Logius Standaard - Vastgestelde versie

Digikoppeling Koppelvlakstandaard REST-API 3.0.0

Logius Standaard
Vastgestelde versie 15 mei 2025

Deze versie:
https://gitdocumentatie.logius.nl/publicatie/dk/restapi/3.0.0/

Laatst gepubliceerde versie:
https://gitdocumentatie.logius.nl/publicatie/dk/restapi/

Laatste werkversie:
https://logius-standaarden.github.io/Digikoppeling-Koppelvlakstandaard-REST-API/

Vorige versie:
https://gitdocumentatie.logius.nl/publicatie/dk/restapi/2.0.1/

Redacteur:
Standaardenbeheer (Logius)

Auteurs:
Peter Haasnoot (Logius)
Pieter Hering (Logius)

Doe mee:
GitHub Logius-standaarden/Digikoppeling-Koppelvlakstandaard-REST-API
Dien een melding in
Revisiehistorie
Pull requests

Dit document is ook beschikbaar in dit niet-normatieve formaat: pdf

Samenvatting

Dit document beschrijft de functionele specificaties voor de Digikoppeling Koppelvlakstandaard REST-API. Het document is bestemd voor architecten en ontwikkelaars die op basis van REST-API's gegevens willen uitwisselen via Digikoppeling.

Het Digikoppeling REST-API profiel is gebaseerd op de *API Design Rules* zoals ontwikkeld door het Kennisplatform API's en in beheer gebracht bij Logius.

Status van dit document

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

Inhoudsopgave

§ 1. 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.

§ 2. Context voor ontwikkeling van het Digikoppeling REST API profiel

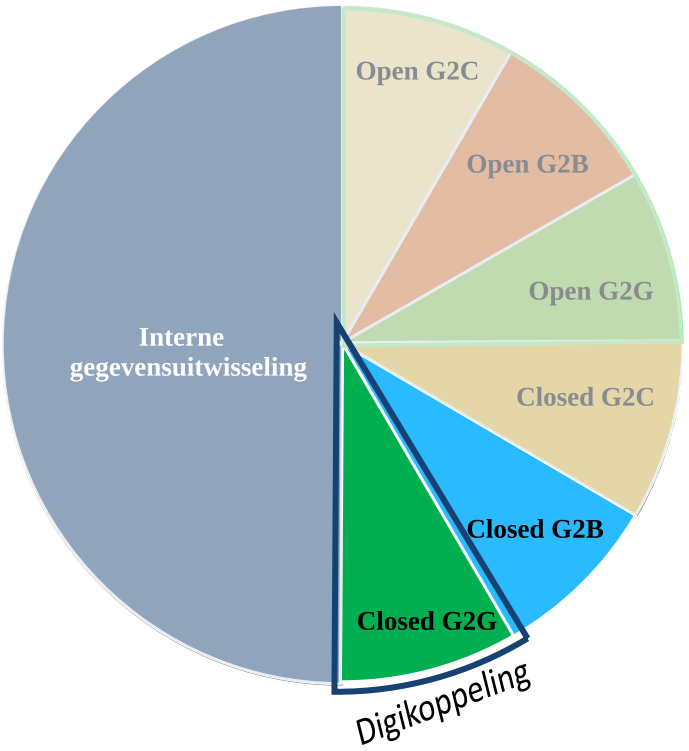
Het Digikoppeling Rest API profiel is gericht op Machine-to-Machine (M2M) en Government-to-Government (G2G) interacties conform de algemene uitgangspunten van de Digikoppeling standaard en het toepassingsgebied van Digikoppeling op de Pas-toe-of-leg-uit lijst (PTLU) van het Forum Standaardisatie;

Opzet Digikoppeling:

<table>
<tr><td rowspan="5">Koppelvlak Specifiek</td><td>ebMS</td><td>WUS</td><td>REST API</td><td></td></tr>
<tr><td>Grote Berichten</td><td>Grote Berichten</td><td>Grote Berichten</td><td rowspan="4">End to End</td></tr>
<tr><td>Best Effort</td><td>Best Effort</td><td>Best Effort</td></tr>
<tr><td>Signing</td><td>Signing</td><td>Signing</td></tr>
<tr><td>Encryption</td><td>Encryption</td><td>Encryption</td></tr>
<tr><td rowspan="3">Koppelvlak Generiek</td><td colspan="3">Identificatie (OIN)</td><td rowspan="3">Point to Point</td></tr>
<tr><td colspan="3">Authenticatie (PKIo)</td></tr>
<tr><td colspan="3">Vertrouwelijkheid (mTLS)</td></tr>
</table>

Figuur 1 Overzicht Digikoppeling Koppelvlakken

§ 3. Toelichting bij de scope van het Digikoppeling REST API profiel



Figuur 2 Digikoppeling voor Closed Data G2G Uitwisseling

In de figuur wordt onderscheid gemaakt tussen open en gesloten diensten:

- Open Diensten: Diensten zonder toegangsbeperking bv open data.
- Gesloten Diensten: Diensten met toegangsbeperking bv persoonsgegevens en vertrouwelijke gegevens of diensten voor specifieke partijen.

Het Digikoppeling REST API profiel richt zich op Machine-to-Machine (M2M) gegevensuitwisseling via een gesloten dienst tussen overheidspartijen. Buiten scope van het profiel zijn:

- REST API's voor open diensten van een overheidspartij.
- REST API's voor gesloten diensten van een overheidspartij (direct) aan burgers of bedrijven.

Het Digikoppeling REST API profiel is wat betreft functionele toepassing vergelijkbaar met het Digikoppeling WUS profiel. De client van de dienstafnemer die gebruik maakt van het Digikoppeling REST API profiel is in deze context een systeem (applicatie) en geen internetbrowser.

Invulling Digikoppeling	DK REST API profiel	DK WUS profiel	DK ebMS2 profiel
Bevragingen / Meldingen			
best-effort	1.0	2W-be	osb-be
best-effort signed	2.0	2W-be-S	osb-be-s
best-effort signed/encrypted	2.0	2W-be-SE	osb-be-e
reliable			osb-rm
reliable signed			osb-rm-s
reliable signed en encrypted			osb-rm-e

Vanaf versie 2.0 van het Digikoppeling REST API profiel wordt signing en encryptie ondersteund (zie 4.5 Signing & Encryptie (in HTTP REST Context)).

§ 4. Digikoppeling REST API profiel

§ 4.1 Historie

Vanuit het TO Digikoppeling zijn al langere tijd de ontwikkelingen rond RESTful API's gevolgd. Binnen het Kennisplatform API zijn de REST-API Design Rules (REST ADR) ontwikkeld en de REST ADR standaard is ook opgenomen op de Pas-toe-of-leg-uit lijst van het Forum

Standaardisatie. De REST ADR standaard is dan ook als basis genomen voor dit Digikoppeling REST API Profiel dat zich specifiek richt op G2G (Government-to-Government) interactie en M2M (Machine-to-Machine verkeer). Daarnaast is de standaard Federated Service Connectivity (FSC) ontwikkeld die voorschrijft hoe organisaties REST API's kunnen ontdekken, aanbieden en consumeren. De FSC standaard is opgenomen in dit Digikoppeling REST API Profiel om de koppelingen met REST API's te standardiseren waardoor er een interoperabel API landschap ontstaat.

§ 4.2 Toepassingsgebied

Het toepassingsgebied is voor Digikoppeling:

> Digikoppeling moet worden toegepast bij digitale gegevensuitwisseling die plaatsvindt met voorzieningen die onderdeel zijn van de GDI, waaronder de basisregistraties, of die sectoroverstijgend is. De verplichting geldt voor gegevensuitwisseling tussen systemen waarbij er noodzaak is voor tweezijdige authenticatie.

Dit profiel is toe te passen bij het aanbieden en/of consumeren van REST API's ten behoeve van het ontsluiten van overheidsinformatie en/of functionaliteit.

§ 4.3 Algemeen

Het Digikoppeling REST API profiel is o.a. gebaseerd op de REST-API Design Rules standaard zoals ontwikkeld door het Kennisplatform API's en in beheer gebracht bij Logius Stelsels & Standaarden: [*ADR*]

Het Digikoppeling REST API profiel conformeert zich volledig aan het normatieve deel van de REST-API Design Rules.

Het Digikoppeling REST API profiel maakt gebruik van de FSC-standaard.

§ 4.4 Koppelvlak Generiek

§ 4.4.1 Vertrouwelijkheid

De Digikoppeling Beveiligingsstandaarden en voorschriften gaan specifiek in op het verplichte gebruik van PKIO certificaten [*PKIO-PvE*].

- Zie *Digikoppeling Beveiligingsstandaarden en voorschriften*

§ 4.4.2 Identificatie & Authenticatie

Digikoppeling maakt gebruik van het OIN (Organisatie Identificatie Nummer) voor de identificatie van organisaties. Binnen dit Digikoppeling REST API profiel zijn er alleen voorschriften m.b.t. het verplicht gebruik van het OIN binnen PKIO certificaten en FSC. Voor OIN gebruik binnen payloads (bv JSON) of resource-pad gelden geen specifieke voorschriften.

- Zie *Digikoppeling Identificatie en Authenticatie*

§ 4.4.3 Federated Service Connectivity Standaard (FSC)

Gebruik van de FSC-standaard binnen het Digikoppeling REST API profiel is verplicht [1], [2]

De FSC standaard bestaat uit het hoofddocument *FSC - Core* en een extensie genaamd *FSC - Logging*. Het is verplicht Core en Logging beide te gebruiken.

[1]: *De verplichting valt onder het pas-toe-of-leg-uit beginsel van het Forum Standaardisatie zoals dat geldt voor de Digikoppeling REST-API Koppelvlakstandaard.*

[2]: *Voor bestaande implementaties is het toegestaan tot 1/1/2027 gebruik te maken van versie 1.1 van de Digikoppeling REST-API Koppelvlakstandaard.*

FSC beschrijft het volgende:

1. Hoe de identiteit van een organisatie wordt bepaald en vertrouwd.

2. Hoe een autorisatie om te mogen koppelen met een API gegeven, geweigerd of ontnomen wordt.
3. Hoe organisaties van een netwerk de API's, en elkaar kunnen vinden.
4. Hoe een verbinding naar een API veilig kan worden opgezet.
5. Hoe logregels weggeschreven moet worden.
6. Hoe een intermediar namens een organisatie een API kan consumeren en/of publiceren.

Het Digikoppeling REST API profiel geeft invulling aan keuzes die gemaakt moeten worden bij het gebruik van FSC. In het Digikoppeling REST API profiel wordt er vanuit gegaan dat de lezer bekend is met de standaard FSC. Er worden namelijk termen gebruikt uit deze standaard.

De bovengenoemde functionaliteit is vastgelegd in FSC Core en de extensies Logging en Delegation. Core beschrijft het koppelen met API's, Logging hoe logregels weggeschreven moeten worden en Delegation hoe een intermediar namens een organisatie een API kan consumeren en/of publiceren.

- *FSC - Core*
- *FSC - Logging*

De bovengenoemde functionaliteit is vastgelegd in FSC Core en de extensie Logging. Core beschrijft het koppelen,aanbieden en ontdekken van API's en de extensie Logging beschrijft hoe logregels weggeschreven moeten worden.

§ *4.4.3.1 Vertrouwelijkheid*

FSC spreekt over een Trust Anchor die door een Group moet worden gekozen. De Trust Anchor is binnen de context van X.509 certificaten de certificate authority (CA) waaruit het vertrouwen wordt afgeleid. De Trust Anchor voor de FSC Group moet daarom de PKIO Private Root zijn.

§ *4.4.3.2 Identificatie & Authenticatie*

Het PeerID binnen de context van FSC is OIN. Het OIN wordt bij PKIO certificaten geplaatst in het SerialNumber veld van het Subject. Het is verplicht vanuit FSC om te bepalen welk veld uit het certificaat de Peer name bepaald. Dit is het organization veld van het Subject van het PKIO certificaat. Binnen dit Digikoppeling REST API profiel zijn er alleen voorschriften m.b.t. het verplicht gebruik van het OIN binnen PKIO certificaten en FSC. Voor OIN gebruik binnen payloads (bv JSON) of resource-pad gelden geen specifieke voorschriften.

- Zie *Digikoppeling Identificatie en Authenticatie*

§ *4.4.3.3 TLS*

De Digikoppeling Beveilingsstandaarden en voorschriften veplichten het gebruik van 2-zijdig TLS met minimaal TLS versie 1.2, FSC verscherpt deze eis door de ciphersuites die geen perfect forward secrecy ondersteunen niet toe te laten.

- Zie *Digikoppeling Beveiligingsstandaarden en voorschriften*

§ *4.4.3.4 Netwerk-poorten*

De Digikoppeling Beveiligingsstandaarden en voorschriften verplichten het gebruik van de netwerkpoort 443 voor data verkeer. FSC voegt daar het gebruik van port 8443 voor managementverkeer aan toe. E.g. toegang aanvragen voor een API.

- Zie *Digikoppeling Beveiligingsstandaarden en voorschriften*

§ *4.4.3.5 Contracten*

FSC gebruikt Contracten om afspraken tussen Peers vast te leggen. Een Contract kan één of meerdere Grants bevatten. Een Grant beschrijft welke interactie er mogelijk is tussen de Peers. FSC plaatst geen beperking op het aantal Grants per Contract. Het Digikoppeling REST API profiel doet dit wel om te voorkomen dat er fragiele Contracten ontstaan met een hoge beheerslast. Het aantal Grants wordt beperkt tot maximaal 10.

§ *4.4.3.6 Retry-mechanisme voor versturen van Contracten en hantekeningen*

De Peer die een Contract aanmaakt of een handtekening plaats op een Contract is zelf verantwoordelijk voor het distribureren van het Contract of handtekening naar de Peers op het Contract. In het scenario dat het versturen van Contract of handtekening mislukt verplicht het Digikoppeling REST API profiel het toepassen van een exponential backoff retry-mechanisme.

> Het retry mechanisme betreft niet de HTTP-requests voor het bevragen van een Service.

Een exponential backoff retry-mechanism is een mechanisme dat een mislukt verzoek opnieuw gaat uitvoeren op een interval die exponentieel groeit. Deze exponentiële groei voorkomt dat een applicatie een veelvoud van verzoeken verstuurd naar een service die niet bereikbaar is.

Voorbeeld: Peer A verstuurt een Contract naar Peer B. Het versturen mislukt. Peer A probeert het opnieuw na 1 seconde, het verzoek mislukt weer. De volgende poging wordt gedaan na 2 seconden, daarna 4 seconden, vervolgens 16 seconden, enzovoort. Om te voorkomen dat er langlopende processen worden gecreëerd hanteerd Peer A een maximale interval van 300 seconden.

§ *4.4.3.7 Logging*

De FSC Logging extensie beschrijft een Transaction ID. Een unieke identifier in de vorm van een UUID voor elke transactie die gedaan wordt, i.e. een bevraging van een API. Deze transactie ID wordt weggeschreven bij elke log regel. Het Digikoppeling REST API profiel verplicht het gebruik van een UUID V7 als Transaction ID.

§ 4.4.4 API Design Rules

§ 4.4.5 Regels

Het Digikoppeling REST-API profiel conformeert zich volledig aan het normatieve deel van de *API Design Rules*. Het is verplicht te voldoen aan alle (normatieve) eisen van de REST-API Design Rules:

Regelnaam	Principe
/core/naming-resources	Use nouns to name resources
/core/naming-collections	Use plural nouns to name collection resources
/core/interface-language	Define interfaces in Dutch unless there is an official English glossary available
/core/hide-implementation	Hide irrelevant implementation details
/core/http-safety	Adhere to HTTP safety and idempotency semantics for operations
/core/stateless	Do not maintain session state on the server

Regelnaam	Principe
/core/nested-child	Use nested URIs for child resources
/core/resource-operations	Model resource operations as a sub-resource or dedicated resource
/core/doc-language	Publish documentation in Dutch unless there is existing documentation in English
/core/deprecation-schedule	Include a deprecation schedule when deprecating features or versions
/core/transition-period	Schedule a fixed transition period for a new major API version
/core/changelog	Publish a changelog for API changes between versions
/core/geospatial	Apply the geospatial module for geospatial data
/core/no-trailing-slash	Leave off trailing slashes from URIs
/core/http-methods	Only apply standard HTTP methods
/core/doc-openapi	Use OpenAPI Specification for documentation
/core/publish-openapi	Publish OAS document at a standard location in JSON-format
/core/uri-version	Include the major version number in the URI
/core/semver	Adhere to the Semantic Versioning model when releasing API changes
/core/version-header	Return the full version number in a response header
/core/transport-security	Apply the transport security module

§ 4.4.6 Afspraken API Design Rules modules

Extensies op de *API Design Rules* zijn geschreven in modules. Hieronder wordt aangegeven welke regels uit de API Design Rules modules in dit profiel verplicht zijn of worden aanbevolen.

Categorie	Module	Principe
Niet van toepassing*	Transport Security	Secure connections using TLS
Verplicht**	Transport Security	No sensitive information in URIs
Verplicht	API access control	Accept tokens as HTTP headers only
Aanbevolen	Error handling	Use default error handling
Aanbevolen	Error handling	Use the required HTTP status codes

* Wat betreft TLS zijn de Digikoppeling beveiligingsvoorschriften leidend (zie *Digikoppeling Beveiligingsstandaarden en voorschriften*).
** Alleen verplicht indien er sprake is van logging in systemen die niet onder controle van de betrokken client- en serverorganisatie staan.

4.4.6.1 Toelichting aanduidingen

Voorschriften zijn aangeduid met 'Verplicht', 'Aanbevolen' en 'Niet van Toepassing' waarvoor de volgende definities gelden:

Categorie	Codering RFC2119	Voorschrift	Toelichting
Verplicht	MUST	De eisen moeten gevolgd worden. Hier kan niet van afgeweken worden.	
Aanbevolen	SHOULD	Aanbevolen is om de eisen conform conform voorschrift te implementeren. Wanneer hier van afgeweken wordt dient een zorgvuldige afweging plaats te vinden	
Niet van Toepassing	-	De eisen zijn niet van toepassing	

4.5 Signing & Encryptie (in HTTP REST Context)

4.5.1 Signing

Signing van HTTP body en/of header kan gebruikt worden voor *authenticatie*, om de *integriteit* van de request/response berichten te controleren en signing realiseert ook *onweerlegbaarheid*. (Onweerlegbaarheid in de zin van: de verzender van de request/response kan niet ontkennen het bericht verzonden te hebben wanneer deze voorzien is van de digitale handtekening van de afzender).

De berichten kunnen ook samen met de digitale handtekeningen worden bewaard zodat deze bij audits of juridische bewijsvoering gebruikt kunnen worden.

Een HTTP requestbericht is opgebouwd uit de volgende onderdelen:

- Header
  - HTTP operatie (GET, POST etc)
  - Pad / URL resource
  - Protocol
  - Header velden
- Body
  - *data*

Door naast de body data ook onderdelen uit de header digitaal te ondertekenen kan worden gecontroleerd dat bv ook de HTTP operatie en resource specificatie in de request echt van de afzender afkomstig zijn en niet onderweg gemanipuleerd.

Indien signing van HTTP body/header wordt toegepast is het Verplicht om dit te doen volgens de regels van de ADR Module ADR-HTTP Message and payload signing with JAdES

4.5.2 Encryptie

Encryptie van HTTP request/response kan gebruikt worden om de vertrouwelijkheid van gegevens te beschermen.

Indien encryptie van HTTP request/response wordt toegepast is het Verplicht om dit te doen volgens de regels van de ADR Module ADR-HTTP Payload encryption

A. HTTP Message Signing & Encryptie in het Digikoppeling REST API profiel

A.1 Inleiding

Door een HTTP-Request/Response te voorzien van een digitale handtekening kan identiteit van de afzender worden gecontroleerd en het HTTP verkeer beschermd worden tegen manipulatie van buitenaf, ook kan met de digitale handtekening onweerlegbaarheid worden gerealiseerd in de zin dat de ontvanger niet kan ontkennen het bericht te hebben ontvangen en de verzender niet kan ontkennen het bericht te hebben verzonden.

Bij HTTP verkeer wordt voor authenticatie, integriteits bescherming (en bescherming van de vertrouwelijkheid) TLS (Transport Level Security) breed toegepast. In Digikoppeling wordt

specifiek 2-zijdig TLS toegepast. Echter TLS beschermt alleen via één TLS-verbinding en het pad tussen de client en de server kan bestaan uit meerdere onafhankelijke TLS-verbindingen (bijvoorbeeld als de applicatie wordt gehost achter een TLS-beëindigende gateway of als de client zich achter een TLS Inspection-apparaat bevindt). In dergelijke gevallen kan TLS geen end-to-end berichtintegriteit of authenticiteit tussen de client en de server garanderen.

HTTP Message signing kan dit geval gebruikt worden om end-to-end bescherming tegen manipulatie te realiseren voor het volledige pad Client naar Server.

HTTP Payload encryptie kan dit geval gebruikt worden om end-to-end vertrouwelijkheid te realiseren voor het volledige pad Client naar Server.

Figuur 3 Bescherming voor het volledige pad Client naar Server

A.2 Wanneer is HTTP Message Signing/Encryptie te gebruiken

Wanneer Client en Server middels 2-zijdig TLS verbonden zijn over 1 connectie (bijvoorbeeld direct of als de gateway d.m.v. 'pass-through' verkeer doorgeeft aan de Server) dan is er al sprake van end-to-end bescherming van het verkeer en is voor bescherming HTTP Message Signing niet nodig. (Voor vastlegging van audit logs of om rekening te houden met toekomstige infrastuktuur wijzigingen die end-to-end 2-zijdig TLS onderbreken zou signing ook in dit geval bruikbaar kunnen zijn).

Wanneer de 2-zijdig TLS connectie bij de Gateway wordt beindigd en de gateway het verkeer ontsleuteld en doorstuurd naar de server is er geen end-to-end beveiliging via TLS. In dit geval kan met behulp van HTTP Message Signing wel end-to-end bescherming geboden worden. De Client kan HTTP header informatie zoals bv de HTTP Operatie (GET/POST/DELETE/etc) en het resource pad samen met de body van het bericht voorzien van een handtekening, De Gateway kan deze ondertekening doorsturen naar de Server en de Server kan dmv de handtekening afzender en integriteit van het bericht controleren en vaststellen.

> NB Of HTTP Signing en Encryptie nodig/wenselijk zijn is afhankelijk van de opzet van de infrastruktuur en het gewenste beschermingsniveau. Signing , en ook Encryptie toepassen zorgt voor meer comlexiteit in de uitwisselingen. Ook zijn Signing en Encryptie relatief zware operaties die ook beslag op resources opleveren dus het is belangrijk om dit alleen toe te passen in use cases waar HTTP Signing en Encryptie toegevoegde waarde hebben

§ B. Referenties

§ B.1 Normatieve referenties

[ADR]
API Design Rules. Jasper Roes; Joost Farla. Logius. URL: https://gitdocumentatie.logius.nl/publicatie/api/adr/2.0

[DK-beveiliging]
Digikoppeling Beveiligingsstandaarden en voorschriften. Logius. URL: https://gitdocumentatie.logius.nl/publicatie/dk/beveilig/

[DK-IDAuth]
Digikoppeling Identificatie en Authenticatie. Logius. URL: https://gitdocumentatie.logius.nl/publicatie/dk/idauth/

[FSC-Core]
FSC - Core. Eelco Hotting; Ronald Koster; Henk van Maanen; Niels Dequeker; Edward van Gelderen; Pim Gaemers. Logius. URL: https://gitdocumentatie.logius.nl/publicatie/fsc/core/1.0.0/

[FSC-Logging]
FSC - Logging. Eelco Hotting; Ronald Koster; Henk van Maanen; Niels Dequeker; Edward van Gelderen; Pim Gaemers. Logius. URL: https://gitdocumentatie.logius.nl/publicatie/fsc/logging/1.0.0/

[PKIO-PvE]
Certificate Policy/Programme of Requirements PKIoverheid. Logius. URL: https://por.pkioverheid.nl/

↑