Logius Praktijkrichtlijn - Vastgestelde versie

Logboek Dataverwerkingen Extensie Guidelines 1.0.0

Logius Praktijkrichtlijn
Vastgestelde versie 09 april 2026

Deze versie:

https://gitdocumentatie.logius.nl/publicatie/logboek/extensie-template/1.0.0/

Laatst gepubliceerde versie:

https://gitdocumentatie.logius.nl/publicatie/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

Status van dit document

Dit is de definitieve versie van dit document. Wijzigingen naar aanleiding van consultaties zijn doorgevoerd.

Inhoudsopgave

§ 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.

De trefwoorden *MAG*, *MOET*, *MOET NIET* en *MOGEN* in dit document moeten worden geïnterpreteerd als in BCP 14 [*RFC2119*] [*RFC8174*] als, en alleen als deze in hoofdletters zijn weergegeven, zoals hier getoond.

Abstract

Dit document is onderdeel van een standaard voor het loggen van dataverwerkingen in de Nederlandse publieke sector. De governance van deze standaard wordt beschreven in het API-Standaarden beheermodel, gepubliceerd door Logius.

§ 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, via een extensie, inzage mogelijk te maken.

§ 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. Richtlijnen voor Extensies in Logboek Dataverwerkingen

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. Of bijvoorbeeld in het onderwijs kunnen er 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.

1.1 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.

1.1.1 Compatibiliteit

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

1.1.2 Interoperabiliteit

Interoperabiliteit is nodig om samenwerking tussen verschillende systemenen 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 *MOET* 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.

1.1.2.1 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.

VOORBEELD 1

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.

§ *1.1.2.2 Namespaces Key*

In de core standaard is het veld `attributes` een object opgebouwd uit velden in een namespace met prefix `dpl.` (data processing log). Elk extra attribuut *MOET* de prefix `dpl.extensienaam` hebben, anders kan dit collisions tussen verschillende extensies veroorzaken.

> VOORBEELD 2
>
> Zo mag een key van de extensie *zorg* bijvoorbeeld:
>
> `dpl.zorg.userid` heten, *maar niet* `dpl.userid.zorg` of enkel `dpl.userid`.

1.1.2.3 Verwijzigen naar registers

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

1. De key *MOET* 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. Er *MOET* geen overlap zijn met 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 *MAG* 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 verwijzigen zijn naar het wetten register, als meerdere wetten van toepassing zijn op datgene wat wordt gelogd

1.2 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.

1.3 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.

§ 1.4 Schrijfwijze en stijl

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

§ 1.5 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. Referenties

§ B.1 Normatieve referenties

[RFC2119]

Key words for use in RFCs to Indicate Requirement Levels. S. Bradner. IETF. March 1997. Best Current Practice. URL: https://www.rfc-editor.org/rfc/rfc2119

[RFC8174]

Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words. B. Leiba. IETF. May 2017. Best Current Practice. URL: https://www.rfc-editor.org/rfc/rfc8174

↑