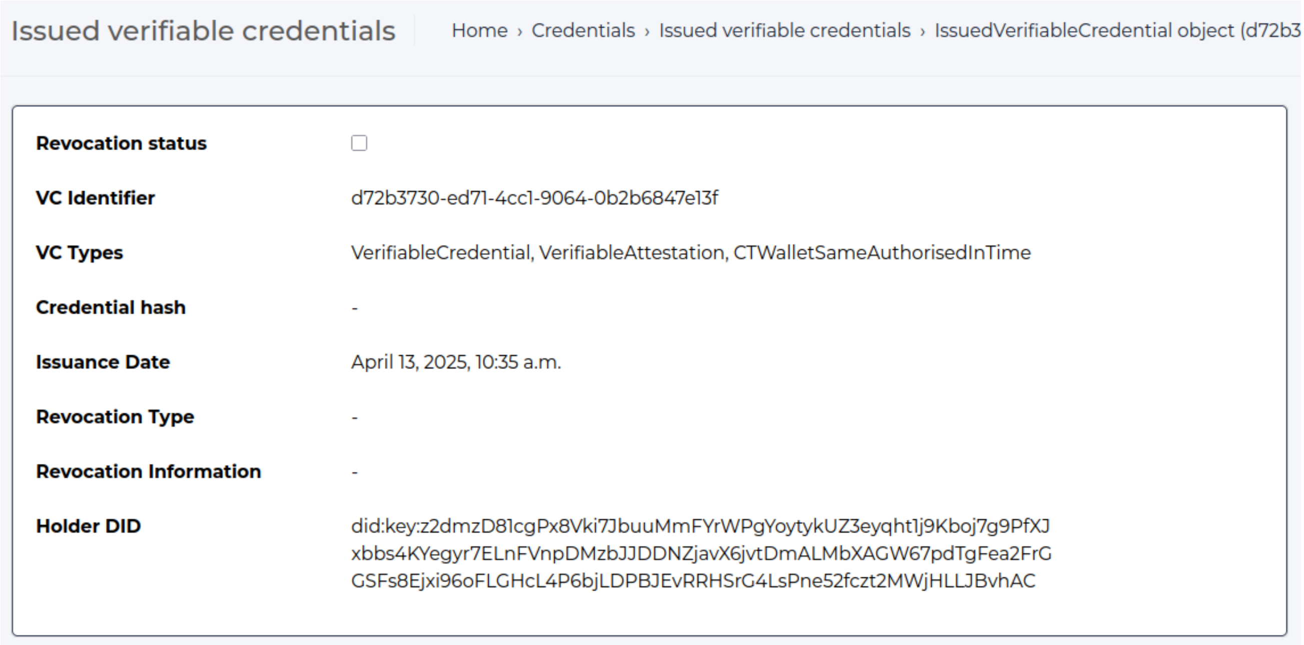
```
{
 "header": {
  "alg": "ES256",
  "kid": "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KboxVwNUw
  "typ": "JWT"
 },
 "payload": {
  "iat": 1744540532,
  "iss": "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KboxVwNUwM
  "jti": "urn:uuid:d72b3730-ed71-4cc1-9064-0b2b6847e13f",
  "nbf": 1744540532,
  "sub": "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9Kboj7g9PfXJ
  "vc": {
   "@context": [
    "https://www.w3.org/2018/credentials/v1"
   ],
   "credentialSchema": {
    "id": "https://api-conformance.ebsi.eu/trusted-schemas-registry/v3/schemas/z3MgU
    "type": "FullJsonSchemaValidator2021"
   },
   "credentialSubject": {
    "id": "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9Kboj7g9PfXJ
    "type": "CTWalletSameAuthorisedInTime",
    "vcid": "b4e1e6ae-68c9-44e2-b160-bae72d712018"
   },
   "id": "urn:uuid:d72b3730-ed71-4cc1-9064-0b2b6847e13f",
   "issuanceDate": "2025-04-13T10:35:32.951Z",
   "issued": "2025-04-13T10:35:32.951Z",
   "issuer": "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KboxVwN
   "type": [
    "VerifiableCredential",
    "VerifiableAttestation",
    "CTWalletSameAuthorisedInTime"
   ],
   "validFrom": "2025-04-13T10:35:32.951Z"
  }
 }
}
```

Credential Type [
"VerifiableCredential",

Figure 19: Example of Verifiable Credential

5.2 Issued Verifiable Credentials

Each issued Verifiable Credential is stored as an entry in this model. Specifically, the stored information includes the credential type, its issuer identifier, its unique identifier, and revocation details—if available. If revocation was not enabled during the issuance flow, such information may be absent. When revocation is supported, the user may request it by ticking the appropriate checkbox and saving the entry. Once revoked, the action is irreversible.


Issued verifiable credentials

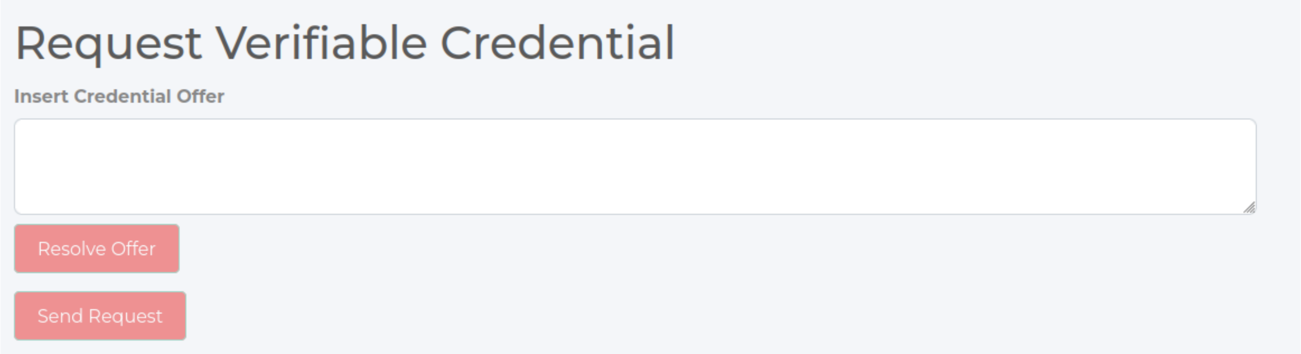
Home › Credentials › Issued verifiable credentials › IssuedVerifiableCredential object (d72b3

Revocation status	□
VC Identifier	d72b3730-ed71-4cc1-9064-0b2b6847e13f
VC Types	VerifiableCredential, VerifiableAttestation, CTWalletSameAuthorisedInTime
Credential hash	-
Issuance Date	April 13, 2025, 10:35 a.m.
Revocation Type	-
Revocation Information	-
Holder DID	did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9Kboj7g9PfXJxbbs4KYegyr7ELnFVnpDMzbJJDDNZjavX6jvtDmALMbXAGW67pdTgFea2FrGGSFs8Ejxi96oFLGHcL4P6bjLDPBJEvRRHSrG4LsPne52fczt2MWjHLLJBvhAC

Figure 20: Example of issued verifiable credential

5.3 Request Verifiable Credentials

The Enterprise Wallet allows users to request Verifiable Credentials from other issuing entities via the "Request VC" section.


Request Verifiable Credential

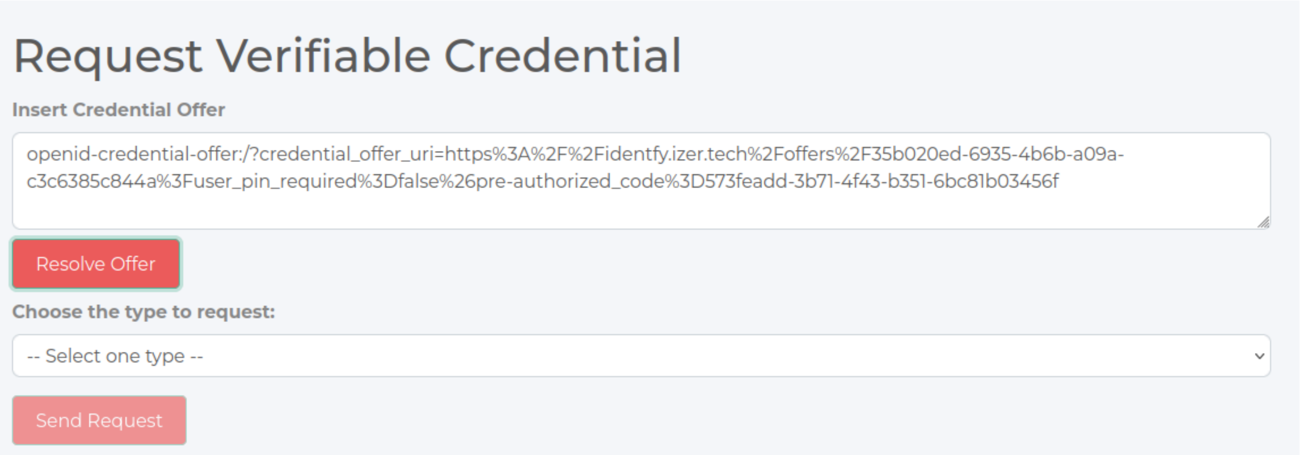
Insert Credential Offer

Figure 21: Request VC Section

This section presents a simple form where the user is prompted to enter a "credential-offer." The offer must follow a specific format:

```
openid-credential-offer:/?credential_offer_uri=https%3A%2F%2Fidentfy.izer.tech%2Foffers%2F35b020ed-6935-4b6b-a09a-c3c6385c844a%3Fuser_pin_required%3Dfalse%26pre-authorized_code%3D573feadd-3b71-4f43-b351-6bc81b03456f
```

After entering the offer, clicking the "Resolve Offer" button will reveal which credentials can be requested based on the offer.



Request Verifiable Credential

Insert Credential Offer

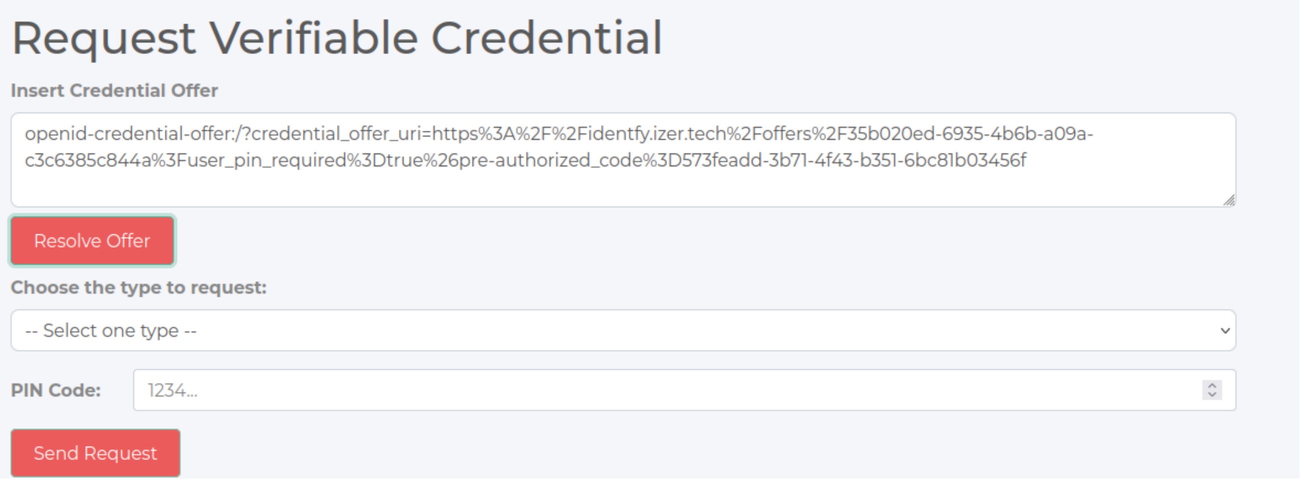
openid-credential-offer:/?credential_offer_uri=https%3A%2F%2Fidentfy.izer.tech%2Foffers%2F35b020ed-6935-4b6b-a09a-c3c6385c844a%3Fuser_pin_required%3Dfalse%26pre-authorized_code%3D573feadd-3b71-4f43-b351-6bc81b03456f

Choose the type to request:

-- Select one type --

Figure 22: Request VC Section - Example

This action expands the form to include a dropdown for selecting a credential type from the available options. If the offer supports the pre-authorized flow and requires a PIN code, the form will expand further to include a field for entering the PIN.



Request Verifiable Credential

Insert Credential Offer

openid-credential-offer:/?credential_offer_uri=https%3A%2F%2Fidentfy.izer.tech%2Foffers%2F35b020ed-6935-4b6b-a09a-c3c6385c844a%3Fuser_pin_required%3Dtrue%26pre-authorized_code%3D573feadd-3b71-4f43-b351-6bc81b03456f

Choose the type to request:

-- Select one type --

PIN Code: 1234...

Figure 23: Request VC Section - Example with PIN code

Once the type is selected, the user can click “Send Request” to initiate the credential request. If successful, the credential will be stored automatically in the Wallet and can be viewed in the “Verifiable Credentials” section. If the requested VC uses the deferred flow, the user must manually store the exchange code and, once the credential becomes available, use the Wallet API to redeem it. The relevant endpoint is ***/credentials/request-deferred***. Future versions of the reference implementation may extend the current data model to support management of exchange codes for this flow.

5.4 Registration of accreditations

The Enterprise Wallet supports the registration of accreditations within the EBSI ecosystem. Accreditations are Verifiable Credentials that grant a legal entity a specific role in the ecosystem, such as Trusted Issuer or TAO. To register them, the user simply needs to obtain the corresponding credentials and store them in the Wallet, as described in section 5.1. The Wallet will automatically execute tasks that, once completed, will result in the accreditation being registered. This same logic applies to the credential required for DID registration.

If the credential grants permission to issue other credentials, the Wallet will automatically register a revocation proxy in EBSI. This proxy allows for future credential revocation using the EBSI system. Through this mechanism, verifiers can query the credential status directly against the EBSI API rather than the Enterprise Wallet, thereby limiting the Wallet's visibility into user activity.

Once at least one credential granting issuance capabilities (TI, TAO, or RTAO) has been registered, the user gains access to additional options in the "EBSI" section.

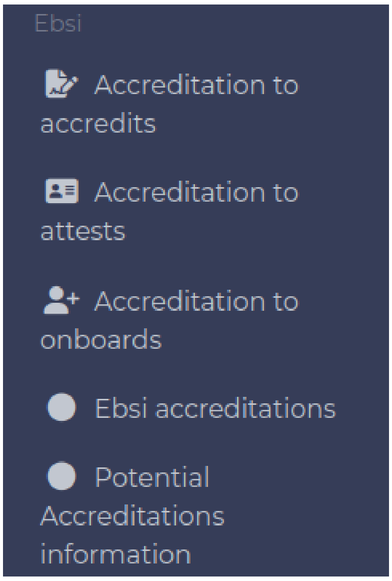
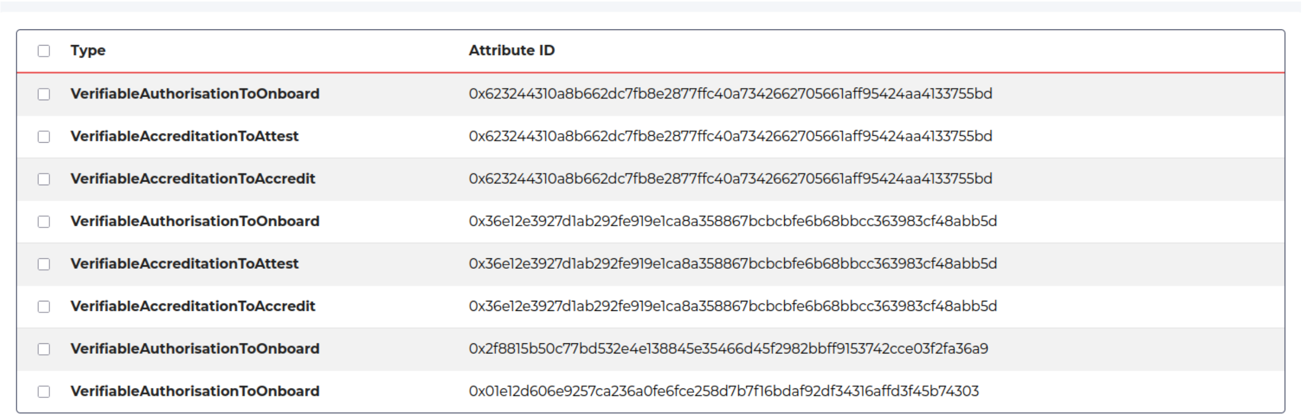

Ebsi
Accreditation to accredits
Accreditation to attests
Accreditation to onboards
Ebsi accreditations
Potential Accreditations information

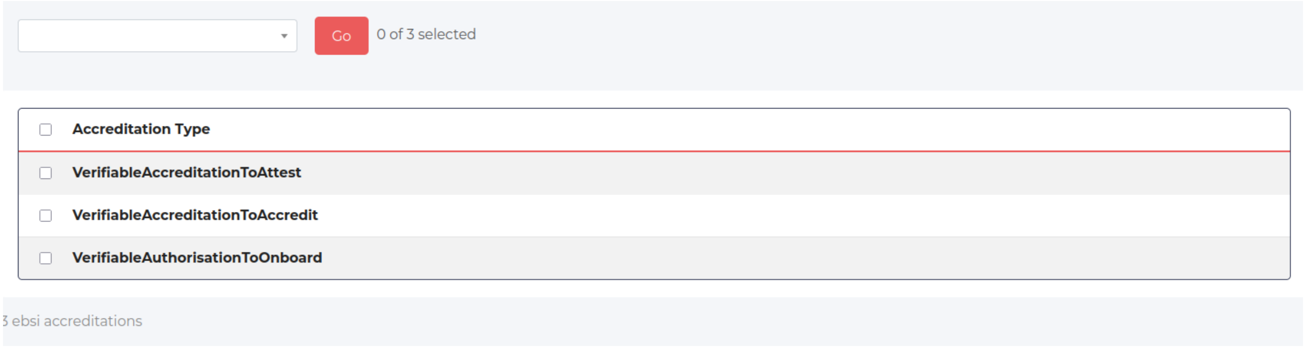
Figure 24: EBSI Section

Specifically, the Wallet will automatically create two data models: "Potential Accreditations Information" and "EBSI Accreditations." The former lists all accreditations that can be issued by the Enterprise Wallet due to its role in the network. Each entry specifies the attribute that enables issuance—essentially a synonym for the received accreditation that authorizes the action. The latter model summarizes the types of accreditations that can be issued, allowing the users to indicate certain conditions, like for example the type of authorization requested to the holder. These models are automatically created and should not be manually modified by the user. However, they can be deleted if necessary.



☐ Type	Attribute ID
☐ **VerifiableAuthorisationToOnboard**	0x623244310a8b662dc7fb8e2877ffc40a734266270566 1aff95424aa4133755bd
☐ **VerifiableAccreditationToAttest**	0x623244310a8b662dc7fb8e2877ffc40a7342662705661aff95424aa4133755bd
☐ **VerifiableAccreditationToAccredit**	0x623244310a8b662dc7fb8e2877ffc40a7342662705661aff95424aa4133755bd
☐ **VerifiableAuthorisationToOnboard**	0x36e12e3927d1ab292fe919e1ca8a358867bcbcbfe6b68bbcc363983cf48abb5d
☐ **VerifiableAccreditationToAttest**	0x36e12e3927d1ab292fe919e1ca8a358867bcbcbfe6b68bbcc363983cf48abb5d
☐ **VerifiableAccreditationToAccredit**	0x36e12e3927d1ab292fe919e1ca8a358867bcbcbfe6b68bbcc363983cf48abb5d
☐ **VerifiableAuthorisationToOnboard**	0x2f8815b50c77bd532e4e138845e35466d45f2982bbff9153742cce03f2fa36a9
☐ **VerifiableAuthorisationToOnboard**	0x01e12d606e9257ca236a0fe6fce258d7b7f16bdaf92df34316affd3f45b74303

Figure 25: Potential Accreditation Information



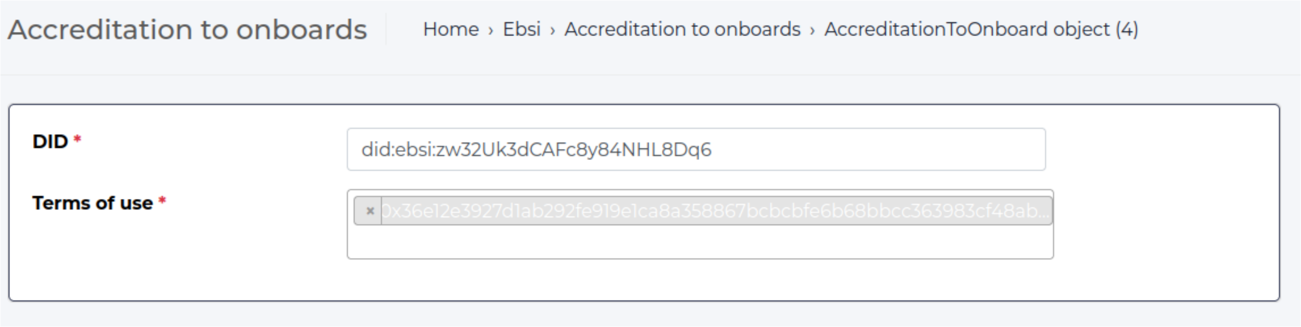
☐ Accreditation Type
☐ **VerifiableAccreditationToAttest**
☐ **VerifiableAccreditationToAccredit**
☐ **VerifiableAuthorisationToOnboard**

3 ebsi accreditations

Figure 26: EBSI Accreditations

Note that although "VerifiableAuthorisationToOnboard" is not technically an accreditation, it has been grouped under the same category for simplicity.

The remaining options in the "EBSI" section enable the issuance of accreditations to entities authorized by the Enterprise Wallet. In essence, this involves configuring whitelists that define which DIDs are authorized to request specific credentials and which accreditation held by the Wallet will be used to issue them, effectively linking both credentials to strengthen the trust chain. For "Accreditation to attest" and "Accreditation to accredit," this linkage is simplified in the user interface by allowing the user to select which credential types should be issuable using the selected accreditation—these options are extracted from prior configurations and serve the same function, but with a simplified user experience.

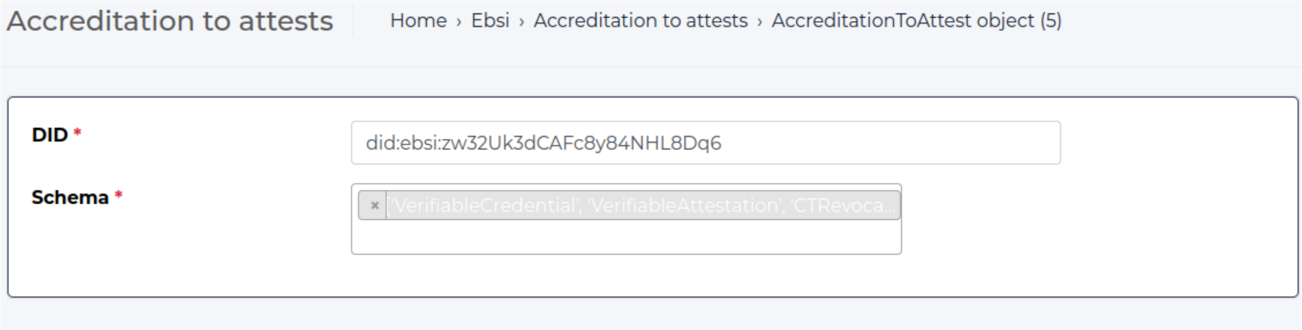


Accreditation to onboards

Home › Ebsi › Accreditation to onboards › AccreditationToOnboard object (4)

DID *	did:ebsi:zw32Uk3dCAFc8y84NHL8Dq6
Terms of use *	× 0x36e12e3927d1ab292fe919e1ca8a358867bcbcbfe6b68bbcc363983cf48ab...

Figure 27: Accreditation to onboard white list



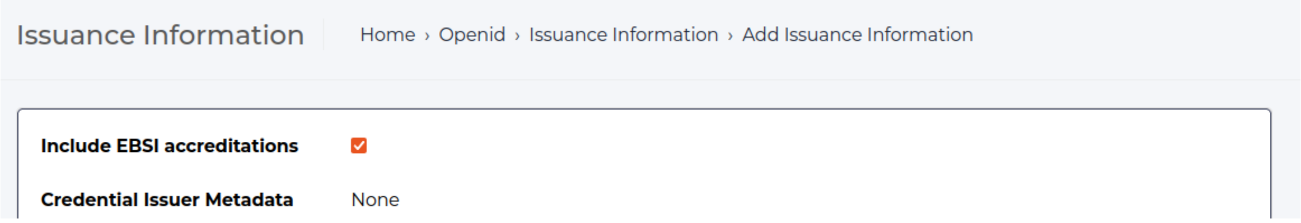
Accreditation to attests

Home › Ebsi › Accreditation to attests › AccreditationToAttest object (5)

DID *	did:ebsi:zw32Uk3dCAFc8y84NHL8Dq6
Schema *	× ['VerifiableCredential', 'VerifiableAttestation', 'CTRevoca...

Figure 28: Accreditation to attest white list

Users whose DIDs have been registered through this system can request these Verifiable Credentials as they would any other. The credential offers generated by the Wallet can be extended to include these accreditations. In the "Issuance Information" model, this is done by simply checking the "Include EBSI Accreditations" box.



Issuance Information

Home › Openid › Issuance Information › Add Issuance Information

Include EBSI accreditations	☑
Credential Issuer Metadata	None

Figure 29: Issuance Information - Including accreditations

No integration with the Authentic Source is required for accreditations. In fact, the Wallet requires no external integration for this type of issuance, unlike other credential types. This is because accreditations do not depend on external business logic—the Wallet holds all the necessary information in its internal database to issue them autonomously.

In addition to the standard OID4VCI protocol for obtaining accreditations, the Wallet also supports direct issuance of credentials without running any protocol. This is particularly useful when the receiving entity is trusted, and credentials are to be transmitted via auxiliary channels such as email or chat. In such cases, the Wallet API must be used, specifically the ***/credentials/ebsi/accreditation*** endpoint.