

SOLID Access

dinsdag 20 december 2022 13:46

Access request/grants links

dinsdag 20 december 2022 13:46

- [General](#)
- [Shapes](#)
- Access request doc:



DNBA-Issui
nganAcce...

- Access grant doc:



DNBA-Appr
ovinganA...

Data examples based on DPV

dinsdag 20 december 2022 17:06

The examples illustrate how a Solid access request and grant would look like if they were aligned with DPV.

We made 2 examples:

- [Access request example](#)
- [Access grant example](#)

Based on:

- [DPV-owl](#)
- [DPV-skos](#)

(version 0.8)

We only illustrate the attributes found in de documentation (see [Access request/grants links](#)). For a more complete example of a Consent instancet see [Example](#) and [Time example](#).

For OSLO-Consent based on DPV: [link](#).

The examples were tested for expansion in the [JSON-Playground](#).

Access request example

dinsdag 20 december 2022 17:07

Based on a data example described in Access request documentation (see [Access request/grants links](#)).

It describes a request by Acme.be to have access to "alarmbel" data in the Solid pod of Isabelle Janssens.

In DPV-terms, Acme.be is a DataController who wants to do some PersonalDataHandling pertaining to PersonalData from Isabelle Janssens who is thus a DataSubject.

The example:

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://vc.inrupt.com/credentials/v1",
    {
      "@vocab": "https://w3id.org/dpv/dpv-owl#",
      "hasPersonalDataHandling": {
        "@reverse": "hasDataController"
      },
      "dpv-skos": "https://w3id.org/dpv/dpv-skos#",
      "dpv-pd": "https://w3id.org/dpv/dpv-pd#",
      "dct": "http://purl.org/dc/elements/1.1/",
      "cl-pdt":
        "https://burgerprofiel.data.vlaanderen.be/concept/persoonlijkedatatype#"
    }
  ],
  "@type": [
    "VerifiableCredential",
    "SolidAccessRequest"
  ],
  "credentialSubject": {
    "@id": "https://acme.be",
    "@type": [
      "Organisation",
      "DataController"
    ],
    "hasPersonalDataHandling": {
      "@type": "PersonalDataHandling",
      "hasProcessing": {
        "@type": "Processing",
        "dct:type": {
          "@type": "skos:Concept",
          "@id": "dpv-skos:Access"
        }
      },
    },
    "hasPersonalData": {
      "@type": "PersonalData",
      "dct:type": [
        {
          "@type": "skos:Concept",
          "@id": "cl-pdt:alarmbel"
        }
      ],
    },
    "dct:references": "https://storage.sandbox-
pod.datanutsbedrijf.be/d8486cf3-eafb-41be-84ee-fad08a7b3d6b/alarmbel/"
  },
  "hasDataSubject": "https://beta.webid.burgerprofiel.dev-
vlaanderen.be/profiles/isabeljanssens",
  "hasLegalBasis": {
    "@type": "Consent",
    "hasConsentStatus": {
      "@type": "skos:Concept",
      "@id": "dpv-skos:ConsentRequested"
    }
  }
}
```

```

    }
  }
}

```

Comments:

- PersonalDataHandling is the central class in DPV, so we start with the link from the DataController to the PersonalDataHandling and go from there to the Consent.
- To go from the DataController (which is Acme.be) to the PersonalDataHandling we use the @reverse of PersonalDataHandling.hasDataController which we called "hasPersonalDataHandling".
- Compare this with the Gconsent:hasConsent (which is actually incorrectly used by INRUPT because the domain of Gconsent:hasConsent is a dataSubject and not a dataController).
- To type the PersonalData we added an example codelist cl-pdt as "alarmbel" is not among the PersonalDataTypes in dpv-pd.
- We used dcterms to type the PersonalData and Processing.
- To refer to the actual PersonalData we used dcterms:references.
- We substituted the mode-attribute with the Access type from dpv-skos.

Access grant example

dinsdag 20 december 2022 17:08

Based on a data example described in Access grant documentation (see [Access request/grants links](#)). It describes consent by Isabel Janssens to allow access to "alarmbel" data from her Solid pod to Acme.be.

In DPV-terms, Isabelle Janssens is a DataSubject for the PersonalDataHandling from her PersonalData by Acme.be which is thus a DataController.

The example:

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://vc.inrupt.com/credentials/v1",
    {
      "@vocab": "https://w3id.org/dpv/dpv-owl#",
      "providedConsent": {
        "@reverse": "hasDataSubject"
      },
      "dpv-skos": "https://w3id.org/dpv/dpv-skos#",
      "dpv-pd": "https://w3id.org/dpv/dpv-pd#",
      "dct": "http://purl.org/dc/elements/1.1/",
      "cl-pdt": "https://burgerprofiel.data.vlaanderen.be/concept/persoonlijkedatatype#"
    }
  ],
  "@id": "https://vc.pod.datanutsbedrijf.be/vc/b34da5d2-8ab1-4e3f-998e-fc84afa2cae8",
  "@type": [
    "VerifiableCredential",
    "SolidAccessGrant"
  ],
  "credentialStatus": "",
  "credentialSubject": {
    "@id": "https://beta.webid.burgerprofiel.dev-vlaanderen.be/profiles/isabeljanssens",
    "@type": [
      "NaturalPerson",
      "DataSubject"
    ],
    "providedConsent": {
      "@type": "Consent",
      "hasConsentStatus": {
        "@type": "skos:Concept",
        "@id": "dpv-skos:ConsentGiven"
      },
      "hasDataSubject": "https://beta.webid.burgerprofiel.dev-vlaanderen.be/profiles/isabeljanssens",
      "hasPersonalDataHandling": {
        "@type": "PersonalDataHandling",
        "hasProcessing": {
          "@type": "Processing",
          "dct:type": {
            "@type": "skos:Concept",
            "@id": "dpv-skos:Access"
          }
        }
      },
      "hasPersonalData": {
        "@type": "PersonalData",
        "dct:type": [
          {
            "@type": "skos:Concept",
            "@id": "cl-pdt:alarmbel"
          }
        ]
      },
      "dct:references": "https://storage.sandbox-
```

```

    pod.datanutsbedrijf.be/d8486cf3-eafb-41be-84ee-fad08a7b3d6b/alarmbel/"
    },
    "hasDataController": "https://acme.be"
  },
  "expirationDate": "",
  "issuanceDate": "",
  "issuer": "",
  "proof": ""
}
}
}

```

Comments:

- To go from the DataSubject (which is Isabel Janssens) to the Consent we use the @reverse of Consent.hasDataSubject which we called "providedConsent".
- Compare this with the Gconsent:providedConsent.
- To type the PersonalData we added an example codelist cl-pdt as "alarmbel" is not among the PersonalDataTypes in dpv-pd.
- We used dcterms to type the PersonalData and Processing.
- To refer to the actual PersonalData we used dcterms:references.
- We substituted the mode-attribute with the Access type from dpv-skos.
- We left VC-attributes like issuer, proof etc empty for our purpose.