Handleiding mappingtool

Blader door een standaard en bouw het pad dat een gegeven volgt.

Met de mappingtool blader je door een erfgoedstandaard — OSLO Cultureel Erfgoed, CIDOC-CRM, FRBRoo of LRMer — en zoek je uit **waar een gegeven thuishoort**: welk pad het volgt van klasse, via eigenschap, naar het verwachte type. Dat pad kopieer je als tekst naar je eigen documentatie of mappingbestand.

Zo ziet een pad eruit: `MensgemaaktObject` → `titel` → `TaalString`

Wil je niet alleen het pad opzoeken, maar een concreet erfgoedstuk beschrijven als JSON-LD? Gebruik dan de Mapping Wizard (handleiding).

1. Kies een standaard

Op het dashboard kies je onder **Mapping pattern tool** de standaard waarin je wil werken. Voor de meeste registratievragen is dat **OSLO Cultureel Erfgoed**; CIDOC-CRM, FRBRoo en LRMer staan er voor wie met die modellen werkt.

2. Kies je startklasse

Bovenaan de mapper kies je met *Kies een entiteit* de klasse waar je pad begint en klik je op *Starten*. Begin bij de klasse die je object voorstelt — voor een museaal object is dat meestal `MensgemaaktObject`. Daaronder verschijnt de tabel met alle eigenschappen van die klasse.

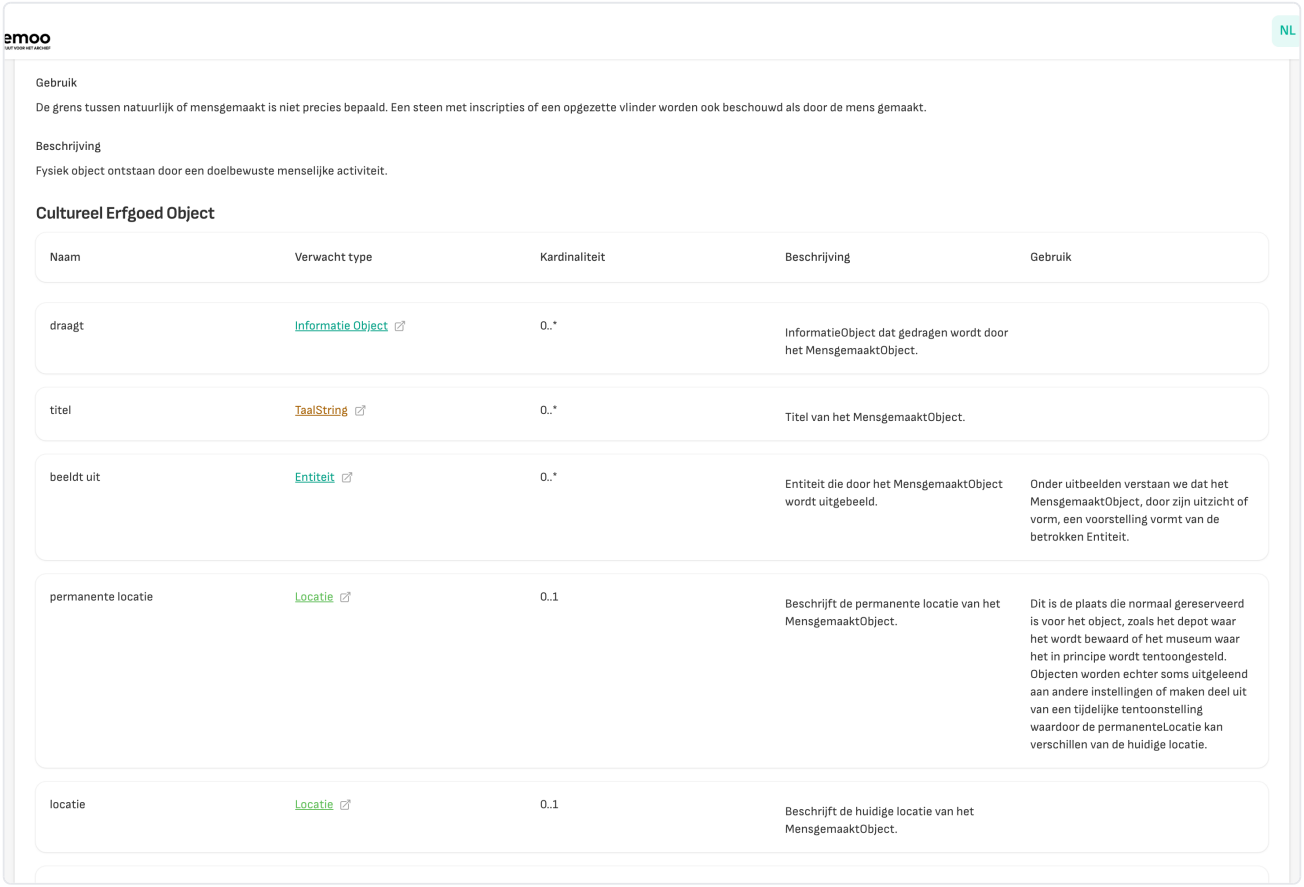
> **Tip:** je ziet ook de eigenschappen die de klasse erft van algemenere klassen. Zo vind je op MensgemaaktObject ook `identificator` en `beschrijving`, die eigenlijk bij `Entiteit` horen — de tabel toont per (super)klasse een eigen blok.

3. Lees de eigenschappentabel

Elke rij is één eigenschap van de klasse. De kolommen:

- **Eigenschap** — de naam, bv. `titel` of `dimensie`.
- **Verwacht type** — wat er als waarde hoort te staan: een andere klasse, een samengesteld datatype (bv. `Dimensie`) of een basistype (bv. `TaalString`, `Getal`).
- **Kardinaliteit** — hoe vaak de eigenschap mag voorkomen: `1` = precies één, `0..1` = hoogstens één, `0..*` = mag herhaald worden.
- **Gebruik** (OSLO) of **Voorbeelden** (CIDOC/FRBR/LRM) — de toelichting uit de standaard zelf.

> **Tip:** in CIDOC-CRM, FRBRoo en LRMer staan de technische codes (`P1_is_identified_by`, `LRM_E4`) bewust naast de namen — zo herken je de elementen uit de officiële documentatie en diagrammen.



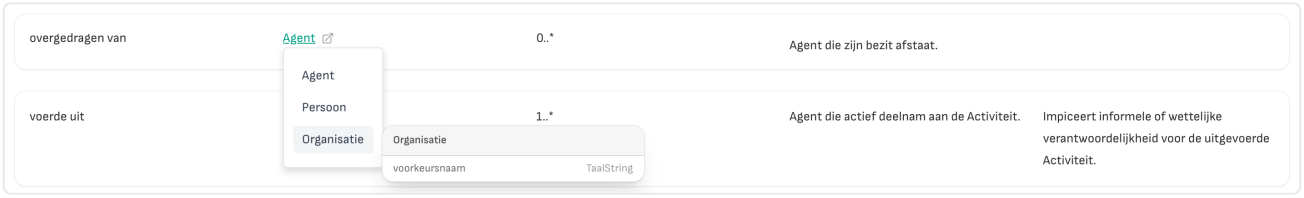
4. Klik door naar het verwachte type

Het verwachte type is klikbaar. Klik erop en de tabel toont de eigenschappen van dát type; bovenaan groeit je pad mee als broodkruimel. Zo klik je stap voor stap van eigenschap naar eigenschap tot je bij een basistype uitkomt — het eindpunt waar de eigenlijke waarde (tekst, getal, datum) hoort.

Verkeerd geklikt? Klik op een eerdere stap in de keten om daarnaartoe terug te keren, of gebruik de knop *stap terug* naast de keten om alleen de laatste stap ongedaan te maken.

> **Voorbeeld:** een afmeting vastleggen loopt in twee stappen: `MensgemaaktObject` → `dimensie` → `Dimensie` → `waarde` → `Getal`. `Dimensie` is een tussenstation met eigen eigenschappen (`type`, `waarde`, `eenheid`); `Getal` is het eindpunt.

Heeft een verwacht type meerdere subklassen, dan verschijnt bij het klikken een keuzelijstje en kies je eerst welke subklasse je bedoelt. Beweeg over een optie en een kaartje toont welke eigenschappen die keuze oplevert — zo zie je meteen dat alleen `Persoon` een voor- en achternaam heeft en alleen `Organisatie` een voorkeursnaam.



5. Achter de schermen: het model

Wat je in de tabellen ziet, is een **ontologie**: een netwerk van klassen die met eigenschappen naar elkaar verwijzen. Vier begrippen verklaren zo goed als alles wat je onderweg tegenkomt:

- **Klassen (entiteiten)** — de bouwstenen van de standaard: `MensgemaaktObject`, `Persoon`, `Organisatie`. Elke klasse heeft zijn eigen eigenschappentabel. In de paden en links herken je klassen aan de **groene** kleur.
- **Sub- en superklassen** — klassen vormen een hiërarchie: `MensgemaaktObject` is een subklasse van `Entiteit`, en **erft al haar eigenschappen**. Daarom zie je onder een klasse ook de tabellen van haar superklassen staan — `identificator` komt niet van `MensgemaaktObject` zelf maar van `Entiteit`, en is er toch gewoon bruikbaar. Andersom: heeft een verwacht type subklassen, dan kies je bij het klikken eerst welke subklasse je bedoelt.
- **Samengestelde datatypes** — **oranje** links zoals `Dimensie` of `Identificator`: tussenstations met eigen eigenschappen (`type`, `waarde`, `eenheid`), maar zonder plaats in de klassenhiërarchie. Je vult er zelf niets in; je klikt door naar een van hun eigenschappen.
- **Basistypes** — **gele** eindpunten zoals `TaalString`, `GetypeerdeString` of `DateTime`: hier houdt het pad op en hoort de eigenlijke waarde. Welke invoer elk basistype vraagt, lees je in de wizard-handleiding.

Loopt een pad dood, dan zegt de tabel dat letterlijk:

> *String heeft geen eigenschappen binnen OSLO Cultureel Erfgoed. Je kan dit pad zo bewaren, dat levert een object op met enkel een type. Ga anders terug en kies een andere eigenschap.*

Dat is geen fout: een basistype (of een klasse zonder eigenschappen binnen deze standaard) heeft nu eenmaal geen eigen tabel om naar door te klikken. Bewaar je het pad tóch op dit punt zonder waarde, dan komt er in het JSON-LD alleen een object met een `@type` te staan. Meestal wil je een stap terug om een andere eigenschap te kiezen — of gewoon hier de waarde invullen als dit je bedoelde eindpunt is.

6. Kopieer je pad

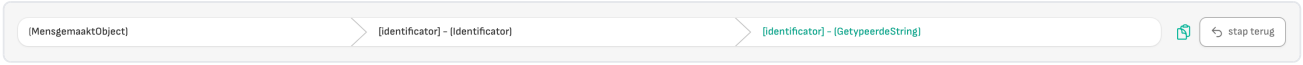
Naast de broodkruimel staat een kopieerknop. Die zet het volledige pad als tekst op je klembord, klaar om in je documentatie of mappingbestand te plakken:

```
(MensgemaaktObject) - [titel] - (TaalString)
```

In CIDOC-CRM, FRBRoo en LRMer gebruikt de gekopieerde tekst de technische namen met hun prefix, zoals in de modeldiagrammen:

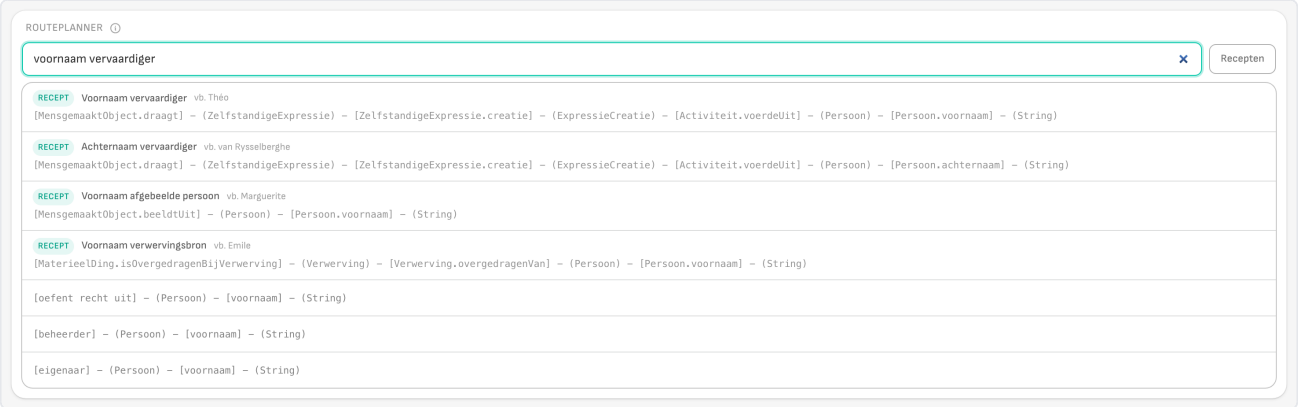
```
(crm:E22_Human-Made_Object)-[crm:P1_is_identified_by]-(crm:E42_Identifier)
```

Elk gekopieerd pad komt in de mappinggeschiedenis rechts op de pagina, zodat je het later kunt terugvinden. Die geschiedenis wordt alleen in je eigen browser bewaard.

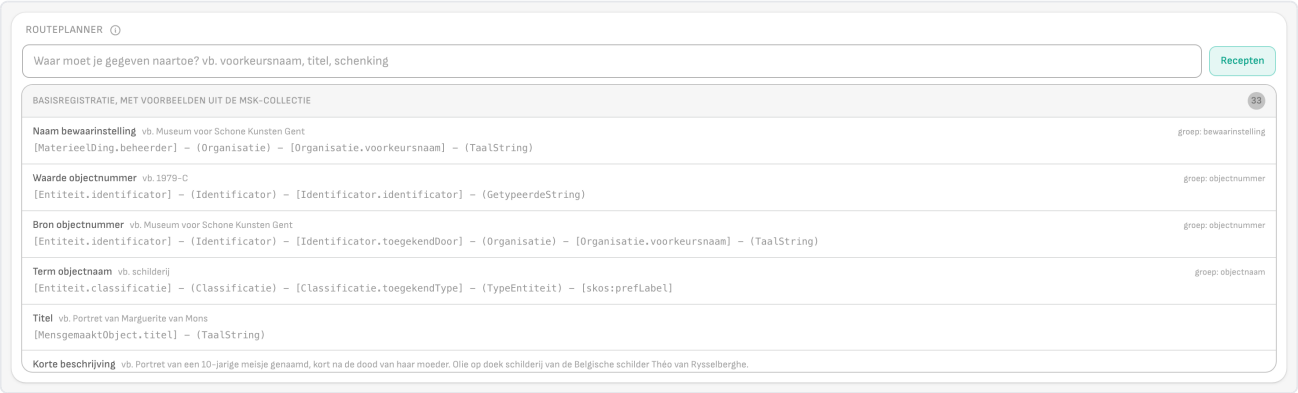


7. Zoek een pad of kies een recept

Je hoeft de route naar een gegeven niet uit je hoofd te kennen. Zodra je keten een startklasse heeft, staat boven de keten een zoekveld: typ waar je gegeven over gaat — *voorkeursnaam*, *titel*, *schenking* — en de tool zoekt zelf de paden vanaf je startklasse die daarbij passen. Klik een voorstel aan en de keten wordt voor je opgebouwd, stap voor stap, precies alsof je zelf klikte. De recepten dragen voorbeeldwaarden uit werken van de MSK-collectie, zodat je ook op registratietermen en concrete waarden kan zoeken.



Werk je in OSLO Cultureel Erfgoed vanaf `MensgemaaktObject`, dan staat naast het zoekveld ook de knop **Recepten**: de velden van de basisregistratie (titel, objectnummer, objectnaam, vervaardiger, herkomst …) als kant-en-klare paden, elk met een voorbeeldwaarde uit het referentiestuk *Portret van Marguerite van Mons*. De zoekfunctie kijkt ook in die recepten, zodat registratietermen als *schenking* of *hoogte* een route vinden terwijl die woorden nergens in het model voorkomen.



8. Eigenschap toevoegen of ontlenen

Via *Eigenschap toevoegen* voeg je een eigenschap aan je pad toe zonder de tabel te verlaten. Het venster toont bovenaan het standaard verwachte type en daaronder de mogelijke subtypes, met het standaardtype al aangevinkt — bevestigen volstaat voor het gewone geval.

Wil je het verwachte type vervangen door een klasse uit een *andere* standaard (bv. een CIDOC-klasse in een OSLO-pad), klap dan *Andere klasse kiezen* open en zoek de klasse op. Zo "ontleen" je elementen tussen standaarden.

9. Gecontroleerde termen (+skos)

Bij type-eindpunten die met een gecontroleerde vocabulaire werken (thesaurustermen zoals een objectnaam of materiaal) staat een `+ skos:prefLabel`-knopje. Klik je dat aan, dan krijgt je pad een extra stap `skos:prefLabel` — de gangbare manier om de voorkeursterm van zo'n concept vast te leggen.

Veelgestelde vragen

Waarom zie ik Engelse namen in de LRMer-mapper?

LRMer is een Engelstalige standaard; de mapper toont daar bewust de technische identifiers uit de officiële IFLA-diagrammen, zodat wat je kopieert één op één overeenkomt met de gepubliceerde documentatie.

Waar wordt mijn mappinggeschiedenis bewaard?

Lokaal in je browser, niet op de server. Op een andere computer of in een andere browser begin je dus met een lege geschiedenis.

De eigenschappenlijst van een klasse is leeg.

Sommige klassen zijn wel aan het OSLO-profiel gekoppeld maar hebben er zelf geen eigenschappen in — hun eigenschappen leven in een andere standaard, zoals CIDOC-CRM. De lijst zegt dan waar ze wél bestaan. Je kan zo'n pad gewoon bewaren (het levert een object met enkel een type op) of teruggaan en een andere eigenschap kiezen.

Ik wil een echt object beschrijven, niet alleen het pad opzoeken.

Daarvoor is de Mapping Wizard: die bouwt dezelfde paden, maar vult ze met jouw waarden en zet alles samen in één JSON-LD-document.