# Register on the Discovery Service

Register the care organization on the use case's Discovery Service, so other parties can find it and its API endpoints. In the Nuts node this is called "activating" the Discovery Service for a subject.

Do this once per care organization, per use case. The node then keeps the registration up to date by itself.

## Before you start

- The node has the use case's Discovery Service definition (see chapter "Configure the node for a use case").
- The care organization has a subject (see "Create a subject").
- The wallet has the credentials the Discovery Service definition requires (see "Load the credential into the wallet").
- You know which registration parameters the use case requires, see below.

## Registration parameters

Registration parameters are values you publish with the registration, typically API endpoints. The use case says which ones you need. The node puts them in a DiscoveryRegistrationCredential, so you don't need to issue that credential yourself.

The node always adds `authServerURL`: your Nuts node's OAuth2 authorization server URL for this subject.

**360-graden-clientbeeld**

The wallet only needs the X509Credential. Registration parameters:

| Parameter | Required | Value |
|---|---|---|
| `fhirR4BaseURL` | At least one of the two | Base URL of your FHIR R4 API |
| `fhirStu3BaseURL` | At least one of the two | Base URL of your FHIR STU3 API |

The service ID is `360-graden`.

## Using the API

```
POST <internal Nuts interface>/internal/discovery/v1/<service ID>/<subject ID>
Content-Type: application/json

{
  "registrationParameters": {
    "<parameter>": "<value>"
  }
}
```

- `<service ID>` is the `id` field of the Discovery Service definition.
- `<subject ID>` is the care organization's subject ID, e.g. `careorg-a-acc`.

**360-graden-clientbeeld**

```
POST <internal Nuts interface>/internal/discovery/v1/360-graden/careorg-a-acc
Content-Type: application/json

{
  "registrationParameters": {
    "fhirR4BaseURL": "https://fhir.example.com/r4"
  }
}
```

The node registers right away:

| Response | Meaning | What to do |
|---|---|---|
| `200 OK` | Registered. The node refreshes the registration by itself before it expires. | Nothing |
| `412 Precondition Failed` | Not registered. The response says why, e.g. missing credentials or the Discovery Server can't be reached. The node does **not** retry by itself. | Fix the cause and call the API again |
| `404 Not Found` | Unknown service ID or subject ID | Check both IDs |

# Check that it worked

Get the registration status:

```
GET <internal Nuts interface>/internal/discovery/v1/<service ID>/<subject ID>
```

- `activated` should be `true` and `status` should be `active`.
- If `status` is `error`, `error` says what went wrong, e.g. a refresh that failed.

Then check that others can find you, by searching for your own care organization, e.g. by name:

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

> **360-graden-clientbeeld**
>
> The name is in the X509Credential:
>
> ```
> GET <internal Nuts interface>/internal/discovery/v1/360-
> graden?credentialSubject.subject.O=Zorgorganisatie%20A*
> ```

- The query parameter is a path in the credential (without `$.`), not a field ID from the definition.
- Write the value with the same upper and lower case as in the credential. `*` is a wildcard at the start or end.
- Your care organization should be in the result, with your DID in `credential_subject_id` and your registration parameters in `registrationParameters`.

See "Find care organizations and their endpoints" for more search options.

# Changing or removing the registration

- **Change the registration parameters**, e.g. a new FHIR URL: call the `POST` again with the new values.
- **Remove the registration**:

  ```
  DELETE <internal Nuts interface>/internal/discovery/v1/<service ID>/<subject ID>
  ```

  `200 OK` means it was removed. `202 Accepted` means the node stopped refreshing, but couldn't remove the registration from the Discovery Server; it disappears there when it expires.

API reference: Discovery Service

---

Revision #13
Created 2025-08-12 15:01:00 UTC by Jorrit Spee
Updated 2026-11-07 08:05:12 UTC by Rein Krul