Logius Standaard - Vastgestelde versie

Digikoppeling Koppelvlakstandaard REST-API 2.0.2

Logius Standaard
Vastgestelde versie 30 januari 2025

Deze versie:
https://gitdocumentatie.logius.nl/publicatie/dk/restapi/2.0.2/

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/1.1.1/

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 (Nederlandse API Strategie IIa)* 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

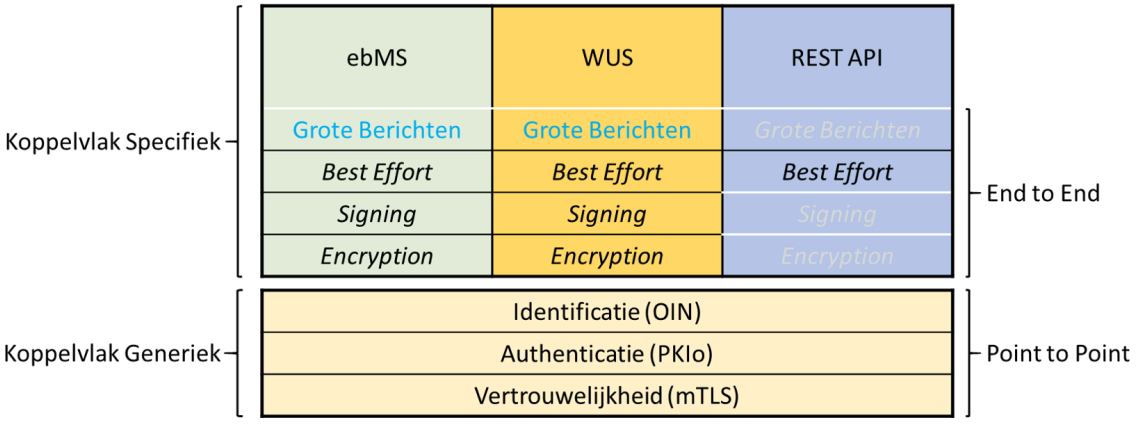
§ 1. Conformiteit

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

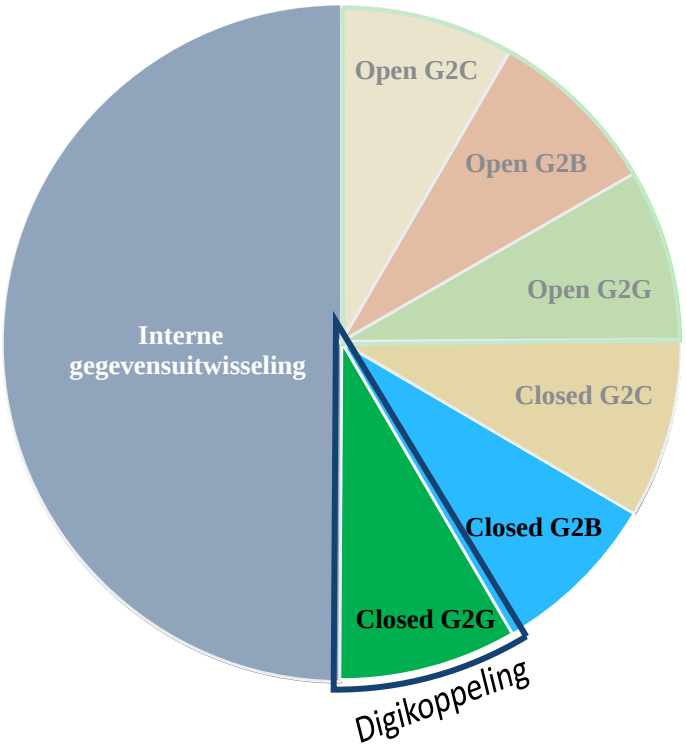
Het Digikoppeling Rest API profiel is gericht op Machine-to-Machine (M2M) en Government-to-Government (G2G) interacties conform de algemene uitgangspunten van de Digikoppeling standaard en het toepassingsgebied van Digikoppeling op de Pas-toe-of-leg-uit lijst (PTLU) van het Forum Standaardisatie;

Opzet Digikoppeling:



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		2W-be-S	osb-be-s
best-effort signed/encrypted		2W-be-SE	osb-be-e

Invulling Digikoppeling	DK REST API profiel	DK WUS profiel	DK ebMS2 profiel
reliable			osb-rm
reliable signed			osb-rm-s
reliable signed en encrypted			osb-rm-e

In versie 1.0 van het Digikoppeling REST API profiel wordt signing en encryptie niet ondersteund. In toekomstige versies van het profiel zal hier wel invulling aan worden gegeven. (Zie ook 5. BIJLAGE Gebruik van Signing & Encryptie in de context van HTTP / Rest API)

4. Digikoppeling REST API profiel

4.1 Inleiding

4.1.1 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.

4.1.2 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.

§ 4.2 Digikoppeling REST API profiel

§ 4.2.1 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.

§ 4.2.2 Koppelvlak Generiek

§ *4.2.2.1 Vertrouwelijkheid*

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

- Zie *Digikoppeling Beveiligingsstandaarden en voorschriften*

§ *4.2.2.2 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*

§ **4.2.3 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 extensie *Logging*. Core beschrijft het koppelen met API's en Logging hoe logregels weggeschreven moeten worden.

- *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.

§ *4.2.3.1 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.

§ *4.2.3.2 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*

§ *4.2.3.3 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*

§ *4.2.3.4 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*

§ *4.2.3.5 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.

§ *4.2.3.6 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.

§ *4.2.3.7 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.

§ **4.2.4 API Design Rules**

§ *4.2.4.1 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	

(Indeling gebaseerd op [*rfc2119*])

§ *4.2.4.2 Regels*

Het Digikoppeling REST API profiel conformeert zich volledig aan het normatieve deel van de [*ADR*].

Categorie	Principe	Toelichting	Link
Verplicht	REST-API Design Rules	Het is verplicht te voldoen aan alle (normatieve) eisen van de REST-API Design Rules	[*ADR*].

In onderstaande tabel worden de normatieve eisen van de [*ADR*] weergegeven:

Normatieve eisen van de REST API Design Rules

Categorie	Principe	Toelichting	Link
Verplicht	3.1 API-01: Adhere to HTTP safety and idempotency semantics for operations		API-01: Adhere to HTTP safety and idempotency semantics for operations
Verplicht	3.3 API-02: Do not maintain state information at the server		API-02: Do not maintain session state on the server
Verplicht	3.2 API-03: Only apply default HTTP operations		API-03: Only apply standard HTTP methods
Verplicht	3.1 API-04: Define interfaces in Dutch unless there is an official English glossary available		API-04: Define interfaces in Dutch unless there is an official English glossary available
Verplicht	3.5 API-05: Use nouns to indicate resources		API-05: Use nouns to name resources
Verplicht	3.4 API-06: Use nested resources for child resources		API-06: Use nested URIs for child resources
Verplicht	3.5 API-10: Model resource operations as a sub-reresource or dedicated resource		API-10: Model resource operations as a sub-resource or dedicated resource
Verplicht	3.6 API-16: Use OpenAPI Specification for documentation		API-16: Use OpenAPI Specification for documentation
Verplicht	3.6 API-17: Publish documentation in Dutch unless there is existing documentation in English		API-17: Publish documentation in Dutch unless there is existing documentation in English
Verplicht	3.7 API-18: Include a deprecation schedule when publishing API changes		API-18: Include a deprecation schedule when publishing API changes
Verplicht	3.7 API-19: Schedule a fixed transition period for a new major API version		API-19: Schedule a fixed transition period for a new major API version
Verplicht	3.7 API-20: Include the major version number in the URI		API-20: Include the major version number in the URI
Verplicht	3.1 API-48: Leave off trailing slashes from URIs		API-48: Leave off trailing slashes from URIs

Categorie	Principe	Toelichting	Link
Verplicht	3.6 API-51: Publish OAS at a standard location in JSON-format		API-51: Publish OAS document at a standard location in JSON-format
Verplicht	3.1 API-53: Hide irrelevant implementation details		API-53: Hide irrelevant implementation details
Verplicht	3.1 API-54: Use plural nouns to name collection resources		API-54: Use plural nouns to name collection resources
Verplicht	3.7 API-55: Publish a changelog for API changes between versions		API-55: Publish a changelog for API changes between versions
Verplicht	3.7 API-56: Adhere to the Semantic Versioning model when releasing API changes		API-56: Adhere to the Semantic Versioning model when releasing API changes
Verplicht	3.7 API-57: Return the full version number in a response header		API-57: Return the full version number in a response header

§ 4.3 Afspraken API Design Rules extensies

De ADR extensie onderderdelen van dit profiel zijn gebaseerd op: [*ADR-ext*].

Hieronder wordt aangegeven welke regels uit de API Design Rules extensies in dit profiel verplicht zijn of worden aanbevolen.

Categorie	Principe	Extensie	Toelichting	Link
Niet van toepassing	API-11: Secure connections using TLS	Security	Vervangen door Digikoppeling beveiligingsvoorschriften (*)	*Digikoppeling Beveiligingsstandaarden en voorschriften*
Verplicht	API-58 No sensitive information in URIs	Security	Alleen verplicht indien er sprake is van logging in systemen die niet onder controle van de betrokken client- en serverorganisatie staan	API-58 No sensitive information in URIs
Verplicht	API-13: Accept tokens as	Security Authorisation		API-13: Accept tokens as HTTP headers only

Categorie	Principe	Extensie	Toelichting	Link
	HTTP headers only			
Aanbevolen	API-46: Use default error handling	Error handling		API-46: Use default error handling
Aanbevolen	API-47: Use the required HTTP status codes	Error handling		API-47: Use the required HTTP status codes

(*) Wat betreft TLS zijn de Digikoppeling beveiligingsvoorschriften leidend , Zie *Digikoppeling Beveiligingsstandaarden en voorschriften*

5. BIJLAGE Gebruik van Signing & Encryptie in de context van HTTP / Rest API

> NOOT
> Deze bijlage is informatief en geen normatief onderdeel van het profiel

5.1 Signing in de context van HTTP Rest

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.

Enkele voorbeelden van signing standaarden die in ontwikkeling zijn:

- https://tools.ietf.org/html/draft-ietf-httpbis-message-signatures
- https://www.openbankingeurope.eu/media/2095/obe-json-web-signature-profile-for-open-banking.pdf

§ 5.2 Encryptie in de context van HTTP Rest

Voor encryptie is de standaard JSON Web Encryption (JWE) [*rfc7516*] beschikbaar

Zie ook de ADR extensie signing en encryptie:

- https://docs.geostandaarden.nl/api/API-Strategie-ext/#signing-and-encryption

§ A. Referenties

§ A.1 Normatieve referenties

[ADR]
API Design Rules (Nederlandse API Strategie IIa). Jasper Roes; Joost Farla. Logius. Juli 2020. URL: https://gitdocumentatie.logius.nl/publicatie/api/adr/

[ADR-ext]
API Designrules Extensions (Nederlandse API Strategie IIb). Jasper Roes; Linda van den Brink. Geonovum/Kennisplatform APi's. Januari 2020. URL: https://docs.geostandaarden.nl/api/API-Strategie-ext

[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/

[rfc7516]
JSON Web Encryption (JWE). M. Jones; J. Hildebrand. IETF. May 2015. Proposed Standard. URL: https://www.rfc-editor.org/rfc/rfc7516

§ A.2 Informatieve 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

↑