Find care organizations and their endpoints

Before you can query another care organization, you need its API endpoint and its authorization server URL. Find both by searching the use case's Discovery Service through your own Nuts node.

Before you start

- The node has the use case's Discovery Service definition (see chapter "Configure the node for a use case").

Search

```
GET <internal Nuts interface>/internal/discovery/v1/<service ID>?<path in the credential>=<value>
```

- `<service ID>` is the `id` field of the Discovery Service definition.
- Without query parameters, you get all registered care organizations.

Query parameters are paths in the registered credentials:

- Write the path without `$.`, e.g. `credentialSubject.subject.0`.
- Field IDs from the definition, like `organization_name`, don't work as query parameters.
- Write the value with the same upper and lower case as in the credential. Use `*` at the start or end of a value as a wildcard.
- With several query parameters, all of them must match the same credential.
- Only text fields can be searched; arrays, numbers and true/false values can't.

Which paths you can search on depends on the credentials the use case requires.

360-graden-clientbeeld

Name and city are in the X509Credential, so you can combine them:

Search by	Query parameter
Name	`credentialSubject.subject.O=Zorgorganisatie*`
City	`credentialSubject.subject.L=Utrecht`

```
GET <internal Nuts interface>/internal/discovery/v1/360-graden?credentialSubject.subject.O=Zorgorganisatie*
```

Your node searches its own copy of the Discovery Service. It updates that copy every 10 minutes by default (`discovery.client.refreshinterval`), so a new registration can take that long to show up.

The result

Part	What it is
`credential_subject_id`	The care organization's DID
`fields`	Values taken from the credentials, named by the field IDs in the Discovery Service definition
`registrationParameters`	The values the care organization registered, e.g. its API endpoints
`registrationParameters.authServerURL`	The authorization server URL. You need it to request an access token (see "Request an access token")

Which fields and registration parameters you get depends on the use case.

360-graden-clientbeeld


```
[
  {
    "id": "did:web:nuts.other.example:iam:2f1b7c3e-0d1a-4a7e-9a55-4b8f3c2e9d10#7c4e...",
    "credential_subject_id": "did:web:nuts.other.example:iam:2f1b7c3e-0d1a-4a7e-9a55-4b8f3c2e9d10",
    "fields": {
      "organization_id": "did:web:nuts.other.example:iam:2f1b7c3e-0d1a-4a7e-9a55-4b8f3c2e9d10",
      "organization_name": "Zorgorganisatie B",
      "organization_city": "Utrecht",
      "organization_ura": "87654321",
      "authorization_server_url": "https://nuts.other.example/oauth2/careorg-b-acc",
      "fhir_base_url_r4": "https://fhir.other.example/r4"
    },
    "registrationParameters": {
      "authServerURL": "https://nuts.other.example/oauth2/careorg-b-acc",
      "fhirR4BaseURL": "https://fhir.other.example/r4"
```

```
    },
    "vp": { ... }
  }
]
```

`organization_ura` is the bare URA, because the definition's pattern has a capture group.

API reference: Discovery Service

Revision #8
Created 2025-11-20 07:46:44 UTC by Rein Krul
Updated 2026-11-07 08:05:24 UTC by Rein Krul