The Global Language of Business

Links Data IN API Specification

Technical specification to manage links using GRP APIs

Release 3.2.5, Approved, Sep 2025

Document Summary

Document Item	Current Value
Document Name	Links Data IN API Specification
Document Date	Sep 2025
Document Version	3.2.5
Document Issue	
Document Status	Approved
Document Description	Technical specification to manage links using GRP APIs

Contributors

Name	Organisation
Birgit Mahler	GS1 Global Office – Director Product Management
Erik Major	GS1 Global Office – Senior Product Owner
Nick Lansley	GS1 Global Office – Lead Developer and Innovation
Rajat Rastogi	GS1 Global Office - Quality Assurance Engineer
Tazeen Zia	GS1 Global Office – Technical Writer

Log of Changes

Release	Date of Change	Changed By	Summary of Change
1.0	May 2023	Contributors	Release alongside Links functionality release to STAGING on the GS1 Registry Platform (GRP)
2.0	May 2023	Contributors	Release alongside Links functionality release to PRODUCTION on the GS1 Registry Platform (GRP)
3.1 (Elevated document version in line with API V3.1)	August 2023	Contributors	Document release alongside API V3.1 release to STAGING. Document version updated to reflect new API version.
3.2	January 2025	Contributors	All sections updated to reflect changes for version 3.2 API
3.2.1	March 2025	Contributors	Document updated alongside functionality release to GRP STAGE environment
3.2.2	June 2025	Contributors	Updated section 3.3 Delete Links with details for “single link deletion” alongside functionality release to STAGING.
3.2.3	July 2025	Contributors	Percent encoding in Check Links (3.4), Template variables(2.3.1), serial component variables (2.3.2)
3.2.4	August 2025	Contributors	Added section 5 “GO Resolver Prefix Redirect API”. Included Technical validation rules (6) and Error codes (7) corresponding to this API.
3.2.5	September 2025	Contributors	Document release alongside GRP API v3.2 release to PRODUCTION

Disclaimer

GS1 seeks to minimise barriers to the adoption of its standards and guidelines by making the intellectual property required to implement them available, to the greatest extent possible, on a royalty-free basis, or when necessary, under a RAND licence. Such royalty-free and RAND licences are provided pursuant to the GS1 IP Policy (available here: https://www.gs1.org/standards/ip), which governs the work of work group participants who contribute to drafting standards and guidelines, including this document. In addition to licences, the GS1 IP Policy provides various benefits and obligations that apply to all implementers of GS1 standards and guidelines, and all implementations of GS1 standards are subject to those terms.

Nevertheless, please note the possibility that an implementation of one or more features of this standard or guideline may be the subject of a patent or other intellectual property right that is not covered by the licences granted pursuant to the IP Policy. In addition, the licences granted under the IP Policy do not include the IP rights or claims of third parties who were not participants in the corresponding standard development work group.

Accordingly, GS1 recommends that any person or organisation developing an implementation of this standard or guideline should determine whether any patents or other intellectual property may encompass such implementation, and whether a licence under a patent or other IP right is needed. The implementer should determine the potential need for licensing in view of the details of the specific implementation being designed in consultation with that party's patent counsel.

The official versions of all GS1 standards and guidelines are provided as PDF files on GS1's online reference directory (https://ref.gs1.org) (the "GS1 Reference"). Any other representations of standards or guidelines in any other format (e.g., web pages) are provided for convenience and descriptive purposes only, and in the event of a conflict, the GS1 Reference document shall govern.

THIS DOCUMENT IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR PARTICULAR PURPOSE, ACCURACY OR COMPLETENESS, OR ANY WARRANTY OTHERWISE ARISING OUT OF THIS DOCUMENT. GS1 disclaims all liability for any damages arising from any use or misuse of this document, whether special, indirect, consequential, or compensatory damages, and including liability for infringement of any intellectual property rights, relating to use of information in or reliance upon this document.

GS1 makes no commitment to update the information contained herein, and retains the right to make changes to this document at any time, without notice. GS1® and the GS1 logo are registered trademarks of GS1 AISBL.

Table of Contents

1 Introduction

Links to other sources of data is a valuable addition to the attributes in the GS1 Registry Platform. Using a URL, links can be registered for GS1 Primary Keys (GTIN, GLN, SSCC, GRAI, and other keys), allowing brand owners to augment the thin number of attributes in the registries with more and richer information about their products, brands, locations and other items. This will help to build trust in GS1 keys used in the supply chain by providing more information to verify 'valid' GS1 Keys, while also providing important information which enables a wide variety of use cases, such as product certifications, sustainability and safety information, etc.

The Verified by GS1 Data OUT API pulls data from all relevant registries (Licence information, identification key data and links from the Links Registry) for allowing users to verify a GS1 key. Link data helps in building trust as it provides access to sources with further information about that product - like a product webpage with nutritional data -, traceability information or "how to use" instructions.

To see an example, try querying GTIN 9506000134352 via the Verifed by GS1 service on gs1.org "Verify Product" search and look at the "Links to product information" provided.

This API specification describes how URLs can be registered, managed, and accessed in the Links Registry within the GS1 Registry Platform.

Important: The GO (Global Office) Resolver is another Data OUT service that uses data from the Links Registry. The GO Resolver functions separately from Verified by GS1 and your MOs **do not need** to offer a Resolver solution for registering Links in the Links Registry and for using them via Verified by GS1.

1.1 Conditions for participation

Member Organisations can use the Links API on an as-needed basis, on the condition that they have existing rights to use other APIs, e.g. for GTIN and GLN. For more information, please contact the GS1 Helpdesk (helpdesk@gs1.org).

1.2 Target audience

This document is intended for all MO software development teams who need to understand the capabilities of the GS1 Registry Platform and APIs, and how these capabilities may integrate into the local MO systems. This documentation covers the APIs used to manage links in the Links Registry and is intended **only to be used by GS1 Member Organisation and GS1 Global Office colleagues**. It's important that this technical documentation MUST NOT be shared with members, industry partners, or other parties.

Furthermore, this document is intended as additional information to the API specifications and other notes available in the GS1 Developer Portal.

Note: Please also see GRP APIs best practice recommendations on the GS1 MO Zone. These recommendations are provided to help increase system reliability and performance, improve data propagation across the ecosystem and provide cost-savings for the GS1 Federation.

1.3 Migration to API version 3.2

Starting in 2025, the GRP platform is providing a new API, version 3.2 (v3.2), which includes significant improvements to the Links APIs. These improvements result in breaking changes.

The registry technical team has designed the new v3.2 Links API solution to be more closely aligned with RFC 9264 (https://datatracker.ietf.org/doc/rfc9264/) and the GS1 Resolver Conformant standard (https://ref.gs1.org/standards/resolver/), both in regards to terminology as well as schema guidance and attribute rules. A thorough read-through and understanding of these documents will help understand the Links re-engineering efforts, as well as how this data will be used by Resolver.

The following summary of changes shows important comparisons of the new API v3.2 as compared to v3.1. The information can be used by technical teams, which have implemented services using v3.1 APIs, to focus on the changes required to use the v3.2 APIs.

1.3.1 Application Identifiers

The Links v3.2 APIs shall only accept Application Identifiers (AI) on Data IN, to represent the primary key and qualifiers. Previously in v3.1, a GLN link, for example, could be uploaded to the Links Registry as type "GLN". (This acronym form is sometimes referred to as a "short code" or "convenience alphas".) However, GRP supports various GLN types, with the supported list of GLN AIs including 414, 415, and 417.

Entry of primary key and qualifier information in the v3.2 Links APIs follows the Digital Link standard and requires the use of AIs (see notes below in Attributes). The Digital Link standard provides a convenient and widely accepted format for expressing information about primary keys and qualifiers. For these reasons, the GRP Links API will only accept Application Identifiers to specify primary keys and qualifiers.

The full list of AIs and associated qualifiers that is supported in v3.2 has been expanded to be in line with standards. See the full list in the following section 2.1.1 anchorRelative and anchor.

Note: For the purposes of this API document, the combination of primary key and qualifiers (if qualifiers are present) may be referred to as the link "**scope**". While other related documentation may refer to this concept as the link "context", this can easily be confused with the Links API attribute called "context" (see section 2.1.3.3). For example, suppose links are added to the Link Registry for GTIN 09506000134352, and links are added for the same GTIN 09506000134352 with some qualifiers for lot and batch number. This will result in two *scopes* of target links. The scope at the GTIN-only level is "less granular" than the scope at the GTIN-plus-qualifiers level.

1.3.2 Attributes

The v3.2 API schema provides alignment of attributes by name and format on both data IN and data OUT, and some attributes are now optional which were previously mandatory. The table below lays out the changes. In the first column "V3.1 attribute name", there exist instances where two names for the v3.1 attributes are provided (e.g. responses / linkset); the first name was the v3.1 Data IN name, and the second was v3.1 Data OUT name.

V3.1 attribute name	V3.2 attribute name	V3.2 optional or mandatory	Notes
identificationKeyType identificationKey qualifierPath	"anchorRelative" on Data IN, and "anchor" on Data OUT	Mandatory	The anchor is based on the Digital Link standard and represents the primary key and type, and any qualifiers (if present). In v3.2, on Data IN, the anchor is expressed as "anchorRelative" without the domain. The domain (for GO Resolver) is added on Data OUT and expressed as "anchor". The application identifiers included in the anchor define the scope for the associated set of links.
itemDescription	description	Optional	
responses / linkset	"links" on Data IN, and "linkset" on Data OUT	Mandatory	An array containing a set of one or more links, as well as a defaultLink.
linkType	@linkType	Mandatory	V3.2 includes the '@' character

V3.1 attribute name	V3.2 attribute name	V3.2 optional or mandatory	Notes
language / hreflang	hreflang	Optional	In v3.2, hreflang is an array of strings on both Data IN and Data OUT. This is a change from v3.1, where the attribute was an array only on Data OUT.
context	context	Optional	In v3.2, context is an array of strings on both Data IN and Data OUT. This is a change from v3.1, where the attribute was an array only on Data OUT.
mimeType / type	type	Optional	
linkTitle / title	title	Mandatory	
targetUrl / href	href	Mandatory	
defaultLinkType	REMOVED – see note	--	Rather than using various attribute for default values, the v3.2 API will use the defaultLink link type to specify explicitly which link is the default.
defaultLanguage			
defaultContext			
defaultMimeType			
fwqs	REMOVED – see note	--	For v3.2, it was determined that the forward query string (fwqs) is not an option that is needed for any functionality.
public	public	Optional	While the attribute is OPTIONAL on Data IN, it is in fact stored in the registry for each link. In v3.2, if the attribute is not provided on Data IN, the system shall assume public = true. The public attribute is provided only at the link level in v3.2. Previously in v3.1, a public attribute could also be set at the links level; this is no longer required in v3.2.

More details on the use of these attributes in v3.2 can be found later in this document.

1.3.3 Creating links with same linkType

A restriction of the previous API version 3.1 was that more than one link for the same primary key and qualifier combination was not allowed when the link is using the same `context`, `language` and `type` (media or mime type), for the key/qualifier 'scope' (See note in section 1.3.1) . V3.2 of the Links API allows the presence of target URLs within the same scope which use the same linktype and other mentioned attributes.

1.3.4 Updated Link Type list; Healthcare restriction

The latest link type code list is loaded into GRP for use with the Links Registry. See the appendix A.2.2 linkType for information on restricted healthcare link types.

1.3.5 Defaults

Rather than using the variety of default attributes (see table in section 1.3.2 Attributes), the link type "`defaultLink`" is now used to explicitly set the default link for `links`. This method is much clearer than the ambiguous process that was used with the variety of default attributes, which are

now removed with v3.2. A `defaultLink` must always be declared for `links` for each scope, even if there is only one link for a particular scope.

1.3.6 Better alignment with GRP platform

The previous Links APIs under version v3.1 did not make full use of the advantages of the GRP platform and there existed various inconsistencies with other GRP APIs (e.g. for GLN, GTIN, and Licence). With the v3.2 APIs, the following advantages are introduced to better align the Links APIs within GRP.

- Properly formatted validation rules (adhering to the E0xx format)
- Status response codes consistent with codes used across all APIs in GRP
- Use of the GUID batch ID format, as well as the common batch status endpoint (rather than a separate Links batch status endpoint)
- Batches support up to 1000 sets of links, with improved performance
- Data IN links payloads overwriting existing links records for the same primary key and qualifiers or 'scope'. This means that when updating link information, GRP always stores the latest data that is uploaded; always upload the full set of links for the primary key and qualifier.
- Some back-end advantages which will not be apparent to the client user of the APIs, such as common platform logging mechanisms

1.3.7 Link Deletion enhancements

In v3.2, the Delete Links API will change in several key ways:

- Determination of which link(s) to delete will be based on `anchorRelative`. This provides more granularity for deletion as compared to v3.1, as you can specify the levels of granularity from primary key to primary key + qualifier(s).
- The API will support batches up to 1000 anchorRelative, and no longer support the single key deletion by providing .. /{identificationKeyType}/{identificationKey} in the API URI
- The key type must be specified as an Application Identifier, e.g. '01' for GTIN
- Single link deletion: The ability to delete a single `href` link (with its associated attributes) from an array of links under an `anchorRelative`. This allows services to quickly delete single links without modifying data for other links within that set.

1.3.8 Impacts to Verified by GS1

With the changes to the attributes and schemas, the Verified by GS1 payload will reflect these same changes. This means that the links information in Verified by GS1 responses will be updated in version 3.2 APIs. This impacts the following endpoints:

- Get Verified GTINs
- Get GLNs by Key
- Get Verified Licensee by Key

1.3.9 Serial component variables

Some primary keys contain a serial component within the key itself, rather than using a qualifier to represent a serial item. The v3.2 Links API provides a way to set variables for these keys so that a target link can utilize the non-serial portion of the key , the serial component, or the entire identification (key+serial). The following primary keys support serial component variables in the v3.2 links API: GRAI, GDTI, GCN, GIAI.

2 Links Registry attribute guidance

The contents of this section are based on the v3.2 APIs. For more details on the requirements for attributes, refer to the Technical Validation rules section of this document.

2.1 Links Entry/Root attributes

The root part of `links` payload consists of the following attributes, which are elaborated in subsequent sections below:

- anchorRelative (Data IN) and anchor (Data OUT)
- description
- links (Data IN) and linkset (Data OUT)

2.1.1 anchorRelative and anchor

`anchorRelative` is a mandatory attribute on Data IN, and is also used to reflect sets of links in the Check Links by Key payload. The same information is returned on Data OUT as `anchor`.

- The anchor information is based on the Digital Link standard and represents the primary key and type, and any qualifiers (if present). Essentially, the anchor represents the scope for a target link or set of target links related to the key and qualifiers.
- In v3.2, on Data IN the anchor is expressed as "`anchorRelative`" without the domain. The domain (for GO Resolver) is added on Data OUT (id.gs1.org for Prod, and resolver-st.gs1.org for stage) and expressed as "`anchor`".

  For example, if there exists a link for GTIN 09506000134352;

  - on Data IN, the information shall be expressed as:

    `"anchorRelative": "01/09506000134352"`

  - while on Data OUT, it shall be returned with the GO Resolver domain (as shown below from GRP Production):

    `"anchor": "https://id.gs1.org/01/09506000134352"`

    In this way, Member Organisations can use the APIs without including, or knowing, the domains for Global Office Resolver which also uses the data from the Links Registry.
- The `anchorRelative` will be validated based on the Digital Link standard, which dictates the format and order of qualifiers. Refer to https://www.gs1.org/standards/gs1-digital-link.
  - **Important exception**: While the Digital Link standard states that a URI must begin with a slash, for purposes of the Links Registry API the anchorRelative attribute value must begin with a numeric digit and must not have a slash (/) in front.

Note: Linksets with qualifiers are only returned via GO Resolver. While linksets with a scope of only a primary key (no qualifiers) will be returned by both GO Resolver and the Verified by GS1 service.

The following table shows a list of Application Identifiers (primary keys and qualifiers) which the GRP Links Registry supports. For more information on Application Identifiers, please see https://ref.gs1.org/ai/.

Acronym	Primary Key	Application Identifier	Key Qualifiers
SSCC	Serial Shipping Container Code	00	
GTIN	Global Trade Item Number	01	10 - batch/lot; 21 - serial number; 22 - Consumer Product Variant (CPV)

Acronym	Primary Key	Application Identifier	Key Qualifiers
GDTI	Global Document Type Identifier	253	
GCN	Global Coupon Number	255	
GINC	Global Identification Number for Consignment	401	
GSIN	Global Shipment Identification Number	402	
GLN	Identification of a physical location - Global Location Number	414	254 - GLN extension component
	Global Location Number of the invoicing party	415	8020 - Payment slip reference number
	Party Global Location Number	417	
GRAI	Global Returnable Asset Identifier	8003	
GIAI	Global Individual Asset Identifier	8004	
ITIP	Identification of an individual trade item piece	8006	10 - batch/lot; 21 - serial number; 22 - Consumer Product Variant (CPV)
GMN	Global Model Number	8013	
GSRN	Global Service Relation Number to identify the relationship between an organisation offering services and the provider of services	8017	8019 - Service Relation Instance Number
	Global Service Relation Number to identify the relationship between an organisation offering services and the recipient of services	8018	8019 - Service Relation Instance Number

2.1.2 description

`description` is an optional attribute which can be used to describe the scope of links for the product, location, or asset the link is associated with. The description shall be meaningful to the links and is chosen at the discretion of the brand owner working with the MO. For example, the description may provide text describing the specific scope of the links (based on primary key and qualifiers, or other content of the link targets).

2.1.3 links (Data IN) and linkset (Data OUT)

`links` is an array consisting of the following lower level links attributes which describe the set of target links for the specified scope. One or more of these elements may exist for "`links`" array:

- @linkType
- hreflang
- context
- type
- title
- href
- public

The following subsections provide details about the attributes within this array.

2.1.3.1 @linkType

`@linkType` is a mandatory attribute and is validated against a linkTypes code list from the GS1 Web Vocabulary.

To specify a `linkType`, you may use values of a curie (compact URL) or a complete URL. For example, for `linkType` 'Activity Ideas' the value for:

- curie is 'gs1:activityIdeas'
- URL is 'https://ref.gs1.org/voc/activityIdeas'

Note: It is recommended to use **curie** values for linkType attribute.

Note: While using the endpoint:" Check Links" (section 3.4) the linkType will always be in form of curie (even if it was specified as a URL on Data IN).

2.1.3.2 hreflang

`hreflang` is an optional string array attribute and specifies the language of content the `href` or targetUrl links to. An optional locale or region code can be provided. The `hreflang` attribute must comply with the applicable ISO language code lists (for details please see section A.2.1 language).

Examples when only language code is specified: "en", "ja", "deu" (for English, Japanese & German) etc.

Examples when language code and optional locale/region code is specified: "deu-DE", "en-US" etc.

2.1.3.3 context

`context` is an optional string array attribute and defines the target markets or countries of sale for which the link's resources are relevant, or any other information that the brand owner feels may be relevant to the scope of the link. For example, a product can have a product information page in Spanish language including some specific information which makes it relevant only for the Costa Rican market (hence Costa Rica would be the `context` here).

Examples: "gb", "us" (for Great Britain & United States) etc.

Note: The context attribute should not be confused with the idea of a links "context" as discussed elsewhere in GS1 standards, where "context" is the scope of data as defined by the primary key and any relevant qualifiers, if present. The presence and number of qualifiers impact the granularity of the data. For this document, we may refer to this concept as the "scope" associated with links (also see Note in section 1.3.1 Application Identifiers).

2.1.3.4 type

`type` refers to MIME (Multipurpose Internet Mail Extensions) type, Media type or Content Type (after the HTTP header that indicates the Media type), and is an optional two-part string identifier that indicates a data format as a pair of type and subtype. The `type` attribute value is optional and is validated against a code list (Media Type code list). The `type` allows you to provide links to the same resource in different formats such as an excel spreadsheet, JSON, XML, PDF or Web page. This attribute is case sensitive.

Examples: image/jpeg, image/gif, image/png, text/html, text/rtf or application/json etc.

2.1.3.5 title

Link `title` is a mandatory attribute and provides a human friendly description of the `href` target URL.

Examples: "Product Info", "Information for patients" etc.

2.1.3.6 href

The target URL within the `href` attribute is mandatory and specifies the URL for the information associated with the linktype and scope. The target URL must not be a Digital Link URL using the domain of GO Resolver, as this will lead to an endless loop when the data is queried via GO Resolver. The string value must start with an accepted scheme from the following list:

Value	Description
'http://'	unencrypted web address
'https://'	encrypted web address
'ftp://'	file transfer protocol (file download)
'sftp://'	secure file transfer protocol (file download)
'rtsp://'	media streaming protocol
'sip:'	internet telephone number
'tel:'	standard telephone number
'did:'	distributed identifier (verifiable credential)
'mailto:'	create an email to send

Example: https://dalgiardino.com/risotto-rice-with-mushrooms

2.1.3.7 public

`public` is an optional Boolean attribute, specified by values of 'true' or 'false'. The value determines if an `href` URL link requires user authentication to be displayed/shared in GS1 Member Organisation or Global Office services. If the link cannot be shared outside of the GS1 Federation and our partners, then it shall be set as 'false'. The expectation is that most if not all links will be public = true.

By default, `links` shall be registered as `public`: 'true', and these links can be used in all connected Data OUT services (e.g., Verifed by GS1 service on gs1.org or the GO Resolver service). As the `public` attribute is optional, if the attribute is not provided on Data IN, the registry will automatically default the value to `public` = true.

If your MO wishes to share the link(s) only via GS1 services for which users must authenticate themselves, they must register that link as restricted, by specifying `public` as 'false'.

2.2 Defaults

A default link must be provided and identified within a set of links for a particular scope (primary key and qualifiers). This requirement to provide a default link is necessary whether there is a single link or multiple links in the set.

While declaration of a default link - when there is only one link - may appear redundant, it is required to support the functions of GO Resolver and makes an explicit and intentional selection of the default using the defaultLink link type.

2.2.1 Default link

`gs1:defaultLink` – is a required linktype that must be included in the Data IN payload. Only one `defaultLink` can be declared for a set of links within a specific scope per anchorRelative.

Along with `gs1:defaultLink`, there must be declared an `href` and a `title`. Data IN examples elsewhere in this document show examples of full payloads with the default link. Below is an example of a `defaultlink` that would be included in a Data IN payload.

```
{
        "href": "https://example.com/product/1234567890128/info",
        "title": "Example default link",
        "@linkType": "gs1:defaultLink"
}
```

The link types of `gs1:defaultLink` and `gs1:defaultLinkMulti` are used by the GO Resolver application when redirecting queries to default links. As such, the data for these link types is not used or disclosed in the Verified by GS1 results.

2.3 Variables

With "variables", anchors and links can be defined in the Links Registry even when the exact value for qualifiers or the serial portion of certain key types is not known, or the data must be applied to a range of potential values.

Variable use for qualifiers is called "Template Variables", while variable use for serial portions of certain keys (GRAI, GDTI, GCN, GIAI) is called "Serial Component Variables". For both Template and Serial Component Variables, the variable is defined as an alphanumeric case sensitive string, within curly brackets '{}'.

2.3.1 Template variables

Template variables can be defined in the anchorRelative and can only be used for qualifiers. A target URL (`href`) for a link under the anchorRelative can reference any variable defined in anchorRelative as long as the URI remains valid, however it is not required to use or reference any variables just because they are defined in the anchorRelative. The variables can be used when the actual value (e.g. a serial number) is not known, or values are known and there is a range of qualifier values, or you simply want to provide a link that may require a value (a serial number) regardless of the actual value.

On Data IN,

- a template variable may be defined and declared in the anchorRelative using curly braces {}.
- template variables are valid for qualifier values, e.g. 21/{serial}, within the anchorRelative.
- template variables cannot be used for primary keys; they can only be used for the qualifiers associated with the primary key.
- a target URL `href` may contain a template variable anywhere in the URI, except in the domain, and as long as it is still a valid URI. For example, http://{cpv}.com is NOT valid because the variable cannot be used in the domain.
- the target URL can contain multiple occurrences of a variable name, e.g. https://example.com/myProductInfo/{serialNum}/?serial ={serialNum}
- any template variable found in a valid href URI MUST be one that was defined in the anchorRelative or it will be rejected.
- Variable names are case sensitive; for example, {serialNum}, {SERIALNUM} and {sErIaLnUm} are all different variable names.

Consider the following example:

```
[{
```



```
        "anchorRelative": "01/09507000008865/21/{serialNumber}",
        "description": "Example with template variable",
        "links": [
         {
             "href": "https://example.com/myProductInfo/{serialNumber}",
             "title": "Product information with serial number",
            "@linkType": "gs1:pip",
             "type": "text/html"
         },
         {
             "href": " https://example.com/myProductInfo/",
             "title": "Product info, general information",
             "@linkType": "gs1:defaultLink"
         }]
}]
```


- With this data, a request to GO Resolver for: https://id.gs1.org/01/09507000008865/21/abc123?linktype=gs1:pip would redirect to https://example.com/myProductInfo/abc123.
- As another example, if the PIP target URL was structured as: https://example.com/myProductInfo/?serial={serialNumber}, the redirect would go to: https://example.com/myProductInfo/?serial=abc123

2.3.2 Serial component variables

Some primary keys contain a serial component within the key itself, rather than using a qualifier to represent a serial item. The v3.2 Links API provides a way to set a variable in the target anchor which will use data from primary keys which may contain serial components. The following keys will support serial component variables in the v3.2 links API: GRAI, GDTI, GCN, GIAI [see Important Note below regarding GIAI].

To use Serial Component Variables, the `anchorRelative` must be for one of the keys listed above, and the following variable names can be used in the target URL. Note the change to meaningful names in the version 3.2 schema:

- '{serial}'
  - previously referred to as '{1}' in v3.1
  - represents the serial component of the identification key
- '{key}'
  - This is a new variable in v3.2
  - represents only the non-serial portion of the identification key
- '{id}'
  - previously referred to as '{0}' in v3.1
  - represents the entire identification key, i.e. both the key and serial component

The following rules apply to Serial Component Variables:

- Only the reserved word variables listed above can be used
- The associated `anchorRelative` must NOT contain a serial component

An example will help illustrate how these variables are used by GO Resolver. Suppose there exists a GRAI (AI 8003) '09502037827427'. You want to provide an `href` to a specific link about serialized asset, but when creating the link data in the Links Registry you do not know what the serial number will be (or you want to use the same link for a range of serial numbers). In this case, the `anchorRelative` in the links registry would be:8003/09502037827427 [Note: this is only the 14 digit GRAI; no serial component is included]. And the target link under this anchor may be something like: https://example.com/GRAI/{id}/{key}/?serialNumber={serial}.

The query will come into GO Resolver as a Digital Link with the serial number included, for example:

https://id.gs1.org/8003/09502037827427abc

Resolver will find the anchor in the Links Registry using this key, take the key and serial portions, and replace the variables in the target `href`, resulting in a redirect to:

https://example.com/GRAI/09502037827427abc/09502037827427/?serialNumber=abc

Important: GIAI has a variable length separate from the serial component which may be contained in the key, which adds a layer of complication in how GIAI is handled. This feature for handling GIAI with serial component variables will be documented in more detail in a future version of this API document.

3 Links Data IN APIs

The contents of this section are based on the v3.2 APIs.

These endpoints allow MOs to manage Links for data IN operations.

Base URL: https://grp.gs1.org/grp-st (for STAGING environment)
https://grp.gs1.org/grp (for PRODUCTION environment)

Authentication: API key

Request Headers:

Name	Data Type	Required?	Description
Content - Type	String	Optional	The media type of the body is sent to the API. If provided, it must be application/json
API key	String	Required	Subscription key which provides access to this API.

3.1 Create or modify Links

Create new and/or modify existing links in the GRP if they are associated with GS1 primary keys (GTIN, GLN, GIAI etc.) derived from licences assigned to your MO. Like other API endpoints in GRP, the payload submitted, after successful validation, will overwrite any previous data that existed for the specific anchorRelative. The associated licence must already exist in the GRP and the licence status must be "active". For GTINs, the management of links falls under the role of the Primary MO (as per the multinational solution). Management of licences is handled via other services and APIs connected to the Licence Registry.

Initial result is the `batchId`, and then the ID is queried via `batchStatus` to see validation errors.

Note: If a Link is associated with a GTIN or GLN, it is NOT required to have a record for this Key in the GTIN or GLN Registry. GRP does not store identification keys for other primary keys beyond GTIN and GLN, and links for other primary keys can be uploaded to the registry as long as the associated licence exists and the licence is ACTIVE.

3.1.1 Request – Create or modify Links

- Endpoint name: Create or Modify Links
- Endpoint: /v3.2/links
- Method: POST
- Request Body:

  With each links Data IN API call, the following attributes are **mandatory or optional**:

Attribute Name	Type	Optional / Mandatory	Notes
anchorRelative	String	Mandatory	Uses the Digital Link format, without the domain, and without a leading slash. Specifies the scope of the link, i.e. the primary keys and qualifiers (if present) for the set of links.
description	String	Optional	A relevant description of the link scope for the product, location or asset, or other information which helps to describe the set of links
links	Array	Mandatory	Must not be NULL or empty. The array of target links contains one or more of the following attributes in this table.
@linkType	String	Mandatory	Values taken from Link Types code list.

Attribute Name	Type	Optional / Mandatory	Notes
hreflang	Array of strings	Optional	The language associated with the information at the href target link.
context	Array of strings	Optional	Information about the context of the identification key, target country for the links, or other relevant data.
type	String	Optional	Values taken from Media Type code list
title	String	Mandatory	Human friendly description of the target URL.
href	String	Mandatory	Must start with an accepted scheme (2.1.3.6).
public	Boolean	Optional	Valid values: true/false. If the attribute is not provided, the link will be saved with public = true.

- Request Example:


```
[{
  "anchorRelative": "01/09507000008865",
  "description": "test",
  "links": [
      {
          "href": "https://example.com/01/123",
          "title": "test",
          "@linkType": "gs1:pip",
          "type": "text/html",
         "public": false
      },
      {
          "href": " https://example.com/01/123",
          "title": "test",
          "@linkType": "gs1:allergenInfo",
          "type": "text/html"
      },
      {
          "href": "test",
          "title": "test",
          "@linkType": "https://ref.gs1.org/voc/careersInfo",
          "hreflang": ["sd"],
          "public": true

      },
      {
          "href": " https://example.com/activities",
          "title": "Activities",
          "@linkType": "gs1:activityIdeas",
          "context": ["test"]
      },
      {
          "href": " https://example.com/activities",
          "title": "Activities - this is the default link",
          "@linkType": "gs1:defaultLink"
      }
```



```
    ]
}]
```

3.1.2 Response – Create or modify Links

The following example provides an overview of the response for this endpoint. Please refer to Technical validation rules, Error codes and Status codes for more information on API responses and error messages.

The response will include a batch ID which will be used to check the status of the batch.

- OK
  - Accepted: 202

    Payload is accepted and is being processed. Querying of the BatchID in the "Query batch status and response" endpoint will provide further details on the processing of the data.

    - BatchID response example in application /json:

      ```
      "29759da7-cea1-4601-a2fa-9eb889249c5e"
      ```

- Error
  - Bad Request: 400
    - Response message: A validation error string with error code, for example:

      ```
      {
      "errorCode": "E021",
      "message": "Batch size of \"2342\" is too large; max allowed is \"1000\""
      }
      ```

  - Unauthorised: 401
  - Forbidden: 403

3.1.3 Links upload – important notes about Licences

Certain rules regarding Licences must be followed to register links (associated with GS1 primary keys) on behalf of your MO and brand owner.

- The Licence which any given primary key is based on must already exist in the GRP Licence Registry.
- The Licence must be assigned to this MO as the Licensing MO. In the case of GTINs, if the licence is managed by the multinational solution, the MO must be set as the licence's Primary MO.
- The licence must be "active" in the Licence Registry. If an MO has already set the status of a licence to "inactive", Links can no longer be registered or modified within the Links Registry.
  - Even though target URLs and link attributes cannot be edited for links in the registry where the associated licence is inactive, the brand owner could still, of course, change the content the Links' target URL points to at their own discretion (for example, changing the content of a brand website).

  More information about the rules can be found in Section 5 Technical Validation Rules.

3.2 Query batch status and response

When creating, updating, or deleting links, the response will be a Batch ID. The Batch ID will need to be queried with the feedback API endpoint to check the batch status and to see results of the API call. Additionally, a feedback Batch status Response will only be returned if the batch belongs to the requesting MO.

3.2.1 Request – Query batch status and response

- Endpoint name: Query batch status and response
- Endpoint: /v3.2/feedback/{batchId}
- Method: GET

  Request body:

Name	Data Type	Optional / Mandatory	Description
batchID	GUID	Mandatory	Batch ID returned from batch Data In operations

- Request Example:

  GET https://grp.gs1.org/grp/v3.2/feedback/29759da7-cea1-4601-a2fa-9eb889249c5e

3.2.2 Response – Query batch status and response

The following example provides an overview of the response for this endpoint. Please refer to Technical validation rules, Error codes and Status codes for more information on API responses and error messages.

- Success
  - Success code: 200 OK
  - Response example in JSON


```
[
    {
        "code": 1,
        "anchorRelative": "414/9507000008865/"
    },
    {
        "code": 2,
        "anchorRelative": "00/295070000088652312/"
    },
    {
        "validationErrors": [
            {
                "property": "anchorRelative",
                "errors": [
                    {
                        "errorCode": "E003",
                        "message": "\"01/09507000008865/10/\" is not in expected Digital Link format"
                    }
                ]
            },
            {
                "property": "qualifiers.value",
```

```
                    "index": 0,
                    "errors": [
                        {
                            "errorCode": "E010",
                            "message": "Value is required and must not be blank"
                        }
                    ]
                }
            ],
            "code": 5,
            "anchorRelative": "01/09507000008865/10/"
        }
]
```


3.3 Delete Links

The Delete Links endpoint allows deletion of existing links in the GRP if they are associated to GS1 Keys derived from licences assigned to your MO. Links can be deleted in the following manner:

- A full **deletion of all the links associated with a specific context** (primary key and qualifiers, if present), by simply providing the anchorRelative for the context.

  For example, suppose a GTIN `09506000134383` has two `anchorRelative` values as:

  1. `"anchorRelative": "01/09506000134383"`
  2. `"anchorRelative": "01/09506000134383/21/abc123"`

  When submitting only the 1st `anchorRelative` `"01/09506000134383"` for deletion, all links under that context will be deleted; links for the 2nd anchor relative will remain.

- Ability to **delete one single link from a set of links for a specific scope**, by providing the `anchorRelative` along with the attributes for the specific link to be deleted. Multiple single links can be deleted at one time per delete request, with a "links" section of the payload under the `anchorRelative`.

  - Links within a set of links do not have an ID as a unique single parameter to identify the link, therefore all the attributes for the link must be provided so that the system can identify the link for deletion, except for the `public` attribute which may be included but is not required when providing attributes for a single link deletion. Note that the order of the attributes does not matter; the order submitted for deletion does not have to match with the order returned in Check Links or Data OUT endpoints.

    See examples below in the Delete Request and Response sections of this document.

  - Use the Check Links API endpoint to get list of `anchorRelative` attributes for a given key. You can copy the payload of check links to help in providing the attributes for single link deletion.

  - You cannot delete a single link which has link type "`defaultLink`", as a defaultLink is always required in the set of links for an `anchorRelative`. Use the Create or modify Links endpoint to modify the contents of a defaultLink. The `defaultLink` will be deleted by the system when a deletion of the entire anchorRelative is submitted, or when the single link being deleted is the last non-defaultLink in the set of links for the anchorRelative.

3.3.1 Request – Delete Links

- Endpoint name: Delete Links
- Endpoint: /v3.2/links
- Method: DELETE
- Request body: A batch array of anchorRelative values, to specify the set of target links to be deleted.
- Sample Payload:


```
[
    {
        "anchorRelative": "8006/095070000088650121"
    },
    {
        "anchorRelative": "8013/950700000886211BK"
    },
    {
        "anchorRelative": "01/09506000134383",
        "links": [
            {   "href": "https://www.example.com/dpplink",
                "title": "DPP link example",
                "type": "application/pdf",
                "hreflang": ["en"],
                "context": ["dpp", "pdf file", "English"],
                "public": true,
                "@linkType": "https://ref.gs1.org/voc/dpp"
            }
                ]
    }
]
```


3.3.2 Response – Delete Links

- OK
  - Accepted: 202

    Payload is accepted and is being processed. Querying of the BatchID in the "Query batch status and response" endpoint will provide further details on the processing of the data.

    - BatchID response example in application /json:


```
"29759da7-cea1-4601-a2fa-9eb889249c5e"
```


    Response will give a Code 4 for a successful deletion, along with the anchorRelative and attributes (in the case of single link deletion), for example:


```
[{
        "code": 4,
        "anchorRelative": "8006/095070000088650121"
```



```
        },
        {
            "code": 4,
            "anchorRelative": "8013/950700000886211BK"
        },
        {
            "code": 4,
            "@anchorRelative": "01/09506000134383",
            "@linkType": https://ref.gs1.org/voc/dpp,
            "href": "https://www.example.com/dpplink",
            "title": "DPP link example",
            "type": "application/pdf",
            "hreflang": ["en"],
            "context": ["dpp", "pdf file", "English"],
            "public": true
        }
    ]
```


Response code status 5 will be given for unsuccessful deletions.

3.4 Check Links

The Check Links endpoint allows you to get all links for a particular primary key. The purpose of this endpoint is that MOs can see all data for links, to check the contents of links data in the registry.

Important: This endpoint is for MO-INTERNAL USE ONLY and it enables MOs to check links that have been registered in the GRP. Furthermore, it provides information at all levels of granularity regardless of if the data is utilized by Verified by GS1 or the Global Office Resolver, and regardless of if the links are set as public TRUE or restricted.

3.4.1 Request – Check Links

- Endpoint name: Check Links
- Endpoint: /v3.2/links/{identificationKeyType}/{identificationKey}
- Method: GET
- Request Body: This API simply includes the identification key type and the key for the key you wish to query. This request will return any level of granularity associated with this key.
- Request example:

  GET https://grp.gs1.org/grp/v3.2/links/01/09506000134352

Important notes:

- Identification keys queried via this endpoint must use percent-encoding if a special character exists in the key. This is sometimes also called URL encoding. (To understand more about percent-encoding and see a list of all the encoding codes, please see Percent-encoding).
- Some keys, such as GMN, allow use of special characters in the key. Suppose there exists a GMN key using the special character percent '%', as: **95089121%SG**

- When querying this key via Check Links, replace the special character '%' with the associated percent-encoding, which is '%25'. Therefore, the key will be queried as: **95089121%25SG**

3.4.2 Response – Check Links

You can see all the Links information which you specified while creating it (Section 3.1.1).

Response examples:

- If no links for the key are found:
  - HTTP Response code 404 "Not Found", with message "Could not find any link contexts for {anchorRelative}"
- If the queried key parameter is not properly formatted:
  - HTTP Response code 400 "Bad Request", with validation errors, for example:


```
[
    {
        "errorCode": "E003",
        "message": "GTIN \"09506793\" has invalid structure or format"
    },
    {
        "errorCode": "E001",
        "message": "Length of 8 is not allowed.  Length must be
        between 12 and 14, inclusive"
    },
    {
        "errorCode": "E002",
        "message": "Check digit for key \"09506793\" is incorrect"
    }
]
```


- If links are found:


```
[{
        "anchorRelative": "01/09506000134352",
        "description": "An example with a couple links",
        "links": [
            {
                "href": "https://example.com/productinfo",
                "title": "test",
                "@linkType": "gs1:pip",
                "type": "text/html",
                "public": false
            },
            {
                "href": " https://example.com/activities",
                "title": "Activities",
                "@linkType": "gs1:activityIdeas",
                "context": ["test"],
                "public": true
            },
            {
```

```
                    "href": " https://example.com/activities",
                    "title": "Activities - this is the default link",
                    "@linkType": "gs1:defaultLink",
                    "public": true
                }
            ]
        },
        {
             "anchorRelative": "01/09506000134352/21/abc123",
             "description": "An example with a couple links",
             "links": [
                 {
                    "href": "https://example.com/productinfoForSerial",
                    "title": "test",
                    "@linkType": "gs1:pip",
                    "type": "text/html",
                   "public": true
                 },
                 {
                    "href": "https://example.com/productinfoForSerial",
                    "title": "PIP - this is the default link",
                    "@linkType": "gs1:defaultLink",
                    "public": true
                 }
            ]
        }
    ]
```


4 Links Data OUT APIs

The contents of this section are based on the v3.2 APIs.

Links registered in the Links Registry will be used for Data OUT services. This includes **Verified by GS1**(https://www.gs1.org/services/verified-by-gs1) and the **GO Resolver** at id.gs1.org.

For Verified by GS1, links are included in the payload of the following APIs:

- GTIN (GET Verified GTINs API)
- GLN (GET GLN by key API)
- Other GS1 Keys (GET Verified Licensee by Key API)

  Member Organisations use these APIs for connecting their local Verified services to the GRP. Global Office will use these APIs for the Verified by GS1 service on gs1.org and the global Verified component.

  See the respective API documentation for these endpoints for more details about how links are provided in Verified by GS1 payload.

  If the scope of Links is only at the primary key level, they may be returned in the Verified by GS1 service and the Global Office Resolver. However, Links at a more granular scope are exclusively returned to GO Resolver. For example, suppose we have two sets of links based on the following two scopes (shown as attribute "anchorRelative")

  A) `"anchorRelative": "01/09506000134352"`

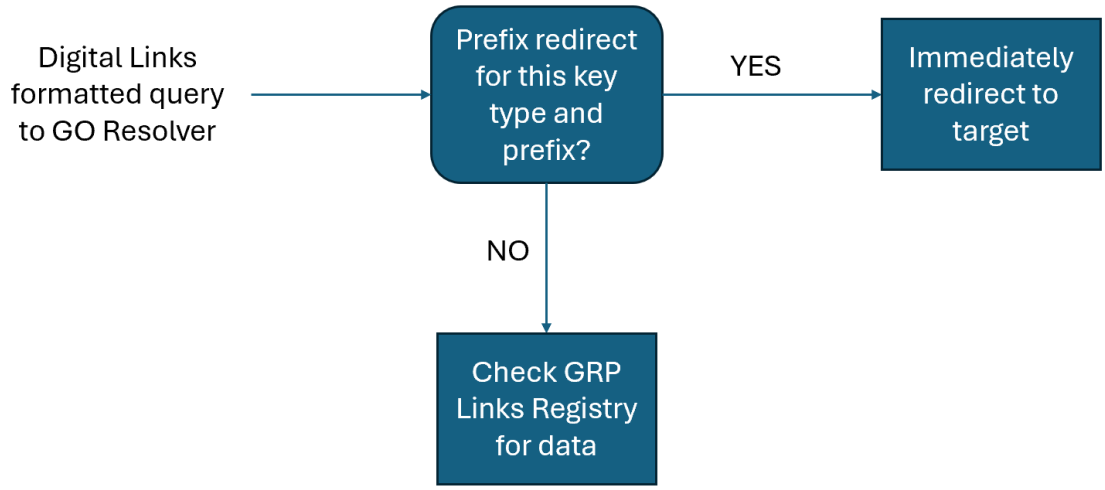
  and

  B) `"anchorRelative": "01/09506000134352/21/ABC123"`

  (A) is only at the primary key level (GTIN 09506000134352), which is less granular than **(B)** which is at the scope of primary key (GTIN 09506000134352) plus a qualifier (serial number ABC123). GO Resolver may return both **(A)** and **(B)**, while Verified by GS1 may only return **(A)**.

5 GO Resolver Prefix Redirect API

The data in the Links Registry is used by both Verified by GS1 services and GO Resolver. However, there will be cases where the brand owner and the Member Organisation know that relevant link data is contained somewhere else in the world, e.g. on a brand managed Resolver or some 3rd-party host. Rather than looking for data in the GRP Links Registry, the Prefix Redirect allows setting of configurations so that queries for keys beginning with a certain licence prefix will be redirected to some other location when the query comes into the GO Resolver, rather than looking in the GRP Links Registry.



The Digital Link URI portion and any query parameters found in the original query will be passed through to the redirected target domain.

For example, suppose the Prefix Redirect has the following configuration:

- AI: 01
- Prefix: 95060001787
- Target Redirect URI: https://example.com/

Then, if the following query comes into GO Resolver:

https://id.gs1.org/01/09506000178714

GO Resolver will see that this GTIN begins with the prefix in the configuration and will immediately redirect the query, along with the Digital Link URI, to:

https://example.com/01/09506000178714

The target redirect URI may have other path components beyond the root domain, and the redirect will function in a similar manner. For example, suppose the Prefix Redirect has the following configuration:

- AI: 01
- Prefix: 95060001787
- Target Redirect URI: https://example.com/resolver3/

Then, if the following query comes into GO Resolver:

https://id.gs1.org/01/09506000178714

GO Resolver will see that this GTIN begins with the prefix in the configuration and will immediately redirect the query, along with the Digital Link URI, to:

https://example.com/resolver3/01/09506000178714

An API is provided so that MOs can set the prefix redirect as they need to, without needing to work through the GS1 Helpdesk (which was the old method for setting these parameters).

The following attributes are used to configure the redirect:

Attribute Name	Type	Optional / Mandatory	Notes
ai	Numerals in a string	optional	The specific AI which this redirect shall be applied to. The AI must be from the list of supported primary keys. If an AI is not provided, then this redirect will be applied to all keys.
prefix	Integer in a string	Mandatory	A number representing the prefix/licence to be redirected to the redirectUri. Length between 3 and 12 characters.
redirectUri	String	Mandatory for create/update; not required when deleting	A URL with a valid scheme, no more than 2048 characters

Notes:

- Prefixes can be configured in batches
- After submitting a batch, a batch ID will be generated which is used in the Batch Status endpoint to check for status and any validation errors
- If a configuration is set for two or more prefixes which have the same initial digits and therefore some of the prefixes have additional digits (longer prefixes), when GO Resolver receives a query it will first check for a match against the longer prefixes before the shorter prefixes.

  For example, the following is configured:

  - **AI**: 01, **Prefix**: 950123, **redirectUri**: https://id.myfavoritebrand.com/
  - **AI**: 01, **Prefix**: 9501234, **redirectUri**: https://id.anotherbrand.com/

  Then:

  - if a query comes into GO Resolver for 01/0**9501234**567891 → redirect to https://id.myfavoritebrand.com
  - if a query comes into GO Resolver for 01/0**950123**9999994 → redirect to https://id.anotherbrand.com
- As the `ai` parameter is optional, it is possible that you set a configuration for the same prefix where one configuration has a specific AI, while the other has no AI specified. When GO Resolver receives a request for that prefix, it will first look at the configuration with the specific AI. If the Digital Link syntax AI parameter does not match this redirect AI, then the system will use the redirect for the prefix without the AI parameter.

  For example, the following is configured:

  - **Prefix**: 9501234, **redirectUri**: https://id.brand.com/
  - **AI**: 01, **Prefix**: 9501234, **redirectUri**: https://id.brandtwo.com/

  Then:

  - if a query comes into GO Resolver for **00**/3**9501234**0013000003 → redirect to https://id.brand.com
  - if a query comes into GO Resolver for **01**/0**9501234**567891 → redirect to https://id.brandtwo.com
- Management of prefix redirects is based on the 3-digit prefix assigned to your MO. Global Office appreciates to hear insights or use cases which could be used to improve the management of prefix redirect configurations. Please reach out to the GS1 Helpdesk helpdesk@gs1.org.

5.1 Request – Create Prefix Redirect Configuration

- Endpoint name: Create Redirect Prefixes

- Endpoint: /v3.2/links/redirect/prefixes
- Method: POST
- Request example with sample payload:

  POST https://grp.gs1.org/grp/v3.2/links/redirect/prefixes

```
[
  {
        "ai": "00",
        "prefix": "950123",
        "redirectUri": "https://id.logisticsbrand.com/"
  },
  {
        "ai": "01",
        "prefix": "950123",
        "redirectUri": "https://id.gtinresolver.com/"
  },
  {
        "prefix": "9502",
        "redirectUri": "https://id.myfavoriteresolver.com/"
  }
]
```

If all of these configurations are successfully created, when querying the batch ID status you will see the following:

```
[
  {
  "code": 1,
  "ai": "00",
  "prefix": "950123"
  },
  {
  "code": 1,
  "ai": "1",
  "prefix": "950123"
  },
  {
  "code": 1,
  "ai": null,
  "prefix": "95012"
  }
```

```
]
```

5.2 Request – Delete Prefix Redirect Configuration

- Endpoint name: Delete Redirect Prefixes
- Endpoint: /v3.2/links/redirect/prefixes
- Method: DELETE
- Request example with sample payload:

  DELETE https://grp.gs1.org/grp/v3.2/links/redirect/prefixes

```
[
  {
        "ai":"00"
        "prefix":"950123",
  },
  {
        "ai": "01",
        "prefix": "950123",
  },
  {
        "prefix": "9502",
  }
]
```

5.3 Checking prefix redirect configuration

A method will be provided to view the prefix configuration at any given time. This method is under development. In the meantime, the best way to verify the redirect is working is to send a relevant query to GO Resolver and confirm the query is redirected to the expected target.

6 Technical validation rules

A validation rule is used to verify that the data a user enters in a record is valid and can be written to the registry. A validation rule contains a formula and expression that evaluates the data in one or more fields and returns an error code if the rule fails. There will be variations for the displayed error code message text, depending on the attributes which are validated.

These validation rules are strictly enforced for data submitted to the GRP Links Registry platform.

ID	Rule	Context / attribute	Possible Attribute Error Codes
Links-1	An ACTIVE licence must be present in the licence registry for the primary key for which a link is being added.	Primary key derived from `anchorRelative`	E010, E040
Links-2	The licence must be assigned to the calling MO as the Licensing MO, or the Primary MO in the case of GTINs (01).	Primary key derived from `anchorRelative`	E012
Links-3	The payload for creating links must not be blank.		E021
Links-4	anchorRelative is required and must be between 9 to 2048 characters in length	`anchorRelative`	E001, E003, E010
Links-5	anchorRelative must be in Digital Link format and use only the supported list of Application Identifiers. NOTE an exception to this rule: the anchorRelative URI must not start with a slash '/'.	`anchorRelative`	E001, E003, E007, E010, E011
Links-6	Primary keys must be expressed as numeric application identifiers. Acronyms, e.g. "GTIN" are not allowed.	`anchorRelative`	E003, E007
Links-7	The primary key value used in the anchorRelative must be in the proper key format	`anchorRelative`	E001, E002, E003, E010, E041
Links-8	The qualifiers for a primary key must only be from the supported list of qualifiers.	`anchorRelative`	E003, E011, E019
Links-9	The qualifiers for a primary key must be in the correct order and without duplications, as per the Digital Link standard	`anchorRelative`	E002, E010, E017, E041
Links-10	The licence for the primary key must exist in the Licence registry, must be under the control of the submitting MO, and the licence must be ACTIVE.	`anchorRelative`	E010, E012, E040
Links-11	When including description, it must contain 1 to 500 characters.	`description`	E001
Links-12	The links array must be present and not be blank	`links`	E010
Links-13	Exact duplicates of elements in the links array are not allowed.	`links`	E017
Links-14	Within the link array, each element must contain one href. href must be between 1 to 2048 characters.	`links.href`	E001, E010
Links-15	Within the link array, an href string cannot be used more than once; each target link href must be unique within the link context scope.	`links.href`	E017
Links-16	Within the link array, each element must contain one title. Title must be between 1 to 500 characters.	`links.title`	E001, E010
Links-17	Within the link array, each element must contain one @linkType, and must be a valid support link type. Note that healthcare specific link types are not supported.	`links.@linkType`	E001, E010, E011
Links-18	@linkType must be expressed as a curie (compact URI) "gs1:[linktype]", or as a full URL "https://ref.gs1.org/voc/[linktype]"	`links.@linkType`	E011

ID	Rule	Context / attribute	Possible Attribute Error Codes
Links-19	Media/MIME type shall be from the supported list of types and is case sensitive.	`links.type`	E011
Links-20	If the context attribute is included, it must contain at least one string in the array.	`links.context`	E010
Links-21	If the context attribute is included, the array must not contain duplicate string values.	`links.context`	E017
Links-22	If the hreflang attribute is included, the values must be from the supported list of languages	`links.hreflang`	E009, E010
Links-23	If the hreflang attribute is included, the array must not contain duplicate language string values.	`links.hreflang`	E017
Links-24	If the description attribute is included, it must not be blank.	`links.description`	E001
Links-25	When included, the public attribute must be either 'true' or 'false'. Note that if the public attribute is not included, the system will save the link as public = true by default.	`links.public`	Result will be a 400 bad request if Boolean value is not provided
Links-26	Within each link array for a link context, there must exist one and only one link which is the defaultLink	`links.@linkType.defaultLink`	E017, E042
Links-27	For each defaultLink, only the following fields are required: href, title, @linkType. This rule also applies when creating links of type defaultLinkMulti.	`links.@linkType.defaultLink`	E001, E010
Links-28	To avoid a looping condition if this target URL is queried via GO Resolver, the href shall not be a Digital Link URI using the GO Resolver domain for Staging or Production	`links.href`	E026
Links-29	AI used for prefix redirection must be numeric and from the list of supported AIs	`ai`	E007, E003
Links-30	Prefix is a required field for redirect configuration	`prefix`	E010
Links-31	Prefix must be between 3 and 12 digits	`prefix`	E001
Links-32	The 3 digit GS1 Company Prefix at the start of prefix must be assigned to your MO; the prefix cannot be reserved, restricted, or assigned to another MO. These rights are based on GCP assignment and do not consider things such as the multinational solution, GBOL, or prefix delegation.	`prefix`	E004, E005, E006, E012
Links-33	RedirectUri is a required field for redirect configuration	`redirectUri`	E010
Links-34	RedirectUri string cannot be longer than 2048 characters	`redirectUri`	E001
Links-35	RedirectUri must have a valid protocol scheme and be a valid absolute URI	`redirectUri`	E003
Links-36	RedirectUri must not contain a fragment ('#') or query parameters ('?')	`redirectUri`	E003

7 Error codes

Error codes are represented as an integer. All error codes correspond to a failed validation rule and will be returned with a variable error message specific to the given use case and/or attribute. Error messages may also return information pertinent to rectifying the data, such as constraints of the validation rule.

Error codes relevant to Links Registry are listed in the table below.

Error code	Description
E001	Invalid length of value for attribute
E002	Invalid check digit or check characters
E003	*Incorrect format.* For example, Contains invalid characters; value out of range; unsupported pattern; invalid structure or format; value is not in Digital Link format; qualifier value has incorrect format, length, or AI; URL contains variables not defined in the anchorRelative; target link and redirectURI must be a valid absolute URI; RedirectUri does not allow fragments or query parameters
E004	Prefix does not exist or is not assigned
E005	Prefix is reserved for special use
E006	Prefix is assigned for use by another MO
E007	Unsupported Application Identifier
E009	Language or locale code is invalid or not supported
E010	Value is required and must not be blank; licence for key does not exist; target link was not found and could not be deleted
E011	Qualifier AI is invalid or not supported; link type is invalid, not currently supported, or restricted; invalid MIME type
E012	Licence belongs or is assigned to another MO; redirect configuration not found
E017	*Duplicate values.* For example, Duplicate qualifiers; duplicate target links; duplicate values in array ; duplicate variable names in anchorRelative
E019	Qualifier AI is not supported for the primary key
E021	Payload must be included and contain at least one element
E026	Target URL should not point to the anchor via resolver
E040	Licence is inactive; licence status must be active
E041	Invalid order of qualifiers; invalid item counts/out of sequence item counts for ITIP key
E042	Link context scope must contain 1 and only 1 gs1:defaultLink; link with type defaultLink cannot be deleted explicitly via single link deletion.

A Appendices

A.1 Appendix 1 – Status Codes

Status codes are integer codes that indicate the status of a Data IN operation. Since all Data IN operations are performed asynchronously in a batch, these codes will not be returned in the original response but will be returned in the body of any Data IN feedback response.

Every batch response will have a code present for each individual record indicating the status of that record.

Code	Meaning	Description
1	CREATED	Record created successfully
2	MODIFIED	Record modified successfully
4	DELETED	Record deleted successfully
5	FAILED	The operation failed due to validation errors or other reasons
7	PENDING	The batch operation has been accepted and is pending processing

A.2 Appendix 2 – Code Lists

A.2.1 language

Language tags consist of a mandatory language code and optional region code. The region code (if present) needs to be preceded by a dash (in the form xx-YY, where xx is a valid language code and YY is a valid region code).

The region is an optional part of the language tag; language tags without a defined region should be provided in form xx by omitting the region portion (including the -) entirely.

Both Alpha-2 and Alpha-3 language codes are supported in GRP as well as the one-off value zh-Hans/zh-Hant, which is for Simplified and Traditional Chinese. This means xx, xxx, xx-YY, xxx-YY are all supported. Additionally, zh-Hans, zh-Hant and zh-Han[s|t]-YY are also supported.

- Supported language codes are all 2-character codes listed for the ISO 639-1:2002 standard (ISO 639 Part 1).

  For reference: https://www.iso.org/iso-639-language-codes.html

- Supported region codes are all country codes listed for the ISO 3166 Alpha-2 standard (extended regional codes are not supported within the language field).

  For reference: https://www.iso.org/iso-3166-country-codes.html

  Example: "nl" for Dutch or "nl-BE" for Dutch in the region Belgium.

A.2.2 linkType

The "@linkType" attribute must be from the GS1 Link Type list. A full list of the Link Types can be found here: https://www.gs1.org/voc/?show=linktypes

Some link types are **restricted use in GRP**. As GRP is not a Healthcare Regulated System, the link types that are intended as exclusive use for Healthcare are not allowed in GRP. Therefore, when attempting to add a link type with one of the following, a validation error will occur, and the associated link will not be written to GRP:

- ***gs1:eifu***
  - Title: Electronic Instructions For Use
  - Description: A link to electronic Instruction For Use instructions for Medical Devices.
- ***gs1:epil***

- Title: Electronic Patient Information Leaflet
- Description: A link to an electronic patient information leaflet.

- ***gs1:smpc***
  - Title: Summary Product Characteristics
  - Description: A link to Summary Product Characteristics. To be used specifically when linking to information for healthcare professionals.

At the time of this writing, the use of link type *gs1:handledBy* is also restricted. There are plans to remove this restriction to enable the ability to configure prefix redirections on the Global Office Resolver instance. Once that feature is implemented via GRP APIs (the "redirect service") this restriction will be lifted.