

Test your setup

Check each part of your setup, from the node itself to a query with another party.

- [Test your setup step by step](#)

Test your setup step by step

Check your setup in the order you built it. Each check tells you which part works; if one fails, fix it before going on.

#	Check	Tests
1	The node runs	Deployment
2	Others can reach your node	Reverse proxy, TLS, URL rewrite
3	The wallet has the credentials	Credentials
4	You're registered and can be found	Discovery Service registration
5	You can get an access token from yourself	Policy definition, credentials, token flow, DPoP
6	You can query another party	Everything, with a real other party

1. The node runs

```
GET <internal Nuts interface>/health
```

Should return `200 OK` with status `UP`. If not, check the node's log.

2. Others can reach your node

From **outside** your network:

- Resolve a subject's DID with the [Universal Resolver](#) (see "Create a subject").
- Open your OAuth2 metadata: `https://<your node>/well-known/oauth-authorization-server/oauth2/<subject ID>`. It should return JSON with `"issuer": "https://<your node>/oauth2/<subject ID>"`. If your node runs on a subpath, use the URL from "Check that it works" in "Configure url rewrite for OAuth" instead.

If this fails, check your reverse proxy, TLS certificate and URL rewrite (see chapter "Deploy").

3. The wallet has the credentials

```
GET <internal Nuts interface>/internal/vcr/v2/holder/<subject ID>/vc
```

The result should contain every credential the use case requires (see "Load the credential into the wallet").

4. You're registered and can be found

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

`activated` should be `true` and `status` `active`. Then search for your own care organization (see "Register on the Discovery Service").

5. Get an access token from yourself

Request a token from your own authorization server, as if you were another party. This tests your policy definition, your credentials and the whole token flow.

```
POST <internal Nuts interface>/internal/auth/v2/<subject ID>/request-service-access-token
Content-Type: application/json
```

```
{
  "authorization_server": "https://<your node>/oauth2/<subject ID>",
  "scope": "<scope>",
  "credentials": [ <user credentials, if the use case requires them> ]
}
```

`authorization_server` is your own `authServerURL`. Your node reaches it through its public URL, so this also tests your reverse proxy.

The result is an access token (see "Request an access token"). Then check it as you would an incoming request (see "Validate the access token"):

1. Introspect the token. `active` should be `true`, and the fields from the policy definition should be there.
2. Create a DPoP header for a test URL (see "Make the FHIR call"), and validate it with the same method and URL. `valid` should be `true`.
3. Validate the same DPoP header again. Now `valid` should be `false`, with reason `jti already used`: a DPoP header can only be used once.

Response to the token request	Likely cause
412 Precondition Failed	The wallet doesn't have the credentials the policy requires
400 Bad Request	Wrong scope (no policy for it), or a wrong <code>credentials</code> value
503 Service Unavailable	Your node can't reach its own public URL

360-graden-clientbeeld

Use scope `360-graden`, and pass a `NutsEmployeeCredential` with test values:

```
{
  "authorization_server": "https://<your node>/oauth2/careorg-a-acc",
  "scope": "360-graden",
  "credentials": [
    {
      "@context": ["https://www.w3.org/2018/credentials/v1",
        "https://nuts.nl/credentials/v1"],
      "type": ["VerifiableCredential", "NutsEmployeeCredential"],
      "credentialSubject": { "name": "Test User", "roleName": "Tester", "identifier":
        "test-1" }
    }
  ]
}
```

The introspection result should contain `organization_id`, `organization_ura`, `organization_name`, `organization_city`, `employee_name`, `employee_role` and `employee_identifier`.

6. Query another party

In a test environment, agree with another vendor to test with each other's care organizations:

1. Find their care organization (see "Find care organizations and their endpoints").
2. Request a token from their authorization server, and make a query (see chapter "Outbound: querying another party").
3. Let them do the same with yours, and check your logs.