

Logboek Dataverwerkingen Extensie Guidelines



Logius Praktijkrichtlijn
Werkversie 04 juli 2025

Deze versie:

<https://logius-standaarden.github.io/logboek-extensie-template/>

Laatst gepubliceerde versie:

<https://gitdocumentatie.logius.nl/publicatie/logboek/logboek-extensie-template/>

Laatste werkversie:

<https://logius-standaarden.github.io/logboek-extensie-template/>

Redacteurs:

Nil Barua ([Logius](#))

Tim van der Lippe ([Logius](#))

Auteurs:

Nil Barua ([Logius](#))

Tim van der Lippe ([Logius](#))

Doe mee:

[GitHub Logius-standaarden/logboek-extensie-template](#)

[Dien een melding in](#)

[Revisiehistorie](#)

[Pull requests](#)

Dit document is ook beschikbaar in dit niet-normatieve formaat: [pdf](#)



Dit document valt onder de volgende licentie:

[Creative Commons Attribution 4.0 International Public License](#)

Context bij de standaard

De overheid wil voor burgers en bedrijven zo transparant mogelijk zijn in de omgang met hun data. Daarom is het bij de informatieverwerking in datasets belangrijk om voor elke mutatie of raadpleging vast te leggen wie deze actie wanneer uitvoert, en waarom. Voor een optimale samenwerking over organisaties en bronnen heen is voor deze logging een algemene standaard nodig.

Transparantie en verantwoording naar burgers is één van de drijfveren voor ontwikkeling van deze logging standaard maar geen onderdeel van de standaard. De standaard richt zich op vastlegging

van de logging en deze eenduidige vastlegging maakt het mogelijk om, indien daar later behoefte aan is, inzage mogelijk te maken.

Het project Logboek Dataverwerkingen maakt deel uit van het [actieplan Data bij de Bron](#) en onderzoekt met [Digilab](#) of we op basis van de tot nu toe opgedane inzichten een overheidsbrede standaard kunnen vaststellen.

§ Verwijzingen

De Logboek Dataverwerkingen (LDV) standaard bestaat uit de volgende vier documenten:

Beschrijving van het document	Gepubliceerde versie	Werk versie	Repository
1. De LDV Normatieve Standaard	-	Logboek dataverwerkingen (werkversie)	logboek-dataverwerkingen
2. De Algemene Inleiding	-	De Algemene Inleiding (werkversie)	logboek-dataverwerkingen-inleiding
3. Het Juridische Beleidskader	-	Juridisch Beleidskader (werkversie)	logboek-dataverwerkingen-juridisch-beleidskader
4. LDV Extensie Guideline	-	Guideline voor het schrijven van een extensie voor LDV (werkversie)	logboek-extensie-template

§ 1. Feedback en Issues

Dit onderdeel is niet normatief.

We moedigen gebruikers aan om meldingen of suggesties aan te maken via [GitHub](#). Mocht dit niet mogelijk zijn, dan kunt u ook een e-mail sturen naar api@logius.nl.

Status van dit document

Dit is een werkversie die op elk moment kan worden gewijzigd, verwijderd of vervangen door andere documenten. Het is geen door het TO goedgekeurde consultatieversie.

Inhoudsopgave

Context bij de standaard

Verwijzingen

1. Feedback en Issues

Status van dit document

2. Richtlijnen voor Extensies in FSC

2.1 Inleiding

2.2 Randvoorwaarden

2.2.1 Compatibiliteit

2.2.2 Interoperabiliteit

2.3 Verschil tussen de Extensie en de Core Standaard

2.4 Wat moet er in een extensiedocumentatie staan?

2.5 Schrijfwijze en stijl

2.6 Relevante Standaarden

A. Structuur van een extensie

B. Conformiteit

§ 2. Richtlijnen voor Extensies in FSC

§ 2.1 Inleiding

Extensies worden toegevoegd aan een standaard wanneer de core-standaard niet alle benodigde functionaliteiten biedt die de gebruiker van de standaard nodig heeft. Dit kan voorkomen wanneer verschillende sectoren andere eisen stellen aan de standaard.

Bijvoorbeeld:

In de zorgsector moeten logs mogelijk strenger gecontroleerd worden vanwege gevoelige gegevens zoals patiëntendossiers. In het onderwijs kunnen er juist andere gegevens worden gelogd, zoals toetsresultaten of inschrijvingsgegevens, waarbij minder strenge controle nodig is.

Of een extensie noodzakelijk of optioneel is, hangt af van de gebruiker van de standaard en de eisen binnen een sector. Als een organisatie haar taken volledig kan uitvoeren met de core-standaard, is een extensie niet verplicht. Echter, in situaties waarin extra functionaliteit nodig is, kan een extensie een waardevolle toevoeging zijn, bijvoorbeeld om sector-specifieke eisen te ondersteunen.

§ 2.2 Randvoorwaarden

Bij het ontwerpen van een extensie moet rekening worden gehouden met:

- **Compatibiliteit:** De extensie moet voldoen aan de voorgeschreven normen of protocollen van de core standaard.
- **Interoperabiliteit:** De extensie moet correct functioneren binnen de core van de standaard in samenwerking met andere extensies.

§ 2.2.1 Compatibiliteit

De extensie mag **geen** verplichte of benodigde velden van de core-standaard tegenspreken, aanpassen of overschrijven en moet zich houden aan de verplichte velden, tenzij de core-standaard dit expliciet toe staat. Voorbeelden zijn de `resources` en `attributes` velden van het Logboek component die expliciet open zijn voor extensie. Extensies mogen optioneel functionaliteit toevoegen.

De extensie mag geen bestaande protocollen vanuit de core-standaard overschrijven, maar wel extra protocollen als ondersteuning toevoegen als deze compatibel zijn met de core-standaard.

§ 2.2.2 Interoperabiliteit

Interoperabiliteit is nodig om samenwerking tussen verschillende systemen aan te sporen en om te voorkomen dat het niet in conflict komt met andere extensies. De extensie mag de samenwerkingen tussen systemen niet verstoren, vooral als data wordt uitgewisseld tussen verschillende organisaties.

Elke extensie moet een unieke namespace hebben om conflicten te voorkomen. Een extensie mag geen velden gebruiken die al door andere extensies gebruikt worden, hierdoor kunnen er juist conflicten ontstaan. Dit is cruciaal om uniek te houden voor het geval dat een systeem gebruik maakt van verschillende extensies en deze met elkaar wilt laten samenwerken.

§ 2.2.2.1 Belang unieke namespaces a.d.h.v. voorbeeldcase student

In deze voorbeeldcase wordt uitgegaan van een school die informatie wil ophalen over de status van een student binnen een hogeschool en de status van zijn examenrecht in één API-call. Dit is nodig om te zien of de student nog is ingeschreven bij de school en of hij het examen mag afleggen.

Het systeem roept hiervoor een bepaalde API-call aan, namelijk `https://www.schoolapi.nl/student?id=123&status_exam`.

Die krijgt vervolgens het volgende response terug:

```
id: student115839
status: ingeschreven
id: examenNederlands
status: gemachtigd
```

De velden `id` en `status` komen twee keer voor, maar het systeem kan niet herkennen welke van deze velden bij de student horen en welke bij het examen. Hierdoor kan het systeem ten onrechte verkeerde data terugkoppelen, omdat er gebruik gemaakt wordt van dezelfde namespaces voor twee verschillende datavelden.

Maar op het moment dat er wel gebruik gemaakt wordt van unieke namespaces zoals `student` en `exam` en duidelijk onderscheid maakt, kan het systeem de velden wel duidelijk herkennen.

Hiermee kan je gebruik maken van twee extensies tegelijkertijd en krijg je het volgende response terug:

```
dpl.student.id: student115839
dpl.student.status: ingeschreven
dpl.exam.id: examenNederlands
dpl.exam.status: gemachtigd
```

Dit is enkel een voorbeeld van als er gebruik wordt gemaakt van twee extensies, maar op het moment dat er meer dan twee extensies gebruikt worden, kan het overzicht van de response zo lang worden dat het onoverzichtelijk voor de gebruiker wordt.

§ 2.2.2.2 Namespaces Key

In de core standaard is het veld `properties` een object opgebouwd uit velden in een namespace met prefix `naam_van_extensie`. (Omdat de onderliggende keys al in een FSC contract zitten is een prefix als `fsc` overbodig). Het is **verplicht** in de extensie om elke key te beschrijven in het begin met de prefix `naam_van_extensie.naam_van_key`.

Zo mag een key van de extensie bijvoorbeeld:

`regulated_area.name` heten, *maar niet* `name.userid` of enkel `name`.

De naam van de extensie moet altijd beschreven worden in de prefix. Anders kan dit collisions tussen verschillende extensies veroorzaken.

§ 2.2.2.3 Verwijzen naar registers

Als een key verwijst naar een register, dan gelden de volgende design regels:

1. De key mag niet het woord "register" bevatten. Gebruik een naam die aangeeft wat de identifier is, niet hoe het register heet. Dus `dpl.object.algorithm_id`, *maar niet* `dpl.object.algorithm_register_id`
2. Gebruik geen keys die al in de `dpl.core` namespace bevatten. Dus geen `dpl.extensie.processing_activity_id`, omdat `dpl.core.processing_activity_id` al bestaat
3. Een verwijzing naar een register kan zowel 1 item als meerdere items bevatten. Het type van het attribuut kan dus een `String` of een `List` zijn. Voorbeeld van een `List` zou meerdere verwijzingen zijn naar het wetten register, als meerdere wetten van toepassing zijn op datgene wat wordt gelogd

§ 2.3 Verschil tussen de Extensie en de Core Standaard

De extensie voegt aanvullende functionaliteit toe die niet verplicht is, zoals bijvoorbeeld extra velden of regels die bedoeld zijn voor specifieke domeinen.

De core biedt dus alleen generieke functionaliteit en de extensie voegt alleen specifieke mogelijkheden toe voor een bepaalde context.

Als een organisatie haar taken niet volledig kan uitvoeren met de core-standaard, dan komt een extensie goed te pas.

§ 2.4 Wat moet er in een extensiedocumentatie staan?

Elke extensie moet minimaal de volgende onderdelen bevatten:

- **Nut:** Wat is het doel van deze extensie? Voor wie is deze extensie ontworpen?
- **Technische details:** Hoe werkt de extensie in samenwerking met de core-standaard? Wat zijn de toegevoegde functionaliteiten vanuit de extensie op de core-standaard?
- **Benoem voorbeeldsituaties van hoe de extensie toegepast kan worden**
- **Beperkingen:** Beschrijf wat je niet met de extensie kan.

§ 2.5 Schrijfwijze en stijl

- Gebruik een formele en eenduidige schrijfstijl.
- Definieer alle termen duidelijk.
- Zorg voor consistente naamgeving.

§ 2.6 Relevante Standaarden

Gebruik referenties naar andere relevante standaarden, om te voorkomen dat de extensie herdefinieert wat in andere standaarden als is vastgelegd. Voor extensies is dit van extra belang omdat ze de relatie beschrijven tussen het Logboek Dataverwerkingen en een domein. Standaardisatie binnen het domein is nodig om ook op een gestandaardiseerde manier te kunnen loggen.

§ A. Structuur van een extensie

Elke extensie moet de volgende structuur hebben:

1. Naam en beschrijving

Hierin wordt de naam van de extensie beschreven en wat het inhoudt.

2. Doel en nut

Hierin moet beschreven worden wat het doel is van deze extensie. Waar dient de extensie voor? Wie is het doelgroep hiervoor en welke toepassingsgebieden zijn er voor deze extensie?

3. Technische specificatie

Beschrijf hierin de aanvullende technische specificaties op de core standaard.

4. Gebruiksscenario's

In dit kopstuk worden de verschillende use cases van de extensie beschreven

5. Versiebeheer

Hierin wordt de versiebeheer beschreven.

§ B. Conformiteit

Naast onderdelen die als niet normatief gemarkeerd zijn, zijn ook alle diagrammen, voorbeelden, en noten in dit document niet normatief. Verder is alles in dit document normatief.