



The Global Language of Business

# GS1 SmartSearch Implementation Guideline

Create web pages that refer to trade items (formerly GTIN+  
on the Web)

*Release 1.0.1, Ratified, Nov 2015*

---

## 1 Document Summary

Document Item	Current Value
Document Name	GS1 SmartSearch Implementation Guideline
Document Date	Nov 2015
Document Version	1.0
Document Issue	1
Document Status	Ratified
Document Description	Create web pages that refer to trade items (formerly GTIN+ on the Web)

## 2 Contributors

Name	Organisation
Richard McKeating, working group co-chair	Tesco Stores
Mark Frey, working group facilitator	GS1 Global Office
Mark Harrison, document editor	Auto-ID Labs Cambridge
Ken Traub, document editor	Ken Traub Consulting LLC
Anarkat, Dipan	GS1 Global Office
Ausili, Andrea	GS1 Italy
Beideman, Robert	GS1 Community Room Staff
Benhaim, Marc	GS1 France
Bos, Frits	GS1 Netherlands
Bratt, Steve	GS1 GO
Burd, Randy	MultiAd Kwiikee
Cornelisse, Rutger	Business Technologies Ltd
Cox, Christie	Wal-Mart Stores, Inc.
Dean, Kevin	GS1 Canada
Farance, Frank	Farance Inc.
Feuerstein, Véra	Nestle
Fuchs, Klaus	Auto-Id Labs ETH Zurich
Fuessler, Andreas	GS1 Germany
Gerasimenko, Alexander	Mars, Inc.
Gomez Sepulveda, Juan Pablo	GS1 Mexico
Gormley, Alan	GS1 Ireland
Hakala, Pertti	GS1 Finland
Hogan, Bernie	GS1 US
Houlette, Cedric	GS1 France
Kauz, Eric	GS1 Global Office
Kravec, Nevena	GS1 Bosnia & Herzegovina
Kungl, Jens	METRO Group

Name	Organisation
Lam, Ken	GS1 Hong Kong
Lockhead, Sean	GS1 GO
Maatsubatahi, Roberto	GS1 Brasil
Muller, Dan	GS1 Switzerland
Ogandzhanov, Georgy	GS1 Russia
Olsen, James	MultiAd Kwiikee
Oney, Ilteris	GS1 GO
Osborne, Andrew	GS1 UK
Paixao, Silverio	GS1 Portugal
Peter, Bijoy	GS1 India
Ramos, Carlos	GS1 Mexico
Ramos Velazquez, Emmanuel	GS1 Mexico
Richardson, Rich	GS1 US
Rivarola, Sebastian	Eway S.A.
Robba, Steven	1WorldSync Holdings, Inc.
Rosell, Pere	GS1 Spain
Rutger, Maciej	GS1 Poland
Sahni, Vipin	GS1 India
Sehorz, Eugen	GS1 Austria
Sheldon, David	Nestle
Soboleva, Olga	GS1 Russia
Strouse, Owen	GS1 Global Office
Swaminathan, Sachidanantham	GS1 India
Tluchor, Tomas	GS1 Czech Republic
Troeger, Ralph	GS1 Germany
van den Bos, Frits	GS1 Netherlands
Vuckovac, Denis	Auto-Id Labs ETH Zurich
Walker, John	Semaku
Wiesmann, Matthias	Google Switzerland GmbH
Yu, Shi	Beijing REN JU ZHI HUI Technology Co. Ltd.
Zhang, Tony	FSE, Inc.
Zuniga, Sergio	GS1 Mexico
Zurn, Darryl	Smiths Medical

### 3 Log of Changes

Release	Date of Change	Changed By	Summary of Change
1.0	June 2015	Eric Kauz	Initial release
1.0.1	Nov 2015	David Buckley	GTIN+ on the web renamed to GS1 SmartSearch

## 4 Disclaimer

5 GS1<sup>®</sup>, under its IP Policy, seeks to avoid uncertainty regarding intellectual property claims by requiring the participants in  
6 the Work Group that developed this **GS1 SmartSearch Implementation Guideline** to agree to grant to GS1 members a  
7 royalty-free licence or a RAND licence to Necessary Claims, as that term is defined in the GS1 IP Policy. Furthermore,  
8 attention is drawn to the possibility that an implementation of one or more features of this Specification may be the subject  
9 of a patent or other intellectual property right that does not involve a Necessary Claim. Any such patent or other  
10 intellectual property right is not subject to the licencing obligations of GS1. Moreover, the agreement to grant licences  
11 provided under the GS1 IP Policy does not include IP rights and any claims of third parties who were not participants in the  
12 Work Group.

13 Accordingly, GS1 recommends that any organisation developing an implementation designed to be in conformance with this  
14 Specification should determine whether there are any patents that may encompass a specific implementation that the  
15 organisation is developing in compliance with the Specification and whether a licence under a patent or other intellectual  
16 property right is needed. Such a determination of a need for licencing should be made in view of the details of the specific  
17 system designed by the organisation in consultation with their own patent counsel.

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

23 GS1 retains the right to make changes to this document at any time, without notice. GS1 makes no warranty for the use of  
24 this document and assumes no responsibility for any errors which may appear in the document, nor does it make a  
25 commitment to update the information contained herein.

26 GS1 and the GS1 logo are registered trademarks of GS1 AISBL.

27

# Table of Contents

28

29	<b>1 Introduction .....</b>	<b>7</b>
30	1.1 Purpose of this Document .....	7
31	1.2 Who Will Use this Document? .....	8
32	1.3 Brief Introduction to Linked Data about products and offerings .....	8
33	1.4 HTTP URIs in Linked Data.....	10
34	<b>2 Business Motivation for Deploying Linked Data About Products .....</b>	<b>10</b>
35	2.1 Why should I introduce structured data on my web site?.....	10
36	2.2 Is this new technology? .....	10
37	2.3 Why is GS1 involved and why have they developed the GS1 Web Vocabulary?.....	11
38	2.4 I already use Schema.org to describe my products. Why should I use the GS1 Web Vocabulary?	
39	11	
40	2.5 What are the benefits for brand owners and manufacturers? .....	11
41	2.6 What are the benefits for retailers? .....	11
42	2.7 What are the benefits for consumers?.....	12
43	2.8 Who publishes the structured data?.....	12
44	2.9 Who can consume the structured data?.....	12
45	2.10 How can structured data for a product be accessed using a QR code?.....	12
46	2.11 How will information about retail product offerings be made available to consumers via search	
47	and apps?.....	12
48	2.12 How do I get started? .....	13
49	2.13 What training and education do GS1 Member Organisations need to develop? Who gives	
50	guidance?.....	13
51	2.14 What are the challenges going to be in terms of getting retailers and brands on board?.....	13
52	2.15 Why is it attractive to marketing / SEO agencies? .....	14
53	2.16 How will search engine listings be impacted by structured data? .....	14
54	2.17 How will the search engine surface products searched under 'red shoes' for example?.....	14
55	2.18 Will the semantic web become integral to search engine optimisation (SEO)? .....	15
56	2.19 How will search engines know who the trusted source of data is?.....	15
57	2.20 How and will sponsored search results, such as Google AdWords, be impacted?.....	15
58	2.21 How do we engage the web development world?.....	15
59	2.22 What are the implications for merchants' product data feeds to search engine marketplaces?.	16
60	2.23 Walk me through how search could be enhanced in future, based upon linked data .....	16
61	<b>3 Implementation Procedures .....</b>	<b>16</b>
62	3.1 Procedure for brand owners or manufacturers to construct an HTTP URI to identify a product in	
63	facts asserted using Linked Data .....	17
64	3.1.1 Pre-Requisite.....	17
65	3.1.2 When Would I Use This? .....	17
66	3.1.3 How To? .....	17
67	3.2 Procedure for retailers to construct an HTTP URI to identify a product offering in facts asserted	
68	using Linked Data.....	18
69	3.2.1 Pre-Requisite.....	18
70	3.2.2 When Would I Use This? .....	18
71	3.2.3 How To? .....	18
72	3.3 Procedure for brand owners or manufacturers to construct a simple block of JSON-LD to	
73	represent basic facts about any product, using the schema.org vocabulary.....	19
74	3.3.1 Pre-Requisite.....	19



75 3.3.2 When Would I Use This? ..... 19

76 3.3.3 How To? ..... 19

77 3.4 Procedure for retailers to construct a simple block of JSON-LD to represent basic facts about any

78 product offering, using the schema.org vocabulary ..... 23

79 3.4.1 Pre-Requisite..... 23

80 3.4.2 When Would I Use This? ..... 23

81 3.4.3 How To? ..... 23

82 3.5 Procedure for brand owners or manufacturers to construct a simple block of JSON-LD to

83 represent basic facts about any product, using the GS1 web vocabulary..... 26

84 3.5.1 Pre-Requisite..... 26

85 3.5.2 When Would I Use This? ..... 26

86 3.5.3 How To? ..... 26

87 3.6 Procedure for retailers to construct a simple block of JSON-LD to represent basic facts about any

88 product offering, using the GS1 web vocabulary ..... 29

89 3.6.1 Pre-Requisite..... 29

90 3.6.2 When Would I Use This? ..... 29

91 3.6.3 How To? ..... 29

92 3.7 Procedure for serving a block of JSON-LD via an existing web page, using embedding – one

93 product per page..... 32

94 3.7.1 Pre-Requisite..... 32

95 3.7.2 When Would I Use This? ..... 32

96 3.7.3 How To? ..... 32

97 3.8 Procedure for serving a block of JSON-LD via an existing web page, using embedding –

98 procedure for multiple products per page ..... 33

99 3.8.1 Pre-Requisite..... 33

100 3.8.2 When Would I Use This? ..... 33

101 3.8.3 How To? ..... 33

102 3.9 Procedure for serving a standalone block of JSON-LD in isolation via a webserver ..... 34

103 3.9.1 Pre-Requisite..... 34

104 3.9.2 When Would I Use This? ..... 34

105 3.9.3 How To? ..... 34

106 3.10 Procedure for checking that structured data is correctly formatted ..... 34

107 3.10.1 Pre-Requisite..... 34

108 3.10.2 When Would I Use This? ..... 34

109 3.10.3 How To? ..... 35

110 3.11 Procedure for accessing structured data in a JSON-LD block using JavaScript within the same

111 web page ..... 35

112 3.11.1 Pre-Requisite..... 35

113 3.11.2 When Would I Use This? ..... 35

114 3.11.3 How To? ..... 35

115 **A Appendix: Technical background for deploying Linked Data about products**

116 **36**

117

# 1 Introduction

## 1.1 Purpose of this Document

This document provides guidance about how machine-readable structured data about a product or product offering can be embedded within an existing web page. "Data about a product" refers to information that describes the product, such as its name, physical dimensions, ingredients, suggested use, and so on. "Structured data" refers to data that is not just free-form text, but rather is organised into individual units of data, often called "data elements" or "attributes," and that the same data elements are used in a consistent way to describe many different products. Structured data about products is often called "product master data," the term "master" suggesting that such data is the foundation for many different data processing tasks that may need to understand the attributes of a product. Such tasks include goals primarily of value to businesses ("internal" or "B2B" applications) and also goals of value to consumers ("B2C" applications).

For many years, there have been GS1 Standards that define product master data, principally the standards comprising the Global Data Synchronisation Network (GDSN). These standards include definitions of thousands of product data attributes, the Global Product Classification standard (GPC), and standards for the B2B exchange of product master data over the public Internet. However, these standards have been primarily aimed at meeting the needs of B2B applications within the supply chain, especially the communication of product master data from manufacturers to retailers in order to automate various business processes in the order-to-cash process between those parties.

The focus of this document, in contrast, is structured data for use in B2C applications. There are key differences between B2C applications and the sort of B2B applications towards which GDSN is aimed:

- The set of data attributes required by B2C applications differs from B2B data attributes, although there are significant areas of commonality.
- The approach to delivering structured data for B2C applications is based the open interaction model of the World Wide Web, not the closed point-to-point approach in GDSN.
- B2C applications require integrating data from multiple sources.

There are many facets to the overall problem of providing structured product data for B2C applications. This document focuses specifically on how structured product data may be embedded into public-facing Web pages that present products and information about products to consumers. Other GS1 Standards address other facets of the B2C problem; for example, the GS1 Source (Trusted Source of Data) standard addresses the distribution of B2C data to Internet application providers via a network of known "data aggregators" that act as hubs for the distribution of data about products from many brand owners.

The technical approach described in this document, in contrast, is based on:

- Resource Description Format (RDF) as the language for expressing structured data
- Schema.org and GS1 vocabularies to populate the structured data
- JavaScript Object Notation for Linked Data (JSON-LD) as the machine-readable syntax for encoding the structured data into a format that can be easily embedded into a web page. Compared to alternative syntaxes for RDF (including Microdata, RDFa, and Microformats), JSON-LD has the advantage of allowing the structured data to be inserted into an existing web page as a single block of text.

These technologies allow structured product data to be inserted directly into public-facing Web pages, where it is available to any application that consumes those pages. This allows web page publishers to distribute product data directly to consumers of the web page.

The data aggregator-based approach of GS1 Source and the web page approach described in this document are complementary. GS1 Source is designed to address the needs of applications that need reliable, authoritative access to product data about a large range of products. Mobile applications that scan a barcode or search for a product are good examples of these. The web page approach described in this document is designed to address the needs of applications that need deeper insight into the content of a particular web page that the application happens to be

170 consuming. Web search engines such as Google, Bing, and others are very important examples of  
 171 such applications; social media sites, shopping engines, and other emerging applications are others.  
 172 While the ultimate aim of this document is to provide technical guidance, the document also  
 173 includes background and business motivation, too.

174 **1.2 Who Will Use this Document?**

175 Brand Owners, Manufacturers, Retailers, Advertising Agencies and Search Engine Optimisation  
 176 (SEO) Strategists can benefit from providing structured data about products and product offerings in  
 177 order to improve the visibility / discoverability of those products or offerings on the web. Near-term  
 178 benefits include enhanced search results such as “rich snippets” presented by search engines such  
 179 as Google and Bing, and recognition of products by social media platforms that are already prepared  
 180 to process structured data. However, the provision of structured data about products and offerings  
 181 from trusted authoritative sources (such as the brand owner or retailer) also serve to build a data  
 182 platform of Linked Data that can support a large number of novel smartphone applications (apps)  
 183 that are related to products and provide consumers with the information they need to:

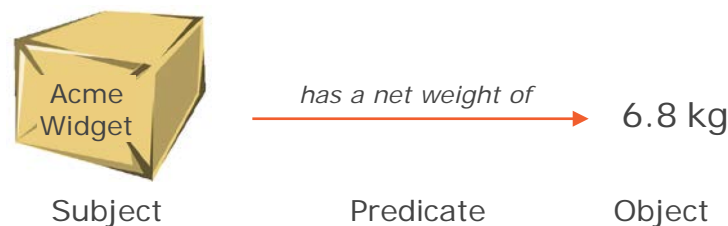
- 184 ■ Choose products that best suit their needs
- 185 ■ Access services, such as instruction manuals, recipes, troubleshooting guides, warranty  
 186 registration, information about related products, accessories and consumables.

187 Section 2 provides further details about the business motivation for implementation.

188 **1.3 Brief Introduction to Linked Data about products and offerings**

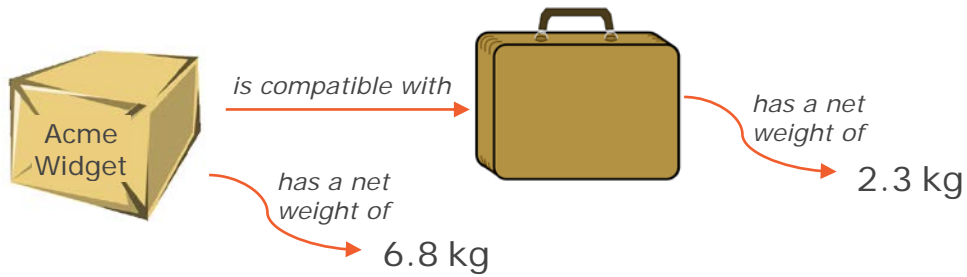
189 Product master data can be thought of as information that connects a product with its Global Trade  
 190 Item Number (GTIN), its brand name, product name, description, net weight, technical data,  
 191 ingredients list or material composition, nutritional information, usage instructions, etc. Some  
 192 properties are generally applicable to all products. Others (such as nutritional information) are only  
 193 applicable to specific sub-classes of products, such as Food / Beverage / Tobacco products.

194 Linked Data technology provides a mechanism for exchanging and linking such structured data  
 195 about products on the web. In Linked Data, each relationship between a thing and an attribute that  
 196 describes the thing is expressed as *triple* of Subject – Predicate – Object; for example, “The Acme  
 197 Widget product (subject) has a net weight (predicate) of 6.8 kilograms (object).” The formal  
 198 computer language for information expressed in this form is called Resource Description Format  
 199 (RDF) and each Subject – Predicate – Object relationship is simply called an “RDF Triple.”



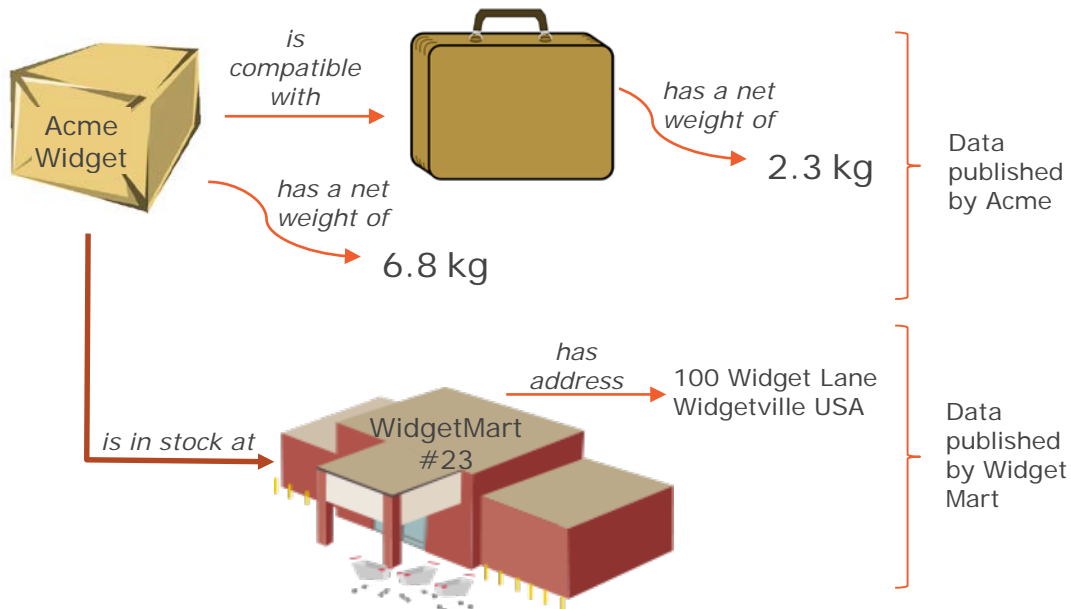
200  
 201  
 202 The Object of one RDF triple might be the Subject of another. For example, in the triple “The Acme  
 203 Widget product (subject) is compatible with (predicate) the Acme Widget Blue Carrying Case  
 204 (object)” the Object of this triple (the carrying case) could well be the Subject of many other triples  
 205 that provide information about the carrying case.





206  
207  
208  
209  
210  
211  
212

The power of Linked Data arises not only from the ability to link RDF Triples together, but also because linked RDF Triples may come from many sources. For example, the manufacturer of Acme Widgets might publish the triples illustrated above, and independently a retailer might publish another triple "The Acme Widget product (subject) is in stock (predicate) at WidgetMart Store #23 (object)". The information about the product published by the manufacturer is now linked with the information about product availability published by the retailer.



213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229

When we want to provide this structured data in a machine-interpretable way, we need to use HTTP URIs to identify the Subject, Predicate and any Object that is not a simple value such as a text string, date/time or number. We use web vocabularies or ontologies to choose concepts, categories or 'classes' and associated properties, attributes or 'predicates' defined in such web vocabularies or ontologies, each associated with an HTTP URI for a specific predicate or class.

For the Subject of such 'facts' about a product or product offering, the brand owner / manufacturer or retailer needs to create HTTP URIs based on one of their own registered domain names. The HTTP URIs for Predicates are usually taken from existing web vocabularies or ontologies, such as Schema.org, GoodRelations or the new GS1 web vocabulary, in order to use specific properties or attributes (e.g. price, weight, description) that are defined in such vocabularies or ontologies. In Section 3.1 and Section 3.2, we explain how brand owners / manufacturers and retailers can construct such HTTP URIs for identifying the product or product offering as the Subject of such 'facts'.

We can use markup formats such as JSON-LD, RDFa or Microdata to embed the structured data in web pages. We then need to serve the Linked Data, either directly or by embedding it within an existing web page.

## 230 1.4 HTTP URIs in Linked Data

231 The following notes should be read as background to both of these sections.

232 HTTP URIs can be used to identify both information resources (such as web pages) and non-  
233 information resources (such as products, people, places and organisations in the real world). Given  
234 an HTTP URI, it is easy to access information simply by making an HTTP request (web request).  
235 Beyond what is specified in the URI standard, the structure and syntax of an HTTP URI is opaque,  
236 which means that by inspection of the URI alone, it is generally not possible to know whether an  
237 HTTP URI serves to identify an information resource (such as a web page) or a non-information  
238 resource (such as a real-world product).

239 Linked Data technology uses HTTP URIs to identify information resources (such as web pages or  
240 collections of facts about a product) and allow those to be retrieved via a web request. Linked Data  
241 technology also uses HTTP URIs in the expression of facts about real-world things; facts are  
242 represented as RDF triples consisting of a Subject, Predicate, Object. The Subject is the topic about  
243 which the facts are expressed. The Predicate represents a specific Property or Attribute of the  
244 Subject or its Relationship to another thing. The Object represents the value of the specified  
245 Predicate, which may be a simple literal value, such as a Text String, DateTime timestamp or  
246 Number – or a complex data object that has further attributes or properties of its own. A very  
247 simple example of a complex data object is a MeasurementType, which consists of a numeric value  
248 and a unit of measure.

249 When we want to write factual information about a product or product offering, it is a good idea to  
250 choose an HTTP URI that we can use as the Subject of a number of RDF triples. Such an HTTP URI  
251 represents a non-information resource; it represents the product itself. Using an HTTP URI means  
252 that via a web request we can retrieve (dereference) a collection of RDF triples for which that HTTP  
253 URI is the Subject (or possibly the Object) and others can also re-use the same HTTP URI in facts  
254 that they write about the same Subject. For example, a brand owner may construct an HTTP URI  
255 for each of its product GTIN values and retailers that sell those products can quote those HTTP URIs  
256 verbatim in facts that they write about their offerings for the product. In this way, the retailer can  
257 write Linked Data that expresses that a specific product offering includes a specific product from a  
258 particular brand owner or manufacturer. In this way, it is not strictly necessary for the retailer to  
259 replicate or embed all of the facts about the product that were asserted by the Brand Owner, since  
260 these are usually retrievable via a web request for the HTTP URI constructed by the brand owner.

## 261 2 Business Motivation for Deploying Linked Data About 262 Products

### 263 2.1 Why should I introduce structured data on my web site?

264 There are a number of different motivations for including structured data within your web site.  
265 The most common are:

- 266 ■ Provide easier access to information about your products and product offerings to consumers  
267 through improved search results and “rich snippets”
- 268 ■ Expose information about your products and product offerings in a way that enables it to be  
269 consumed by applications and mobile apps
- 270 ■ Expose information in a way that will enable it to be linked to other sources of information about  
271 the same product e.g. to link together information from brands, retailers, consumers and other  
272 parties
- 273 ■ Provide a way to link between physical products and their identity on the web (by means of QR  
274 code and mobile scanning)

### 275 2.2 Is this new technology?

276 No. Many brands and retailers have already introduced some structured data within their websites in  
277 order to enhance the search results for their products and product offerings. The GS1 Web  
278 Vocabulary uses this existing technology to extend the range of product attributes that can be

279 included within your web pages by reusing the standard definitions that already exist within the GS1  
280 data dictionary and global product classification.

### 281 **2.3 Why is GS1 involved and why have they developed the GS1 Web** 282 **Vocabulary?**

283 Over the years the GS1 community have developed standard ways to represent information about  
284 products in order to support members in their ability to introduce new products and manage their  
285 end-to-end supply chain. The GS1 SmartSearch initiative was developed to enable these existing  
286 standards to be used to expose consumer facing product information directly within web pages. It is  
287 anticipated that the GS1 Web Vocabulary for product information will enable brand owners and  
288 retailers to describe product characteristics in richer detail than they can currently do using existing  
289 ontologies such as [schema.org](http://schema.org) or [GoodRelations](http://GoodRelations). By including GS1 keys and classifications (GTIN  
290 and GPC) it is anticipated that users will facilitate the linking of data for products and offers between  
291 retailer, manufacturer, search provider and other sites.

### 292 **2.4 I already use Schema.org to describe my products. Why should I use the** 293 **GS1 Web Vocabulary?**

294 The GS1 Web Vocabulary is intended to complement existing ontologies. It is anticipated that web  
295 sites will use both (you can use both within the same page). Reasons to use the GS1 Web  
296 Vocabulary are:

- 297 ■ It is more detailed so enables you to assert more facts about your products and offerings. For  
298 example: It is fully aligned to the latest EU regulation regarding the online disclosure of product  
299 information to consumers (EU 1169/2011).
- 300 ■ Attributes within the GS1 Web Vocabulary are derived directly from existing GS1 standard terms  
301 which will assist companies who are already using GS1 standards

302 Note that where an attribute in the GS1 Web Vocabulary already exists within Schema.org this is  
303 expressed using a [‘sameAs’ statement](#) to provide a mapping between the terms.

### 304 **2.5 What are the benefits for brand owners and manufacturers?**

305 Brand owners can benefit from Linked Data because this will ensure that their products are highly  
306 visible on the web (including detailed product specifications, ingredients, nutritional information,  
307 health, environmental and ethical accreditations, as well as links to technical datasheets, instruction  
308 manuals and online help). Using Linked Data technology, small manufacturers of specialised niche  
309 products can achieve the same web visibility of their products as larger manufacturers of mass-  
310 market products. By using structured data manufacturers and brands can make unambiguous and  
311 authoritative information about their products directly accessible to consumers, thus reducing the  
312 risk that consumers will rely on potentially poor quality or inaccurate/outdated information from  
313 alternative sources.

314 Additionally, by providing rich structured data about products and product offerings, brand owners  
315 can improve insight into the search criteria that consumers are using to find and select the products  
316 that best match their needs; the open availability of rich structured data enables a range of new  
317 consumer-facing applications for product search and comparison – these may be in the form of  
318 smartphone apps or in-store consumer apps. This in turn can provide brand owners with  
319 information about consumer behaviour and preferences that enables them to improve their products  
320 to focus on improving the criteria of interest to consumers (e.g. particular product specifications or  
321 environmental / ethical accreditations).

### 322 **2.6 What are the benefits for retailers?**

323 Retailers can benefit from Linked Data technology by ensuring that their product offerings are highly  
324 visible on the web (including details of promotions and special offers, availability, customer reviews  
325 and ratings, cross-selling opportunities for related products and accessories). Linked Data can be  
326 used to provide an enhanced customer experience on the web and via retailer smartphone apps or  
327 in-store consumer apps. For example, some retailers have already developed digital shopping list

328 apps that enable consumers to make intelligent choices about products, based on their individual  
329 dietary requirements, or preferences on health, environmental or ethical issues.

330 Additionally, by providing rich structured data about products and product offerings, retailers may  
331 gain insight into the search criteria that consumers are using to find products and select the  
332 products that best match their needs; the open availability of rich structured data enables a range  
333 of new consumer-facing applications for product search and comparison – these may be in the form  
334 of smartphone apps or even be installed within in-store scan-as-you-shop handheld terminals for  
335 customer use. This in turn can provide retailers with information about consumer behaviour and  
336 preferences that enables them to give higher priority to stocking products that match the particular  
337 criteria of interest to consumers (e.g. particular product specifications or environmental / ethical  
338 accreditations)

## 339 **2.7 What are the benefits for consumers?**

340 The introduction of structured data will benefit consumers by:

- 341 ■ Giving them more accurate and helpful search results for products that meet their needs e.g. a  
342 search result for a product that meets my dietary needs, near me now, provided with a suitable  
343 rich snippet of information about the product and where it can be bought
- 344 ■ Making it easier to carry out side-by-side comparison of products because the meaning of data  
345 is less ambiguous than plain text within a page
- 346 ■ Helping consumers have confidence in the trustworthiness of product data (that brands choose  
347 to publish)
- 348 ■ Giving consumers new ways to access product information, by search or QR scanning, using  
349 applications and apps that analyse and present information in a standard way regardless of  
350 brand or retailer e.g. nutrition and ingredients.

## 351 **2.8 Who publishes the structured data?**

352 It is envisaged that brand owners will choose to publish facts about their products (GS1 trade items)  
353 and that retailers will publish facts about their product offerings (and often reference or include facts  
354 asserted by the brand owner)

## 355 **2.9 Who can consume the structured data?**

356 Including structured data within a web page makes it accessible to any machine with access to that  
357 page. Typical consumers of structured data are search engines and data aggregators –but could be  
358 anyone or any other piece of software that wishes to make use of the data.

## 359 **2.10 How can structured data for a product be accessed using a QR code?**

360 By including an HTTP URI such as `http://id.examplebrand.com/gtin/05011476100885` within a QR code  
361 on a product, a scanning app will be directed to the product's identity on the web. Depending upon  
362 the type of request made by the app it will served either with structured data about the product or  
363 will be re-directed to any normal web page relating to the product. It is therefore possible to serve  
364 the needs of both existing QR scanning applications and new applications that may want to process  
365 structured data relating to the product.

## 366 **2.11 How will information about retail product offerings be made available to consumers via search and apps?**

368 Linked data will enable search engines and other web information providers to blend together  
369 information about products about retailer offers in a way that best fits the requirements and context  
370 of a given request. For example: A picture and description of a product (from the brand) merged  
371 with nearby retailer offerings of that product its availability and delivery/collection options.

## 372 2.12 How do I get started?

373 If you have no prior experience with structured data then the following practical suggestions may  
374 help guide your first steps.

### 375 Establish the opportunity within your company

- 376 ■ Understand how your company currently makes information about its products or offers  
377 available on the web (including how accurately or prominently this information can be found  
378 within web search results).
- 379 ■ Find out whether your company is already using structured data within its web pages, either  
380 from your web site team or using freely available tools such as:
  - 381 □ <http://www.google.com/webmasters/tools/richsnippets>
  - 382 □ <http://linter.structured-data.org/>
- 383 ■ Examine the information displayed on your website and how it maps to the schema.org and GS1  
384 vocabularies.
- 385 ■ Experiment with adding structured data to your pages, using the schema.org and GS1  
386 vocabularies, to improve your understanding and measure the impact that this has upon search  
387 and the visibility of your product or offer. Initially this could be for just a small number of  
388 attributes such as GTIN, product or offer description and image. (You should strongly consider  
389 including GTIN if you want to make it easy for search and other applications to associate  
390 structured data on your web pages with other relevant data.)

### 391 Build knowledge and capability within your company

- 392 ■ Promote and build awareness of the potential benefits of structured data with your business  
393 commercial and marketing teams using the information in this implementation guide and any of  
394 your own experience.
- 395 ■ Share the technical sections of this guide with your web developers and understand whether  
396 they are already familiar with or using any of the related standards such as RDFa, schema.org,  
397 JSON-LD. Encourage them to experiment and become familiar with the standards and  
398 technology.

### 399 Provide feedback and get involved in the development of GS1 standards for the web

- 400 ■ Review and provide feedback on the GS1 web vocabulary and this guide based upon your own  
401 experience.
- 402 ■ Get involved with the GS1 working groups that maintain this guideline and the related GS1  
403 Standards. There are also industry associations that do related work.

## 404 2.13 What training and education do GS1 Member Organisations need to 405 develop? Who gives guidance?

406 We recommend that GS1 Member Organisations co-ordinate very closely with GS1 SmartSearch  
407 activities sponsored by the GS1 Global Office, to avoid duplication of effort or conflicting  
408 information.

409 The GS1 SmartSearch MSWG is already committed to developing guidance material about how to  
410 publish Linked Data for products - including how to use the GS1 Web Vocabulary and HTTP URIs for  
411 products and product offers, incorporating the GTIN and other qualifiers (e.g. Serial Number,  
412 Lot/Batch) where further granularity is required.

## 413 2.14 What are the challenges going to be in terms of getting retailers and 414 brands on board?

415 Today – many brands and retailers have already introduced structured data within their websites in  
416 order to improve search results. But many companies are only just beginning to understand how the  
417 web is evolving and the benefits of embracing semantic / linked data technologies. For many, the  
418 initial incentive may be the opportunity to achieve enhanced search results, such as “rich snippets,”

419 whereas a few are considering the longer term benefits and the new kinds of opportunities and  
420 product-related services that could be enabled in the next generation of smartphone apps if rich  
421 structured data about products is readily available on the web. As with any new technology, there  
422 can be a 'first mover advantage' but some companies are still unclear how to proceed with new  
423 unfamiliar technology – or are concerned about whether they somehow relinquish control if they  
424 publish such data using Linked Data technology. The fact is that initially, much of the structured  
425 machine-processable data will not be data that is commercially sensitive (such as traceability data)  
426 but information that is already effectively in the public domain because it appears on the packaging  
427 or products or in human-readable format in public web pages. For its part, GS1 is trying to educate  
428 its users about Linked Data technology, the potential benefits – and provide not only a GS1 web  
429 vocabulary that is aligned with the precise definitions of terms developed by the GS1 community  
430 through a consensus process spanning several decades – but also to provide tools such as JSON-LD  
431 templates that should make it much easier for companies to adopt Linked Data technology. GS1 is  
432 also encouraging and supporting a number of pilot activities in this area.

## 433 **2.15 Why is it attractive to marketing / SEO agencies?**

434 For marketing agencies, Linked Data technology helps to fine-tune web search results and helps  
435 consumers to find exactly the products and services that are of interest to them, because the  
436 detailed product characteristics become more readily available to search engines and smartphone  
437 apps. Advertising agencies that become deeply familiar with semantic / Linked Data technologies  
438 will be in the strongest position to provide the greatest value to clients as the web evolves.

## 439 **2.16 How will search engine listings be impacted by structured data?**

440 Using the schema.org, GoodRelations or GS1 vocabularies, it is possible to associate rich structured  
441 data about products with the globally unique identity of the product (its Global Trade Item Number  
442 or GTIN – typically the number on the barcode) by using properties such as  
443 <http://schema.org/gtin13>.

444 The use of the GTIN provides a consistent cross-reference between information provided by the  
445 brand owner and multiple retailers and resellers of the product. It enables search engines to quickly  
446 determine which data about a product is consistent – and which information is likely to be  
447 misleading.

448 Furthermore, GS1 is working on a mapping of its product classification systems (such as GPC Global  
449 Product Classification) to a Linked Data representation, which means that consumers will be able to  
450 more reliably find products that match particular categories and criteria / attributes even when the  
451 consumer searches by keyword and does not have a specific brand in mind.

452 Search engines including Google and Bing use structured data in order to produce "rich snippets."  
453 These are the enhanced search listings that typically appear on the right-hand side of the search  
454 results page, often including photos, maps, opening hours, price information (or in the case of music  
455 bands, lists of their albums and upcoming concerts).

456 Rich snippets may draw upon information from multiple sources - there is not always an exact 1-1  
457 relationship with the structured data added to the website - and they might only display a subset of  
458 the information or complement it with information from elsewhere (e.g. blending product master  
459 data from the brand owner with information about the offering (price, promotions etc.) from the  
460 retailer). So in the case of music bands, the rich snippet may be a blend of information from the  
461 band's own web page and sites such as MusicBrainz and Wikipedia.

462 However, for products, adding structured data to an existing web page is probably the best method,  
463 although it is also possible to provide new 'pages' (scripts or servlets) that serve only the data about  
464 products, e.g. as JSON-LD or [RDF Turtle](#) even if there is no corresponding web page.

## 465 **2.17 How will the search engine surface products searched under 'red shoes' for example?**

466  
467 Within GS1, we have the Global Product Classification (GPC) system, although currently very few  
468 brand owners and retailers include the GPC brick values and attribute-value pairs within the markup  
469 for their web pages. We have done an initial crude translation of GPC into RDF format, but need to  
470 do some further work to make it more web-developer friendly, avoiding the need for them to

471 understand the current 8-digit GPC codes, some of which appear to have been assigned on a first-  
472 come-first-served basis.

473 In parallel, GS1 US is working with some major companies on the design of a Structured Commerce  
474 Classification code.

475 We expect that a developer-friendly GS1-endorsed product classification system suitable for use  
476 with Linked Data will emerge from these efforts and that when brand owners and/or retailers make  
477 use of this, searches by product category and attribute will become easier using Linked Data. The  
478 GS1 Web Vocabulary will also include some support for product categories and attributes, with  
479 particular support for quantitative attribute-value pairs (as currently expressed in the GDSN data  
480 model, whereas GPC is primarily concerned with qualitative attribute-value pairs).

## 481 **2.18 Will the semantic web become integral to search engine optimisation** 482 **(SEO)?**

483 Linked Data technologies are already becoming integral to SEO. Even outside of retail and consumer  
484 products, organisations and individuals are improving their own SEO by using semantic  
485 technology. For example, many musicians and bands are already benefitting from Google Rich  
486 Snippets because they have put structured data about their discography into MusicBrainz, a  
487 biography into Wikipedia, their upcoming concert listings into BandsInTown or SongKick and they  
488 have cross-referenced between these sites and also linked to their websites and channels on various  
489 social media networks (e.g. Facebook, YouTube, Twitter), which means that search engines identify  
490 the connections across these constellations of linked data and begin to recognise them as 'Named  
491 Entities' with interesting facts and figures – and the bands or musicians benefit from enhanced  
492 search results such as rich snippets.

## 493 **2.19 How will search engines know who the trusted source of data is?**

494 The structured data includes property tags such as `http://schema.org/brand` so if for example a  
495 brand owner page at `nestle.com` includes that property tag that points to a data object of type  
496 `http://schema.org/Brand` and itself having a `http://schema.org/name` property of "Nestlé", then  
497 a search engine should be fairly confident that the data is coming from that brand owner.

498 If a retailer such as Tesco has a web page about a Nestlé product (e.g. KitKat), they can also  
499 include structured data markup to say that the product in their offer is from the brand "Nestlé" -  
500 which might result in a search engine merging some structured data provided by the brand owner  
501 with other structured data from the retailer (e.g. about price and availability or retailer promotions)

502 At a technical level, it is also possible to use Data Provenance standards from W3C  
503 ([http://www.w3.org/standards/techs/provenance#w3c\\_all](http://www.w3.org/standards/techs/provenance#w3c_all)) to express the source of the data and  
504 how it has been transformed. Usage of JSON-LD or quads rather than triples make it very easy to  
505 attach metadata assertions about authorship to the structure data.

## 506 **2.20 How and will sponsored search results, such as Google AdWords, be** 507 **impacted?**

508 It's hard to predict. [Google AdWords](#) and similar offerings from other search engines are commercial  
509 offerings from those search engines to promote search results to a higher position when the search  
510 query contains those words. At this time, no search engine has indicated if or how it intends to use  
511 Linked Data to affect sponsored search results.

## 512 **2.21 How do we engage the web development world?**

513 The GS1 Web Vocabulary and guidance material mentioned above will be published, to provide  
514 sufficient information to web developers about how they can make use of the rich structured data  
515 about products.

516 **2.22 What are the implications for merchants' product data feeds to search**  
517 **engine marketplaces?**

518 GS1 is in discussions with Google about their Google Shopping merchant feeds and how we can map  
519 from the GS1 Web Vocabulary to the characteristics they expect in the Google Shopping merchant  
520 feeds. It is conceivable that we or they might develop some translation tools so that data marked  
521 up with the GS1 Web Vocabulary can easily be transformed into the expected markup for the Google  
522 Shopping merchant feeds, even if Google and GS1 currently use slightly different terminology for  
523 some attributes that are semantically equivalent.

524 **2.23 Walk me through how search could be enhanced in future, based upon**  
525 **linked data**

526 The more widespread the use of linked data becomes the more valuable it will become as an aid to  
527 consumers who are searching for products and product offers. An illustration of this is show below:

- 528 ■ A consumer enters a general search term into an app or search engine
- 529 ■ The search term is matched against the global product classification (GPC) or similar to reveal  
530 the product attributes and values relevant to the class of products being searched for
- 531 ■ The consumer is offered the option to refine their search by setting values of the attributes  
532 provided and specifying additional criteria, e.g. their budget, location or how urgently they need  
533 the product
- 534 ■ Products (GTINs) that match the GPC and attribute-values provided are considered. (The GTIN  
535 identifier links to additional information such as product specifications, weight, ingredients,  
536 nutritional information, image or description).
- 537 ■ Retailers offering these GTINs can be identified (which provides a link to the retailers price,  
538 current availability etc).
- 539 ■ Linked data about the location of retailers can be used to display products meeting the original  
540 search criteria on a map.

541 **3 Implementation Procedures**

542 This section explains procedures for implementing structured linked open data about products or  
543 product offerings, using a single block of JavaScript Object Notation for Linked Data (JSON-LD)  
544 embedded within a web page, either within the header (<head> section) or at the end of the <body>  
545 section.

546 Appendix A provides the essential technical background about Linked Data / Semantic Web and  
547 specifically about HTTP URIs, JSON-LD and the GS1 Web Vocabulary.

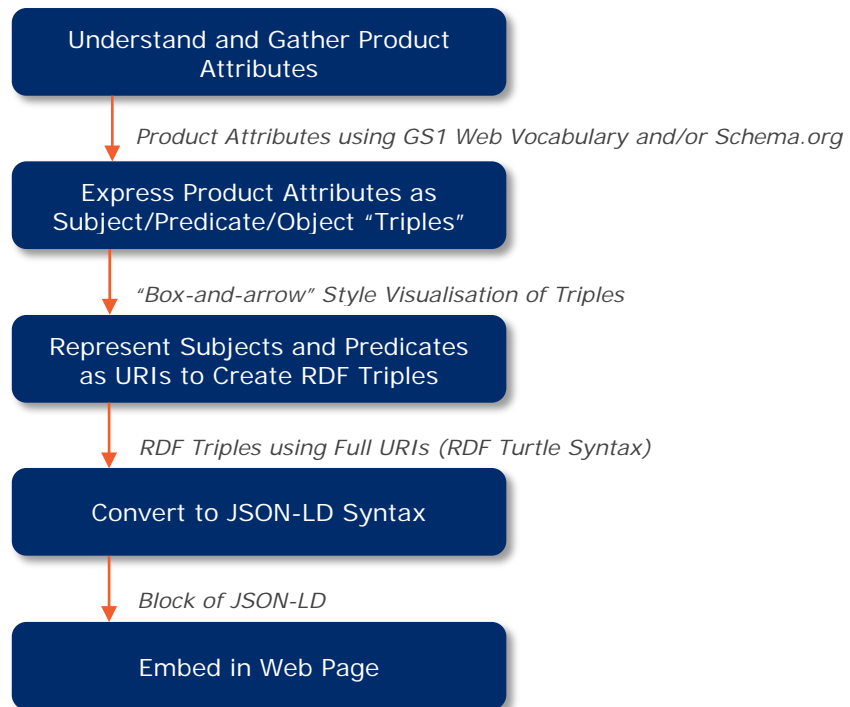
548 Brand Owners / Manufacturers should consider the implementation procedures detailed in Sections  
549 3.1, 3.3, 3.5, and 3.7 through 3.11.

550 Retailers should consider the implementation procedures detailed in all sections 3.1 through 3.11.

551 The process of creating linked data is somewhat complex, so it is best to conceive of the process in  
552 stages:

553





554

### 555 3.1 Procedure for brand owners or manufacturers to construct an HTTP URI 556 to identify a product in facts asserted using Linked Data

557 A brand owner or manufacturer wishes to construct an HTTP URI within one of their own registered  
558 domain names for the purpose of identifying a non-information resource (e.g. a physical product or  
559 digital product) so that they can assert facts about that object as Linked Data.

#### 560 3.1.1 Pre-Requisite

561 We advise reading Technical Appendix A in order to familiarise yourself with the essentials of Linked  
562 Data technology.

#### 563 3.1.2 When Would I Use This?

564 A brand owner or manufacturer should create one HTTP URI per product GTIN for each product  
565 GTIN they issue. If additionally they also wish to be able to write facts or serve data at finer  
566 granularity than the GTIN (e.g. GTIN + Lot/Batch or GTIN+Serial Number), then they should  
567 additionally create HTTP URIs based on those combinations of GTIN and other qualifiers. Typically  
568 this procedure is only performed once, to define a pattern for constructing HTTP URIs for a given  
569 GTIN (or GTIN + qualifiers).

#### 570 3.1.3 How To?

571 A brand owner or manufacturer uses an existing Internet domain name registered to it – or  
572 registers a new domain name specifically for this purpose. It is a good idea to use relatively short  
573 domain names if the corresponding URLs will be used with QR codes, since short URLs require fewer  
574 pixels for encoding in a QR code, resulting in a code that is more 'chunky' and easier to read at a  
575 distance or when the optical resolving power of some smartphone camera optics is relatively low.

576 From this domain name, the brand owner constructs an HTTP URI pattern to be used for each of its  
577 products.

578 For example, if a brand owner currently leases the domain name `brandexample.com`, they might  
579 construct HTTP URIs such as:

580 `http://id.brandexample.com/gtin/00614141123452`

581 or

582 `http://brandexample.com/id/gtin/00614141123452`

583 Of course from a Linked Data perspective, there is no requirement that the GTIN value should  
584 appear within the HTTP URI, whether that URI is used in RDF triples or encoded within a QR code,  
585 Near-Field Communication (NFC) tag, or other data carrier. Including the GTIN within the URI is  
586 merely a convenience for the brand owner or manufacturer, to make it easier to avoid duplication  
587 and to remember which URI is intended to refer to which product.

588 A separate document will provide guidance about encoding of HTTP URIs in QR codes (either regular  
589 QR codes or GS1 QR codes). For use with GS1 QR codes in which the GTIN and other qualifiers  
590 (such as Lot/Batch number or Serial Number) are separately encoded using GS1 Application  
591 Identifiers, such guidance is likely to propose a mechanism for constructing a virtual canonical HTTP  
592 URI from a short HTTP URI prefix together with the GTIN and other qualifiers. However, that virtual  
593 canonical HTTP URI might never be written in any facts and might only be configured (via the URI  
594 rewriting rules of a webserver) to redirect to the HTTP URI that a brand owner prefers to use.

595 It should also be noted that the appearance of a GTIN (or other qualifiers) within an HTTP URI  
596 should not be interpreted as a reliable assertion that the URI represents a product or product  
597 offering with a specific GTIN or links to information about a product or product offering with that  
598 GTIN. The only reliable way to draw that conclusion is if there is a specific RDF triple that asserts  
599 that the Subject has a specific GTIN value. One way to do this is by using the schema.org  
600 Predicates: `http://schema.org/gtin13` or `http://schema.org/gtin14`.

## 601 **3.2 Procedure for retailers to construct an HTTP URI to identify a product** 602 **offering in facts asserted using Linked Data**

603 A retailer wishes to construct an HTTP URI within one of their own registered domain names for the  
604 purpose of identifying a non-information resource (e.g. an offering for a particular product) so that  
605 they can assert facts (such as price, availability and promotional offers) about that offering as  
606 Linked Data.

### 607 **3.2.1 Pre-Requisite**

608 We advise reading Technical Appendix A in order to familiarise yourself with the essentials of Linked  
609 Data technology.

### 610 **3.2.2 When Would I Use This?**

611 A brand owner or manufacturer should create one HTTP URI per product GTIN for each product  
612 GTIN they offer for sale. If additionally they also wish to be able to write facts or serve data at finer  
613 granularity than the GTIN (e.g. GTIN + Lot/Batch or GTIN+Serial Number), then they should  
614 additionally create HTTP URIs based on those combinations of GTIN and other qualifiers. Typically  
615 this procedure is only performed once, to define a pattern for constructing HTTP URIs for a given  
616 GTIN (or GTIN + qualifiers).

### 617 **3.2.3 How To?**

618 A retailer or reseller uses an existing Internet domain name registered to it – or registers a new  
619 domain name specifically for this purpose. It is a good idea to use relatively short domain names if  
620 the corresponding URLs will be used with QR codes on retailer-specific packaging, since short URLs  
621 require fewer pixels for encoding in a QR code, resulting in a code that is more 'chunky' and easier  
622 to read at a distance or when the optical resolving power of some smartphone camera optics is  
623 relatively low.

624 From this domain name, the retailer constructs an HTTP URI pattern to be used for each of the  
625 products it offers for sale.

626 For example, if a retailer currently leases the domain name `retailerexample.com`, they might  
627 construct HTTP URIs such as:

628 `http://id.retailerexample.com/gtin/00614141123452`

629 or

630 `http://retailerexample.com/id/gtin/00614141123452`

631 Of course from a Linked Data perspective, there is no requirement that the GTIN value should  
632 appear within the HTTP URI, whether that URI is used in RDF triples or encoded within a QR code,  
633 NFC tag, or other data carrier. Including the GTIN within the URI is merely a convenience for the  
634 retailer, to make it easier to avoid duplication and to remember which URI is intended to refer to  
635 which product.

636 A separate document will provide guidance about encoding of HTTP URIs in QR codes (either regular  
637 QR codes or GS1 QR codes). For use with GS1 QR codes in which the GTIN and other qualifiers  
638 (such as Lot/Batch number or Serial Number) are separately encoded using GS1 Application  
639 Identifiers, such guidance is likely to propose a mechanism for constructing a virtual canonical HTTP  
640 URI from a short HTTP URI prefix together with the GTIN and other qualifiers. However, that virtual  
641 canonical HTTP URI might never be written in any facts and might only be configured (via the URI  
642 rewriting rules of a webserver) to redirect to the HTTP URI that a retailer prefers to use.

643 It should also be noted that the appearance of a GTIN (or other qualifiers) within an HTTP URI  
644 should not be interpreted as a reliable assertion that the URI represents a product or product  
645 offering with a specific GTIN or links to information about a product or product offering with that  
646 GTIN. The only reliable way to draw that conclusion is if there is a specific RDF triple that asserts  
647 that the Subject has a specific GTIN value. One way to do this is by using the schema.org  
648 Predicates: `http://schema.org/gtin13` or `http://schema.org/gtin14`.

### 649 **3.3 Procedure for brand owners or manufacturers to construct a simple block** 650 **of JSON-LD to represent basic facts about any product, using the** 651 **schema.org vocabulary**

652 A brand owner or manufacturer wishes to include create linked data about their product so that they  
653 can assert basic facts about that product such as the product's name, description, and image, and  
654 enable others (e.g., retailers) to link to this information.

#### 655 **3.3.1 Pre-Requisite**

656 Section 3.1.

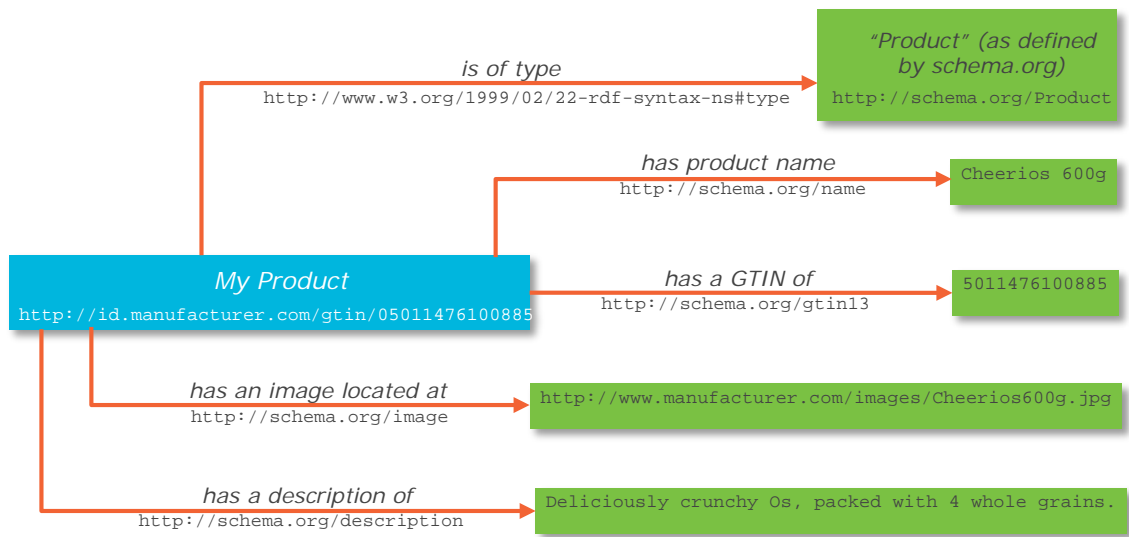
#### 657 **3.3.2 When Would I Use This?**

658 Use this procedure when creating linked data for your product using the schema.org vocabulary.

#### 659 **3.3.3 How To?**

660 The JSON-LD snippet below is a very minimal example showing how a brand owner or manufacturer  
661 could use the schema.org vocabulary to write some simple facts about a product and mark them up  
662 as a single block of JSON-LD.

663 Start with a visualisation of the facts we want to write:



664

665 In this figure, concepts are written in italics, and the URI representation of those concepts as used  
666 in RDF written below that.

667 This corresponds to the following set of RDF triples, written below in [RDF Turtle](#) notation:

```
668 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
669 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
670 @prefix schema: <http://schema.org/> .
671 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
672
673 <http://id.manufacturer.com/gtin/05011476100885> rdf:type schema:Product .
674 <http://id.manufacturer.com/gtin/05011476100885> schema:gtin13 "5011476100885" .
675 <http://id.manufacturer.com/gtin/05011476100885> schema:name "Cheerios 600g"@en .
676 <http://id.manufacturer.com/gtin/05011476100885> schema:image
677 <http://www.manufacturer.com/images/Cheerios600g.jpg> .
678 <http://id.manufacturer.com/gtin/05011476100885> schema:description "Deliciously
679 crunchy Os, packed with 4 whole grains."@en .
```

680 The first four lines of the RDF Turtle notation define namespace prefixes that are used in the  
681 remainder, and the remaining lines of RDF Turtle contain the triples. Each triple is simply written as  
682 Subject Predicate Object, separated by spaces and terminated by a period.

683 Writing RDF Turtle is a good intermediate step because it makes the RDF triples very clear. But to  
684 embed in a web page, you next have to translate this into a web-compatible format. This guideline  
685 recommends JSON-LD as such a format, as it allows the structured data to be inserted into a web  
686 page in a single block rather than being interspersed with the human-facing content. Here is how  
687 the above triples look when written in JSON-LD:

```
688 {
689   "@context": {
690     "schema": "http://schema.org/",
691     "xsd": "http://www.w3.org/2001/XMLSchema#",
692
693     "Product": "schema:Product",
694     "gtin13": { "@id": "schema:gtin13", "@type": "xsd:string" },
695     "name": { "@id": "schema:name", "@language": "en" },
696     "description": { "@id": "schema:description", "@language": "en" },
697     "image": { "@id": "schema:image", "@type": "@id" }
698   }
699 ,
700   "@id": "http://id.manufacturer.com/gtin/05011476100885",
701   "@type": [ "Product" ],
702   "gtin13": "5011476100885",
```

```

703     "name": "Cheerios 600g",
704     "description": "Deliciously crunchy Os, packed with 4 whole grains.",
705     "image": http://www.manufacturer.com/images/Cheerios600g.jpg
706   }
  
```

707 The full JSON-LD block has two parts: the context (shaded orange) and the body (shaded blue). The  
 708 body contains the structured data we wish to publish, and the context sets up abbreviations used in  
 709 the body so that the body is easier to read and process. (In this example, it may not look like the  
 710 context is saving us much. But a block of JSON-LD embedded in a real web page could contain  
 711 much more data and the benefit of the context would more obvious.)

712 Let's first consider the body part of the JSON-LD block (shaded blue). The general structure is a set  
 713 of *property* : *value* pairs. Mostly, in each pair the *property* is an RDF predicate (interpreted with the  
 714 help of the context) and the *value* is an RDF object. But there are some special pairs, too.

715 Let's consider it line by line:

```

716     "@id": "http://id.manufacturer.com/gtin/05011476100885",
  
```

717 The @id property says that we are writing triples about a particular URI; *i.e.* that this URI is the  
 718 subject in all the triples that follow. In this example the URI is the HTTP URI for the product or trade  
 719 item, constructed by the brand owner or manufacturer, using one of their own registered domain  
 720 names. See Section 3.1 for some example patterns of how to construct such an HTTP URI.

```

721     "@type": [ "Product" ],
  
```

722 The @type property says that the identified thing has a particular type, in this case "Product". Note  
 723 however, that the context block defines an expansion of the term "Product" to `schema:Product` (and  
 724 in turn to `http://schema.org/Product`).

725 Taken together, these first two lines are equivalent to the RDF Turtle triple:

```

726 http://id.manufacturer.com/gtin/05011476100885 rdf:type http://schema.org/Product .
  
```

727 The remaining lines are the product data.

```

728     "gtin13": "5011476100885",
  
```

729 This asserts that the identified thing has a specific GTIN-13, in this case 5011476100885. Again, the  
 730 context block expands the term `gtin13` to `schema:gtin13` (and in turn to  
 731 `http://schema.org/gtin13`).

```

732     "name": "Cheerios 600g",
  
```

733 This asserts that the identified thing has the name "Cheerios 600g", through the name property term  
 734 expanded by the context block to `http://schema.org/name`.

```

735     "description": "Deliciously crunchy Os, packed with 4 whole grains.",
  
```

736 This asserts that the identified thing has a particular description as shown, through the description  
 737 property term expanded by the context block to `http://schema.org/description`.

```

738     "image": http://www.manufacturer.com/images/Cheerios600g.jpg
  
```

739 This asserts that the identified thing has an associated image, whose URI is indicated through the  
 740 image property term expanded by the context block to `http://schema.org/image`.

741 Now let's go back to the @context part of the JSON-LD block (shaded orange). It provides  
 742 abbreviations so that the JSON-LD body (shaded blue) is written using simple name strings but  
 743 those local name strings are mapped to HTTP URIs of properties or predicates defined in specified  
 744 web vocabularies or ontologies.

745 Let's consider the context block line by line:

```

746     "schema": "http://schema.org/",
747     "xsd": "http://www.w3.org/2001/XMLSchema#",
  
```

748 These two lines define namespace prefixes that are used in the remaining lines of the context block.

```
749 "Product": "schema:Product",
```

750 This defines `Product` as an abbreviation for `Product` as defined in the `schema.org` namespace, so  
 751 that when `Product` appears in the JSON-LD body it is understood to mean  
 752 `http://schema.org/Product`.

```
753 "gtin13": {"@id": "schema:gtin13", "@type": "xsd:string" },
```

754 This does two things. First, it defines `gtin13` as an abbreviation for `gtin13` as defined in the  
 755 `schema.org` namespace, so that when `gtin13` appears in the JSON-LD body it is understood to  
 756 mean `http://schema.org/gtin13`. Second, it defines the data type of values of the `gtin13`  
 757 property to be strings.

```
758 "name": {"@id": "schema:name", "@language": "en"},
```

759 This does two things. First, it defines `name` as an abbreviation for `name` as defined in the `schema.org`  
 760 namespace, so that when `name` appears in the JSON-LD body it is understood to mean  
 761 `http://schema.org/name`. Second, it says that values of the `name` property are written in English.

```
762 "description": {"@id": "schema:description", "@language": "en"},
```

763 This is similar to the declaration for `name`, but applies to the `description` attribute.

```
764 "image": {"@id": "schema:image", "@type": "@id"}
```

765 This does two things. First, it defines `image` as an abbreviation for `image` as defined in the  
 766 `schema.org` namespace, so that when `image` appears in the JSON-LD body it is understood to mean  
 767 `http://schema.org/image`. Second, it defines the data type of values of the `image` property to be  
 768 identifiers (URIs) which themselves could be subject of other RDF triples.

769 An important point to note is that we have a free choice of the local property names we use – so for  
 770 example, we could have written the following JSON-LD and it would still result in the *same* set of  
 771 RDF triples, even though we have changed all of the local property names compared with the  
 772 previous example. This is important because it means that if a company is internally using JSON  
 773 data within their website and using their own local property names, the `@context` block provides a  
 774 very flexible way to map the local terms to terms defined globally via URIs in web vocabularies and  
 775 ontologies.

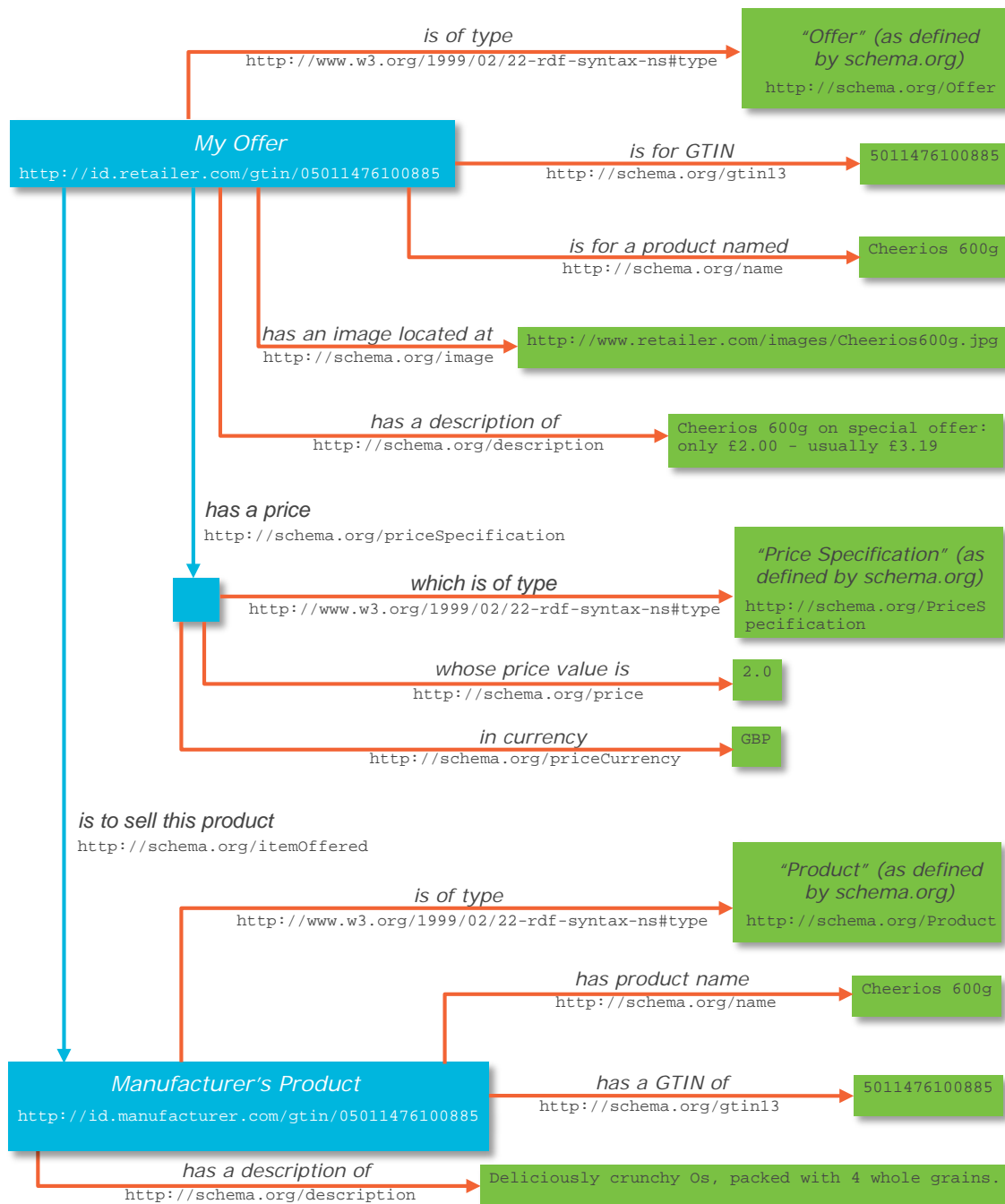
```
776 {
777   "@context": {
778     "schema": "http://schema.org/",
779     "xsd": "http://www.w3.org/2001/XMLSchema#",
780
781     "TradeItem": "schema:Product",
782     "ean13": {"@id": "schema:gtin13", "@type": "xsd:string" },
783     "productName": {"@id": "schema:name", "@language": "en"},
784     "tagline": {"@id": "schema:description", "@language": "en"},
785     "photo": {"@id": "schema:image", "@type": "@id"}
786   }
787 ,
```

```
788   "@id": "http://id.manufacturer.com/gtin/05011476100885",
789   "@type": [ "TradeItem" ],
790   "ean13": "5011476100885",
791   "productName": "Cheerios 600g",
792   "tagline": "Deliciously crunchy Os, packed with 4 whole grains.",
793   "photo": "http://www.manufacturer.com/images/Cheerios600g.jpg"
```

```
794 }
```

795 The above JSON-LD results in exactly the same RDF triples as the JSON-LD given earlier in this  
 796 section, even though all of the local names used in the body are different. The reason it is  
 797 equivalent to the earlier JSON-LD example is that the full predicate URIs defined in the context  
 798 section are the same.

- 799 **3.4 Procedure for retailers to construct a simple block of JSON-LD to**  
800 **represent basic facts about any product offering, using the schema.org**  
801 **vocabulary**
- 802 A retailer wishes to include create linked data about a product offering so that they can assert facts  
803 about that offering such as the offering price and a retailer's own product image, and enable others  
804 to link to this information. At the same time, the retailer wishes its information to be linked to the  
805 manufacturer's information about the same product.
- 806 **3.4.1 Pre-Requisite**
- 807 Sections 3.1, 3.2, and 3.3.
- 808 **3.4.2 When Would I Use This?**
- 809 Use this procedure when creating linked data for your product offering using the schema.org  
810 vocabulary.
- 811 **3.4.3 How To?**
- 812 Start with a visualisation of the facts we want to write:



813

814

815

In this figure, concepts are written in italics, and the URI representation of those concepts as used in RDF written below that.

816

817

818

819

820

821

822

This is rather more complicated than the previous example in Section 3.3 because the retailer needs to assert facts about their offer for a product (such as price information) – but they might also want to include facts asserted by the brand owner or manufacturer. Notice how the Offer has predicates that relate it to two other objects: one, the manufacturer's product which itself is the subject of its own descriptive triples; and two, the price which as a structured value is represented as a subject, with triples providing the price and currency as separate data values. Because the price structure only has local meaning within this triple graph, it does not need a globally unique URI of its own.

823

824

The `http://schema.org/Offer` contains the facts asserted by the retailer (e.g. about price information etc.), while the `http://schema.org/Product` may contains facts originally asserted by



825 the brand owner or manufacturer (e.g. about the product characteristics and specifications, as well  
826 as description).

827 The corresponding RDF triples we want to assert in this example are the following:

```
828 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
829 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
830 @prefix schema: <http://schema.org/> .
831 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
832
833 <http://id.retailer.com/gtin/05011476100885> rdf:type schema:Offer .
834 <http://id.retailer.com/gtin/05011476100885> schema:gtin13 "5011476100885" .
835 <http://id.retailer.com/gtin/05011476100885> schema:name "Cheerios 600g"@en .
836 <http://id.retailer.com/gtin/05011476100885> schema:description "Cheerios 600g on
837 special offer: only £2.00 - usually £3.19"@en .
838 <http://id.retailer.com/gtin/05011476100885> schema:image
839 <http://www.retailer.com/img/Cheerios-600g.jpg> .
840 <http://id.retailer.com/gtin/05011476100885> schema:priceSpecification _:b0 .
841 _:b0 rdf:type schema:PriceSpecification .
842 _:b0 schema:price "2.00"^^xsd:float .
843 _:b0 schema:priceCurrency "GBP" .
844 _:b0 schema:priceType "List" .
845
846 <http://id.retailer.com/gtin/05011476100885> schema:itemOffered
847 <http://id.manufacturer.com/gtin/05011476100885> .
848
849 <http://id.manufacturer.com/gtin/05011476100885> rdf:type schema:Product .
850 <http://id.manufacturer.com/gtin/05011476100885> schema:gtin13 "5011476100885" .
851 <http://id.manufacturer.com/gtin/05011476100885> schema:org:name "Cheerios 600g"@en
852 .
853 <http://id.manufacturer.com/gtin/05011476100885> schema:description "Deliciously
854 crunchy Os, packed with 4 whole grains"@en .
```

855 In RDF Turtle notation, the underscore before the colon in `_:b0` indicates that this is just a local  
856 name that has no significance outside this triple graph (corresponding to the blue square in the  
857 figure above).

858 The JSON-LD block looks like:

```
859 {
860   "@context": {
861     "schema": "http://schema.org/",
862     "xsd": "http://www.w3.org/2001/XMLSchema#",
863
864     "Offer": "schema:Offer",
865     "Product": "schema:Product",
866
867     "productName": {"@id": "schema:name", "@language": "en"},
868     "offerName": {"@id": "schema:name", "@language": "en"},
869     "productDescription": {"@id": "schema:description", "@language": "en"},
870     "offerDescription": {"@id": "schema:description", "@language": "en"},
871     "gtin13": {"@id": "schema:gtin13", "@type": "xsd:string"},
872
873     "image": {"@id": "schema:image", "@type": "@id"},
874     "price": {"@id": "schema:price", "@type": "xsd:float"},
875     "currencyUnit": {"@id": "schema:priceCurrency", "@type": "xsd:string"},
876     "priceType": {"@id": "schema:priceType", "@type": "xsd:string"},
877     "hasPrice": {"@id": "schema:priceSpecification", "@type": "@id"},
878     "includes": {"@id": "schema:itemOffered", "@type": "@id"}
879   },
880   "@id": "http://id.retailer.com/gtin/05011476100885",
881   "@type": "Offer",
882   "gtin13": "5011476100885",
883   "offerName": "Cheerios 600g",
```

```

884 "offerDescription": "Cheerios 600g on special offer: only £2.00 - usually
885 £3.19",
886 "image": "http://www.retailer.com/img/Cheerios-600g.jpg",
887 "hasPrice": {
888   "price": "2.00",
889   "currencyUnit": "GBP",
890   "priceType": "List",
891   "@type": "schema:PriceSpecification"
892 },
893 "includes": {
894   "@id": "http://id.manufacturer.com/gtin/05011476100885",
895   "@type": [ "Product" ],
896   "gtin13": "5011476100885",
897   "productName": "Cheerios 600g",
898   "productDescription": "Deliciously crunchy Os, packed with 4 whole
899 grains."
900 }
901 }
  
```

902 In RDF Turtle, the special notation `_:b0` had to be used to represent the local subject used for the  
 903 price. In JSON-LD, this is expressed more naturally by simply nesting the price attributes within the  
 904 value for `hasPrice`, thereby avoiding the need to introduce the name `_:b0`.

### 905 **3.5 Procedure for brand owners or manufacturers to construct a simple block** 906 **of JSON-LD to represent basic facts about any product, using the GS1** 907 **web vocabulary**

908 A brand owner or manufacturer wishes to include create linked data about their product so that they  
 909 can assert facts about that product such as technical specifications, ingredients, and nutritional  
 910 information, and enable others (e.g. retailers) to link to this information. This is similar to the  
 911 procedure in Section 3.3, but in this case we are using both `schema.org` vocabulary and the GS1  
 912 Web Vocabulary. This allows for the inclusion of a much richer set of product attributes.

913 This example shows a food product and nutritional attributes, but the GS1 Web Vocabulary includes  
 914 specialised product attributes for many other product categories as well.

#### 915 **3.5.1 Pre-Requisite**

916 Section 3.3.

#### 917 **3.5.2 When Would I Use This?**

918 Use this procedure when creating linked data for your product using the extended GS1 Web  
 919 Vocabulary offered by GS1.

#### 920 **3.5.3 How To?**

921 Here is some sample JSON-LD:

```

922 {
923   "@context": {
924     "gs1": "http://gs1.org/voc/",
925     "schema": "http://schema.org/",
926     "xsd": "http://www.w3.org/2001/XMLSchema#",
927
928     "TradeItem": "schema:Product",
929
930     "tradeItemDescription": { "@id": "schema:description", "@language": "en" },
931     "gtin13": { "@id": "schema:gtin13", "@type": "xsd:string" },
932
933     "image": { "@id": "schema:image", "@type": "@id" },
934
  
```

```

935 "healthClaimDescription":{"@id":"gsl:healthClaimDescription","@language": "en"},
936 "allergenStatement":{"@id":"gsl:allergenStatement","@language": "en"},
937
938 "gsl:measurementUnitCode": { "@type": "xsd:string" },
939 "value":{"@id":"gsl:measurementValue","@type":"xsd:float"},
940 "unit":{"@id":"gsl:measurementUnitCode","@type":"xsd:string"},
941
942 "ingredientpercentage":{"@id":"gsl:ingredientContentPercentage","@type":"xsd:float"},
943 "ingredientseq":{"@id":"gsl:ingredientSequence","@type":"xsd:integer"},
944 "ingredientname":{"@id":"gsl:ingredientName","@language": "en"},
945
946 "hasAllergenRelatedInformation":{"@id":"gsl:hasAllergenRelatedInformation","@type":"@id"},
947 "hasIngredients":{"@id":"gsl:hasIngredients","@type":"@id"},
948 "hasIngredientDetail":{"@id":"gsl:hasIngredientDetail","@type":"@id"},
949
950 "nutrientBasisQuantity":{"@id":"gsl:nutrientBasisQuantity","@type":"@id"},
951 "energyPerNutrientBasis":{"@id":"gsl:energyPerNutrientBasis","@type":"@id"},
952 "proteinPerNutrientBasis":{"@id":"gsl:proteinPerNutrientBasis","@type":"@id"},
953 "carbohydratesPerNutrientBasis":{"@id":"gsl:carbohydratesPerNutrientBasis","@type":"@id"},
954 "sugarsPerNutrientBasis":{"@id":"gsl:sugarsPerNutrientBasis","@type":"@id"},
955 "fatPerNutrientBasis":{"@id":"gsl:fatPerNutrientBasis","@type":"@id"},
956 "saturatedFatPerNutrientBasis":{"@id":"gsl:saturatedFatPerNutrientBasis","@type":"@id"},
957 "fibrePerNutrientBasis":{"@id":"gsl:fibrePerNutrientBasis","@type":"@id"},
958 "sodiumPerNutrientBasis":{"@id":"gsl:sodiumPerNutrientBasis","@type":"@id"},
959 "saltPerNutrientBasis":{"@id":"gsl:saltPerNutrientBasis","@type":"@id"},
960 "vitaminCPerNutrientBasis":{"@id":"gsl:vitaminCPerNutrientBasis","@type":"@id"},
961 "thiaminPerNutrientBasis":{"@id":"gsl:thiaminPerNutrientBasis","@type":"@id"},
962 "riboflavinPerNutrientBasis":{"@id":"gsl:riboflavinPerNutrientBasis","@type":"@id"},
963 "niacinPerNutrientBasis":{"@id":"gsl:niacinPerNutrientBasis","@type":"@id"},
964 "vitaminB6PerNutrientBasis":{"@id":"gsl:vitaminB6PerNutrientBasis","@type":"@id"},
965 "folicAcidPerNutrientBasis":{"@id":"gsl:folicAcidPerNutrientBasis","@type":"@id"},
966 "vitaminB12PerNutrientBasis":{"@id":"gsl:vitaminB12PerNutrientBasis","@type":"@id"},
967
968 "pantothenicAcidPerNutrientBasis":{"@id":"gsl:pantothenicAcidPerNutrientBasis","@type":"@id"},
969 "calciumPerNutrientBasis":{"@id":"gsl:calciumPerNutrientBasis","@type":"@id"},
970 "ironPerNutrientBasis":{"@id":"gsl:ironPerNutrientBasis","@type":"@id"},
971
972 "dv":{"@id":"gsl:dailyValueIntakePercent","@type":"xsd:float"},
973
974 "Ingredient":"gsl:FoodAndBeverageIngredientDetail",
975 "Measurement":"gsl:NutritionMeasurementType"

```

```

976 },
977 "@id": "http://id.manufacturer.com/gtin/05011476100885",
978 "gtin13":"5011476100885",
979 "@type": [ "TradeItem" ],
980 "tradeItemDescription": "Deliciously crunchy Os, packed with 4 whole grains. Say Yes to
981 Cheerios",
982 "healthClaimDescription":"8 Vitamins & Iron, Source of Calcium & High in Fibre",
983 "hasAllergenRelatedInformation": {"@type":
984 "gsl:AllergenRelatedInformation","allergenStatement":"May contain nut traces"},
985 "hasIngredients": {"@type": "gsl:FoodAndBeverageIngredient","hasIngredientDetail":{
986 {"@type": "Ingredient", "ingredientseq": "1", "ingredientname": "Cereal
987 Grains", "ingredientpercentage": "77.5"},
988 {"@type": "Ingredient", "ingredientseq": "2", "ingredientname": "Whole Grain
989 OATS", "ingredientpercentage": "38.0"},
990 {"@type": "Ingredient", "ingredientseq": "3", "ingredientname": "Whole Grain
991 WHEAT", "ingredientpercentage": "18.6"},
992 {"@type": "Ingredient", "ingredientseq": "4", "ingredientname": "Whole Grain
993 BARLEY", "ingredientpercentage": "12.8"},
994 {"@type": "Ingredient", "ingredientseq": "5", "ingredientname": "Whole Grain
995 Rice", "ingredientpercentage": "5.5"},
996 {"@type": "Ingredient", "ingredientseq": "6", "ingredientname": "Whole Grain
997 Maize", "ingredientpercentage": "2.6"},
998 {"@type": "Ingredient", "ingredientseq": "7", "ingredientname": "Sugar"},
999 {"@type": "Ingredient", "ingredientseq": "8", "ingredientname": "Wheat Starch"},
1000 {"@type": "Ingredient", "ingredientseq": "9", "ingredientname": "Partially Inverted Brown Sugar
1001 Syrup"},
1002 {"@type": "Ingredient", "ingredientseq": "10", "ingredientname": "Salt"},
1003 {"@type": "Ingredient", "ingredientseq": "11", "ingredientname": "Tripotassium Phosphate"},
1004 {"@type": "Ingredient", "ingredientseq": "12", "ingredientname": "Sunflower Oil"},
1005 {"@type": "Ingredient", "ingredientseq": "13", "ingredientname": "Colours: Caramel, Annatto,
1006 Carotene"},
1007 {"@type": "Ingredient", "ingredientseq": "14", "ingredientname": "Antioxidant: Tocopherals"},
1008 {"@type": "Ingredient", "ingredientseq": "15", "ingredientname": "Vitamin C"},

```

```

1009 {"@type": "Ingredient", "ingredientseq": "16", "ingredientname": "Niacin"},
1010 {"@type": "Ingredient", "ingredientseq": "17", "ingredientname": "Pantothenic Acid"},
1011 {"@type": "Ingredient", "ingredientseq": "18", "ingredientname": "Thiamin (B1)"},
1012 {"@type": "Ingredient", "ingredientseq": "19", "ingredientname": "Vitamin B6"},
1013 {"@type": "Ingredient", "ingredientseq": "20", "ingredientname": "Riboflavin (B2)"},
1014 {"@type": "Ingredient", "ingredientseq": "21", "ingredientname": "Folic Acid (Folacin)"},
1015 {"@type": "Ingredient", "ingredientseq": "22", "ingredientname": "Vitamin B12"},
1016 {"@type": "Ingredient", "ingredientseq": "23", "ingredientname": "Calcium Carbonate"},
1017 {"@type": "Ingredient", "ingredientseq": "24", "ingredientname": "Iron"}
1018 ]},
1019 "nutrientBasisQuantity": {"@type": "Measurement", "value": "100", "unit": "GRM"},
1020 "energyPerNutrientBasis":
1021 [{"@type": "Measurement", "value": "1615", "unit": "KJO"}, {"@type": "Measurement", "value": "382", "unit":
1022 "E14"}],
1023 "proteinPerNutrientBasis": {"@type": "Measurement", "value": "8.6", "unit": "GRM"},
1024 "carbohydratesPerNutrientBasis": {"@type": "Measurement", "value": "74.3", "unit": "GRM"},
1025 "sugarsPerNutrientBasis": {"@type": "Measurement", "value": "21.4", "unit": "GRM"},
1026 "fatPerNutrientBasis": {"@type": "Measurement", "value": "4.0", "unit": "GRM"},
1027 "saturatedFatPerNutrientBasis": {"@type": "Measurement", "value": "1.0", "unit": "GRM"},
1028 "fibrePerNutrientBasis": {"@type": "Measurement", "value": "7.1", "unit": "GRM"},
1029 "sodiumPerNutrientBasis": {"@type": "Measurement", "value": "0.41", "unit": "GRM"},
1030 "saltPerNutrientBasis": {"@type": "Measurement", "value": "1.04", "unit": "GRM"},
1031 "vitaminCPerNutrientBasis": {"@type": "Measurement", "value": "71.0", "unit": "MGM", "dv": "89"},
1032 "thiaminPerNutrientBasis": {"@type": "Measurement", "value": "1.24", "unit": "MGM", "dv": "113"},
1033 "riboflavinPerNutrientBasis": {"@type": "Measurement", "value": "1.10", "unit": "MGM", "dv": "79"},
1034 "niacinPerNutrientBasis": {"@type": "Measurement", "value": "14.0", "unit": "MGM", "dv": "88"},
1035 "vitaminB6PerNutrientBasis": {"@type": "Measurement", "value": "1.20", "unit": "MGM", "dv": "86"},
1036 "folicAcidPerNutrientBasis": {"@type": "Measurement", "value": "200", "unit": "MC", "dv": "100"},
1037 "vitaminB12PerNutrientBasis": {"@type": "Measurement", "value": "1.90", "unit": "MC", "dv": "76"},
1038
1039 "pantothenicAcidPerNutrientBasis": {"@type": "Measurement", "value": "4.40", "unit": "MGM", "dv": "73"}
1040
1041 "calciumPerNutrientBasis": {"@type": "Measurement", "value": "460", "unit": "MGM", "dv": "58"},
1042 "ironPerNutrientBasis": {"@type": "Measurement", "value": "14.7", "unit": "MGM", "dv": "105"}
1043 }
  
```

#### 1044 Explanation:

1045 The context section (shaded orange) references three namespaces – the GS1 vocabulary, the  
 1046 schema.org vocabulary and XSD (XML Schema Definition). (XSD is used for standard data types  
 1047 such as `xsd:float` and `xsd:integer`). The RDF namespace is implicitly included through the JSON-  
 1048 LD `@type` keyword, which maps to `rdf:type`.

1049 Some basic fields such as the description of the offer or the trade item (product), the `gtin13`  
 1050 property, image and price information are mapped to terms from the schema.org vocabulary.

1051 Specialised terms specific to food and beverage products are mapped to terms from the `gs1`  
 1052 vocabulary.

1053 Some of these specialised terms for food product ingredients or nutritional information do not take  
 1054 simple string values but instead take complex data values such as a  
 1055 `gs1:NutritionMeasurementType` (which can be used to express a quantity, a unit of measure and  
 1056 percentage of the recommended daily intake of a nutrient as recommended by authorities of the  
 1057 target market) – or a `gs1:FoodAndBeverageIngredientDetail` (which can accept an ingredient  
 1058 sequence number, ingredient name and ingredient as a percentage of the total composition of the  
 1059 product).

#### 1060 Note: About properties with multiple values, lists, sequences etc.

1061 Another important point to note is that unlike RDF Turtle or N-Triples, in JSON-LD, the name  
 1062 of each property or predicate **should appear only once** in the data block. There may be  
 1063 situations where in RDF triples we might write several triples each containing the same  
 1064 property or predicate, perhaps using blank nodes if the value is not a simple data type. When  
 1065 we want to express these in JSON-LD, we must write the name of the property or predicate  
 1066 **once only** – and use a list for the sets of values corresponding to that property. In the  
 1067 example above, we can see examples of lists in JSON-LD (enclosed in square brackets) for  
 1068 the properties 'hasIngredientDetail' and 'energyPerNutrientBasis'. Lists are used in these  
 1069 examples to allow for multiple ingredients and for two different energy units, respectively.  
 1070

### 3.6 Procedure for retailers to construct a simple block of JSON-LD to represent basic facts about any product offering, using the GS1 web vocabulary

A retailer wishes to include create linked data about a product offering so that they can assert facts about that offering such as the offering price and a retailer's own product image, and enable others to link to this information. At the same time, the retailer also wishes to include detailed product information such as ingredients and nutritional information. This is similar to the procedure in Section 3.43.3, but in this case we are using both schema.org vocabulary and the GS1 Web Vocabulary. This allows for the inclusion of a much richer set of product attributes.

This example shows a food product and nutritional attributes, but the GS1 Web Vocabulary includes specialised product attributes for many other product categories as well.

#### 3.6.1 Pre-Requisite

Sections 3.4.

#### 3.6.2 When Would I Use This?

The following example shows how a retailer can use the GS1 vocabulary in combination with the schema.org vocabulary to write facts about a product offer for a food product. In this example, we have used schema.org properties and classes (shown in red) wherever we can express properties sufficiently precisely using the schema.org vocabulary. For the nutritional information and ingredients list, we use the GS1 vocabulary because it supports a wider variety of nutrients and also allows us to specify an explicit nutrient basis quantity (e.g. 100g or 100ml of product), so that there is no ambiguity about what the quantities (e.g. protein content) relate to.

However, we note that schema.org does define some related properties in <http://schema.org/NutritionInformation> and support expression of a list of ingredients within the context of a <http://schema.org/Recipe> - but schema.org does not currently provide any guidance about how these might be applied reliably to express the nutritional information or ingredients of a food product.

#### 3.6.3 How To?

Here is some sample JSON-LD:

```
{
  "@context": {
    "gs1": "http://gs1.org/voc/",
    "schema": "http://schema.org/",
    "xsd": "http://www.w3.org/2001/XMLSchema#",

    "TradeItem": "schema:Product",
    "Offering": "schema:Offer",

    "offerDescription": {"@id": "schema:description", "@language": "en"},
    "tradeItemDescription": {"@id": "schema:description", "@language": "en"},
    "gtin13": {"@id": "schema:gtin13", "@type": "xsd:string" },

    "image": {"@id": "schema:image", "@type": "@id"},
    "price": {"@id": "schema:price", "@type": "xsd:float"},
    "currencyUnit": {"@id": "schema:priceCurrency", "@type": "xsd:string"},
    "priceType": {"@id": "schema:priceType", "@type": "xsd:string"},
    "hasPrice": {"@id": "schema:priceSpecification", "@type": "@id"},
    "includes": {"@id": "schema:itemOffered", "@type": "@id"},

    "healthClaimDescription": {"@id": "gs1:healthClaimDescription", "@language": "en"},
    "allergenStatement": {"@id": "gs1:allergenStatement", "@language": "en"},

    "gs1:measurementUnitCode": { "@type": "xsd:string" },
    "value": {"@id": "gs1:measurementValue", "@type": "xsd:float"},
    "unit": {"@id": "gs1:measurementUnitCode", "@type": "xsd:string"},

    "ingredientpercentage": {"@id": "gs1:ingredientContentPercentage", "@type": "xsd:float"},
    "ingredientseq": {"@id": "gs1:ingredientSequence", "@type": "xsd:integer"},
  }
}
```

```

1128   "ingredientname":{"@id":"gs1:ingredientName","@language":"en"},
1129
1130   "hasAllergenRelatedInformation":{"@id":"gs1:hasAllergenRelatedInformation","@type":"@id"},
1131   "hasIngredients":{"@id":"gs1:hasIngredients","@type":"@id"},
1132   "hasIngredientDetail":{"@id":"gs1:hasIngredientDetail","@type":"@id"},
1133
1134   "nutrientBasisQuantity":{"@id":"gs1:nutrientBasisQuantity","@type":"@id"},
1135   "energyPerNutrientBasis":{"@id":"gs1:energyPerNutrientBasis","@type":"@id"},
1136   "proteinPerNutrientBasis":{"@id":"gs1:proteinPerNutrientBasis","@type":"@id"},
1137   "carbohydratesPerNutrientBasis":{"@id":"gs1:carbohydratesPerNutrientBasis","@type":"@id"},
1138   "sugarsPerNutrientBasis":{"@id":"gs1:sugarsPerNutrientBasis","@type":"@id"},
1139   "fatPerNutrientBasis":{"@id":"gs1:fatPerNutrientBasis","@type":"@id"},
1140   "saturatedFatPerNutrientBasis":{"@id":"gs1:saturatedFatPerNutrientBasis","@type":"@id"},
1141   "fibrePerNutrientBasis":{"@id":"gs1:fibrePerNutrientBasis","@type":"@id"},
1142   "sodiumPerNutrientBasis":{"@id":"gs1:sodiumPerNutrientBasis","@type":"@id"},
1143   "saltPerNutrientBasis":{"@id":"gs1:saltPerNutrientBasis","@type":"@id"},
1144   "vitaminCPerNutrientBasis":{"@id":"gs1:vitaminCPerNutrientBasis","@type":"@id"},
1145   "thiaminPerNutrientBasis":{"@id":"gs1:thiaminPerNutrientBasis","@type":"@id"},
1146   "riboflavinPerNutrientBasis":{"@id":"gs1:riboflavinPerNutrientBasis","@type":"@id"},
1147   "niacinPerNutrientBasis":{"@id":"gs1:niacinPerNutrientBasis","@type":"@id"},
1148   "vitaminB6PerNutrientBasis":{"@id":"gs1:vitaminB6PerNutrientBasis","@type":"@id"},
1149   "folicAcidPerNutrientBasis":{"@id":"gs1:folicAcidPerNutrientBasis","@type":"@id"},
1150   "vitaminB12PerNutrientBasis":{"@id":"gs1:vitaminB12PerNutrientBasis","@type":"@id"},
1151
1152   "pantothenicAcidPerNutrientBasis":{"@id":"gs1:pantothenicAcidPerNutrientBasis","@type":"@id"},
1153   "calciumPerNutrientBasis":{"@id":"gs1:calciumPerNutrientBasis","@type":"@id"},
1154   "ironPerNutrientBasis":{"@id":"gs1:ironPerNutrientBasis","@type":"@id"},
1155
1156   "dv":{"@id":"gs1:dailyValueIntakePercent","@type":"xsd:float"},
1157
1158   "Ingredient":"gs1:FoodAndBeverageIngredientDetail",
1159   "Measurement":"gs1:NutritionMeasurementType"
1160 },
1161
1162   "@id": "http://id.retailer.com/gtin/05011476100885",
1163   "@type": "Offering",
1164   "gtin13": "5011476100885",
1165   "offerDescription": "Nestle Cheerios Cereal 600G",
1166   "image": "http://www.retailer.com/Groceries/pi/885/5011476100885/IDShot_225x225.jpg",
1167   "hasPrice": {
1168     "price": "2.00",
1169     "currencyUnit": "GBP",
1170     "priceType": "List",
1171     "@type": "schema:PriceSpecification"
1172   },
1173   "includes": {
1174     "@id": "http://id.manufacturer.com/gtin/05011476100885",
1175     "gtin13": "5011476100885",
1176     "@type": [ "TradeItem" ],
1177     "tradeItemDescription": "Deliciously crunchy Os, packed with 4 whole grains. Say Yes to
1178     Cheerios",
1179     "healthClaimDescription": "8 Vitamins & Iron, Source of Calcium & High in Fibre",
1180     "hasAllergenRelatedInformation": {"@type":
1181     "gs1:AllergenRelatedInformation", "allergenStatement": "May contain nut traces"},
1182     "hasIngredients": {"@type": "gs1:FoodAndBeverageIngredient", "hasIngredientDetail": [
1183     {"@type": "Ingredient", "ingredientseq": "1", "ingredientname": "Cereal
1184     Grains", "ingredientpercentage": "77.5"},
1185     {"@type": "Ingredient", "ingredientseq": "2", "ingredientname": "Whole Grain
1186     OATS", "ingredientpercentage": "38.0"},
1187     {"@type": "Ingredient", "ingredientseq": "3", "ingredientname": "Whole Grain
1188     WHEAT", "ingredientpercentage": "18.6"},
1189     {"@type": "Ingredient", "ingredientseq": "4", "ingredientname": "Whole Grain
1190     BARLEY", "ingredientpercentage": "12.8"},
1191     {"@type": "Ingredient", "ingredientseq": "5", "ingredientname": "Whole Grain
1192     Rice", "ingredientpercentage": "5.5"},
1193     {"@type": "Ingredient", "ingredientseq": "6", "ingredientname": "Whole Grain
1194     Maize", "ingredientpercentage": "2.6"},
1195     {"@type": "Ingredient", "ingredientseq": "7", "ingredientname": "Sugar"},
1196     {"@type": "Ingredient", "ingredientseq": "8", "ingredientname": "Wheat Starch"},
1197     {"@type": "Ingredient", "ingredientseq": "9", "ingredientname": "Partially Inverted Brown Sugar
1198     Syrup"},
1199     {"@type": "Ingredient", "ingredientseq": "10", "ingredientname": "Salt"},
1200     {"@type": "Ingredient", "ingredientseq": "11", "ingredientname": "Tripotassium Phosphate"},
1201     {"@type": "Ingredient", "ingredientseq": "12", "ingredientname": "Sunflower Oil"}

```

```

1201 {"@type": "Ingredient", "ingredientseq": "13", "ingredientname": "Colours: Caramel, Annatto,
1202 Carotene"},
1203 {"@type": "Ingredient", "ingredientseq": "14", "ingredientname": "Antioxidant: Tocopherals"},
1204 {"@type": "Ingredient", "ingredientseq": "15", "ingredientname": "Vitamin C"},
1205 {"@type": "Ingredient", "ingredientseq": "16", "ingredientname": "Niacin"},
1206 {"@type": "Ingredient", "ingredientseq": "17", "ingredientname": "Pantothenic Acid"},
1207 {"@type": "Ingredient", "ingredientseq": "18", "ingredientname": "Thiamin (B1)"},
1208 {"@type": "Ingredient", "ingredientseq": "19", "ingredientname": "Vitamin B6"},
1209 {"@type": "Ingredient", "ingredientseq": "20", "ingredientname": "Riboflavin (B2)"},
1210 {"@type": "Ingredient", "ingredientseq": "21", "ingredientname": "Folic Acid (Folacin)"},
1211 {"@type": "Ingredient", "ingredientseq": "22", "ingredientname": "Vitamin B12"},
1212 {"@type": "Ingredient", "ingredientseq": "23", "ingredientname": "Calcium Carbonate"},
1213 {"@type": "Ingredient", "ingredientseq": "24", "ingredientname": "Iron"}
1214 ]},
1215   "nutrientBasisQuantity": {"@type": "Measurement", "value": "100", "unit": "GRM"},
1216   "energyPerNutrientBasis":
1217   [{"@type": "Measurement", "value": "1615", "unit": "KJO"}, {"@type": "Measurement", "value": "382", "unit
1218   ": "E14"}],
1219   "proteinPerNutrientBasis": {"@type": "Measurement", "value": "8.6", "unit": "GRM"},
1220   "carbohydratesPerNutrientBasis": {"@type": "Measurement", "value": "74.3", "unit": "GRM"},
1221   "sugarsPerNutrientBasis": {"@type": "Measurement", "value": "21.4", "unit": "GRM"},
1222   "fatPerNutrientBasis": {"@type": "Measurement", "value": "4.0", "unit": "GRM"},
1223   "saturatedFatPerNutrientBasis": {"@type": "Measurement", "value": "1.0", "unit": "GRM"},
1224   "fibrePerNutrientBasis": {"@type": "Measurement", "value": "7.1", "unit": "GRM"},
1225   "sodiumPerNutrientBasis": {"@type": "Measurement", "value": "0.41", "unit": "GRM"},
1226   "saltPerNutrientBasis": {"@type": "Measurement", "value": "1.04", "unit": "GRM"},
1227   "vitaminCPerNutrientBasis": {"@type": "Measurement", "value": "71.0", "unit": "MGM", "dv": "89"},
1228   "thiaminPerNutrientBasis": {"@type": "Measurement", "value": "1.24", "unit": "MGM", "dv": "113"},
1229   "riboflavinPerNutrientBasis": {"@type": "Measurement", "value": "1.10", "unit": "MGM", "dv": "79"},
1230   "niacinPerNutrientBasis": {"@type": "Measurement", "value": "14.0", "unit": "MGM", "dv": "88"},
1231   "vitaminB6PerNutrientBasis": {"@type": "Measurement", "value": "1.20", "unit": "MGM", "dv": "86"},
1232   "folicAcidPerNutrientBasis": {"@type": "Measurement", "value": "200", "unit": "MC", "dv": "100"},
1233   "vitaminB12PerNutrientBasis": {"@type": "Measurement", "value": "1.90", "unit": "MC", "dv": "76"},
1234
1235   "pantothenicAcidPerNutrientBasis": {"@type": "Measurement", "value": "4.40", "unit": "MGM", "dv": "73"}
1236
1237   "calciumPerNutrientBasis": {"@type": "Measurement", "value": "460", "unit": "MGM", "dv": "58"},
1238   "ironPerNutrientBasis": {"@type": "Measurement", "value": "14.7", "unit": "MGM", "dv": "105"}
1239 }
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256

```

### Explanation:

The context section (shaded orange) references three namespaces – the GS1 vocabulary, the schema.org vocabulary and XSD (XML Schema Definition). (XSD is used for standard data types such as `xsd:float` and `xsd:integer`). The RDF namespace is implicitly included through the JSON-LD `@type` keyword, which maps to `rdf:type`.

Some basic fields such as the description of the offer or the trade item (product), the `gtin13` property, image and price information are mapped to terms from the schema.org vocabulary.

Specialised terms specific to food and beverage products are mapped to terms from the `gs1` vocabulary.

Some of these specialised terms for food product ingredients or nutritional information do not take simple string values but instead take complex data values such as a `gs1:NutritionMeasurementType` (which can be used to express a quantity, a unit of measure and percentage of the recommended daily intake of a nutrient as recommended by authorities of the target market) – or a `gs1:FoodAndBeverageIngredientDetail` (which can accept an ingredient sequence number, ingredient name and ingredient as a percentage of the total composition of the product).



### Note: *About properties with multiple values, lists, sequences etc.*

Another important point to note is that unlike RDF Turtle or N-Triples, in JSON-LD, the name of each property or predicate **should appear only once** in the data block. There may be situations where in RDF triples we might write several triples each containing the same property or predicate, perhaps using blank nodes if the value is not a simple data type. When we want to express these in JSON-LD, we must write the name of the property or predicate

1264 **once only** – and use a list for the sets of values corresponding to that property. In the  
 1265 example above, we can see examples of lists in JSON-LD (enclosed in square brackets) for  
 1266 the properties 'hasIngredientDetail' and 'energyPerNutrientBasis'. Lists are used in these  
 1267 examples to allow for multiple ingredients and for two different energy units, respectively.

### 1268 3.7 Procedure for serving a block of JSON-LD via an existing web page, using 1269 embedding – one product per page

1270 The publisher of a web page includes a block of JSON-LD that describes the single product appearing  
 1271 on that page.

#### 1272 3.7.1 Pre-Requisite

1273 Section 3.3, 3.4, 3.5, or 3.6.

#### 1274 3.7.2 When Would I Use This?

1275 Use this procedure when there is a single product described on a given web page, and you want to  
 1276 include structured data about that product.

#### 1277 3.7.3 How To?

1278 You add JSON-LD to a web page simply by putting it inside of a `<script>` tag that specifies an  
 1279 Internet Media type of `application/ld+json`. This can be inserted within the `<head>` section of  
 1280 your page, like this:

```
1281 <html>
1282   <head>
1283     <script type="application/ld+json">
1284       (JSON-LD block goes here)
1285     </script>
1286     ... (rest of head section)
1287   </head>
1288   <body>
1289     ... (visible part of the web page)
1290   </body>
1291 </html>
```

1292 Alternatively, the JSON-LD can be added as the last child element within the `<body>` section of your  
 1293 page, like this:

```
1294 <html>
1295   <head>
1296     ... (rest of head section)
1297   </head>
1298   <body>
1299     ... (visible part of the web page)
1300     <script type="application/ld+json">
1301       (JSON-LD block goes here)
1302     </script>
1303   </body>
1304 </html>
```

1305 Either way, the JSON-LD block will be understood as referring to the entire page.

1306 This illustrates the chief advantage of JSON-LD compared to RDFa or other means of embedding  
 1307 structured data in a web page: unlike inline formats such as RDFa and Microdata, JSON-LD is  
 1308 inserted in just one place in the page markup, well away from the visible content. This makes  
 1309 adding JSON-LD to a web page much easier and much less prone to error.

1310 It is important to note that when JSON-LD (or any other structured data format) is added to a web  
 1311 page, the semantic information in machine readable format must match the information in the  
 1312 human readable section. Failure to do so is considered abusive use by search engines, which could  
 1313 result in a lower rank for the web page or the page not being listed at all.



### 3.8 Procedure for serving a block of JSON-LD via an existing web page, using embedding – procedure for multiple products per page

The publisher of a web page includes a block of JSON-LD that describes multiple products appearing on that page.

#### 3.8.1 Pre-Requisite

Section 3.3, 3.4, 3.5, or 3.6.

#### 3.8.2 When Would I Use This?

Use this procedure when there are more than one single product described on a given web page, and you want to include structured data about each of those products.

#### 3.8.3 How To?

The procedure is almost the same as presented in Section 3.7, except that separate JSON-LD blocks, each within their own `<script>` tags, are included for each product. Each JSON-LD block corresponds to exactly one GTIN, and contains one subject for the Product GTIN and at most one subject for the Offer GTIN (the latter only being applicable for a retailer's web page).

As in Section 3.7, you add JSON-LD to a web page by putting it inside of a `<script>` tag that specifies an Internet Media type of `application/ld+json`. This can be inserted within the `<head>` section of your page, like this:

```

<html>
  <head>
    <script type="application/ld+json">
      (JSON-LD block for GTIN 1 goes here)
    </script>
    <script type="application/ld+json">
      (JSON-LD block for GTIN 2 goes here)
    </script>
    ... (and so forth for remaining GTINs)
    ... (rest of head section)
  </head>
  <body>
    ... (visible part of the web page)
  </body>
</html>
  
```

Alternatively, the JSON-LD blocks can be added as the last child element within the `<body>` section of your page, like this:

```

<html>
  <head>
    ... (rest of head section)
  </head>
  <body>
    ... (visible part of the web page)
    <script type="application/ld+json">
      (JSON-LD block for GTIN 1 goes here)
    </script>
    <script type="application/ld+json">
      (JSON-LD block for GTIN 2 goes here)
    </script>
    ... (and so forth for remaining GTINs)
  </body>
</html>
  
```

It is important to note that when JSON-LD (or any other structured data format) is added to a web page, the semantic information in machine readable format must match the information in the human readable section. Failure to do so is considered abusive use by search engines, which could result in a lower rank for the web page or the page not being listed at all.

1367 In the case of multiple products per web page, there should be exactly as many JSON-LD blocks as  
1368 there are GTINs in the human-readable portion of the page, and the JSON-LD blocks should appear  
1369 in the same order as the order the corresponding GTINs appear in the HTML markup for the human-  
1370 readable portion. To further establish the correspondence between the JSON-LD and the human-  
1371 readable portion, and because of policies regarding abuse as discussed above, all attributes in the  
1372 JSON-LD should match information presented in the human-readable HTML (a notable exception  
1373 being the GTIN itself, which might not appear in the human-readable section).

### 1374 **3.9 Procedure for serving a standalone block of JSON-LD in isolation via a** 1375 **webserver**

1376 The data publisher exposes web resources that return a JSON-LD representation of the resource  
1377 when dereferenced (as opposed to an HTML web page that embeds the JSON-LD).

#### 1378 **3.9.1 Pre-Requisite**

1379 Section 3.3, 3.4, 3.5, or 3.6.

#### 1380 **3.9.2 When Would I Use This?**

1381 Many modern front-end frameworks, such as [AngularJS](#), use JavaScript to manipulate the webpage  
1382 and asynchronously load JSON data from an API. Using JSON-LD rather than plain-old JSON allows  
1383 the use of shared identifiers for properties and the possibility of embedding links to other resources  
1384 into the JSON (regular JSON does not support the URI datatype). Using this approach to make the  
1385 data and external resource that can be referenced to makes it possible to share and re-use data  
1386 across different webpages without having to embed the same data into each and every page. This  
1387 can improve cacheability of resources and reduce the tidal wave effect whereby a small change can  
1388 result in many hundreds or thousands of HTML pages needing to be updated.

#### 1389 **3.9.3 How To?**

1390 JSON-LD data or context files can be served using a conventional webserver. However, it is  
1391 important to configure the webserver to specify the appropriate MIME type in the Header  
1392 information before it sends the JSON-LD file. The MIME type for JSON-LD is application/ld+json

1393 If using an Apache webserver, you can achieve this by modifying the .htaccess file in the same  
1394 directory as the JSON-LD files so that it includes the following lines:

```
1395 Header set Access-Control-Allow-Origin "*"
1396 AddType application/ld+json .jsonld
```

1397 The first line enables Cross-Origin Resource Sharing (CORS) [see <http://enable-cors.org/>], so that  
1398 javascript from other domains can access your JSON-LD files.

1399 The second line forces the webserver to indicate a MIME type of application/ld+json whenever  
1400 it serves a JSON-LD file, provided that the JSON-LD files are named with a .jsonld filename suffix.

### 1401 **3.10 Procedure for checking that structured data is correctly formatted**

1402 Once you have created JSON-LD for your product or product offering you will want to check that it is  
1403 formatted correctly so that it can be processed by any applications and apps that may wish to  
1404 consume it. This section explains the procedure and tools to help you achieve this.

#### 1405 **3.10.1 Pre-Requisite**

1406 Section 3.7 or 3.8.

#### 1407 **3.10.2 When Would I Use This?**

1408 Use one of the tools suggested to check that your data is syntactically correct and that it will be  
1409 interpreted in the manner expected.

### 3.10.3 How To?

The JSON-LD Playground tool at <http://json-ld.org/playground/index.html> can be used to check the JSON-LD you have generated.

You can use the JSON-LD contained at the following web address to see results.

<http://www.autoidlabs.org.uk/GS1Digital/Demos/GS1vocab/gS1JSON-LD-Demo.html>

Just view the page source and paste the JSON-LD block in to the JSON-LD Playground form. (You must exclude the enclosing `<script>` tags as these are not part of the JSON-LD block.) The tool will check and report upon any syntax errors (e.g., "JSON markup - SyntaxError: Unexpected token {"). You can also use the tool to view your content in a number of different formats to help ensure the intended meaning of your JSON-LD.

Another useful tool can be found at <http://linter.structured-data.org/>. By pasting your page URL or uploading your page content you can use this tool to get a visual confirmation of the structured data in your page.

## 3.11 Procedure for accessing structured data in a JSON-LD block using JavaScript within the same web page

The publisher of a web page wishes to exploit the embedded JSON-LD content for other purposes within the web page itself.

### 3.11.1 Pre-Requisite

Section 3.7 or 3.8.

### 3.11.2 When Would I Use This?

Modern web pages do many data manipulations in Javascript. Embedding the information once in JSON-LD and then using it from Javascript can be useful for the following use-cases:

- Building rich user interfaces: instead of having duplicate content in the HTML portion of the web page and the JSON-LD, the web page includes just the JSON-LD and Javascript code that reads the JSON-LD to populate the user-facing content (via the [DOM](#)). This can offer benefits including:
  - Pagination of long content.
  - Translation of attributes into icons or procedurally generated graphics.
  - Displaying information in various languages without requiring a page-reload.
- Populating tracking data. Systems like Google Analytics take their data in Javascript structures, which can be populated by reading the JSON-LD data. This enables tracking by the various attributes that are added.

### 3.11.3 How To?

View the instructions at:

<http://www.autoidlabs.org.uk/GS1Digital/Demos/GS1vocab/gS1JSON-LD-with-JavaScript.html>

This page uses the same block of JSON-LD as the previous example and explains how JavaScript can access the data.

Note that JavaScript is not natively aware of JSON-LD, which means that it ignores the `@context` header and does not expand local keys or `prefix:name` constructs to full URIs, nor is it aware of `@type` or `@language`.

It is possible to access the data from JSON-LD – but not always via the dot (.) notation familiar in JSON.

## A Appendix: Technical background for deploying Linked Data about products

### What is the semantic web?

According to the World Wide Web Consortium (W3C) [<http://www.w3.org/2001/sw/>], The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework ([RDF](#)). It refers to a collection of technologies that can be used to transform the web of documents (e.g. web pages) into a global web of interlinked interoperable data that is machine-interpretable because the meaning of each data relationship is explicitly stated – and because the semantic web uses HTTP URIs (e.g. web addresses), it is possible to access related data as well as definitions of properties and attributes, multi-lingual names and descriptions simply via a regular HTTP web request.

### What is Linked Data?

According to [LinkedData.org](#), Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods. Linked Data is sometimes considered as being either synonymous with the Semantic Web or being a subset of it. Linked Data can be provided and retrieved via web requests, either as standalone data – or embedded within regular web pages, as additional semantic markup of the facts contained within the page, which are accessible without ambiguities to software including search engines, smartphone apps etc.

See Section 1.3 for a brief introduction to Linked Data.

### How is the data structured?

In Semantic Web / Linked Data technology, we don't think of the data as being structured in well-defined tables with rows and columns as in a relational database. The Semantic Web uses a simpler data structure in which facts, factual claims or data relationships are expressed as a directed graph of data. You can think of a graph of data as being very similar to a mind-map. A mind-map uses circles, ovals or rectangles to represent 'things' and arrows connecting these 'things' to represent the relationships between them.

In order to convert this 'mind-map' or 'graph' of data relationships from a pictorial representation to a format that can be processed by computer software, we usually represent each arrow on the mind-map as a triple that connects a subject (the 'thing' being described, at the start of the arrow) to an object (another 'thing' that appears at the end of the arrow). The arrow itself corresponds to a specific named property or predicate, which represents the data relationship that connects the subject to the object.

In this way, even very complicated data structures can be collapsed to essentially a 3-column table of 'triples'. This is the essence of Resource Description Framework (RDF) - a W3C technical standard that is at the foundation of the Semantic Web technology stack. There are a number of ways in which such RDF data can be exchanged or communicated. These include inline markup formats such as RDFa (RDF in annotations) or Microdata, and block-oriented formats such as JSON-LD (JavaScript Object Notation for Linked Data).

### RDF Triples - Subject, Predicate, Object

As mentioned above, RDF enables us to write simple logical sentences to express factual assertions (e.g. a product has a specific weight) in a way that computer software can use, in order to 'understand' the meaning and potentially even generate some new facts ('inferencing') from existing facts that are explicitly stated, either by making use of precise logical assertions defined in an ontology - or by using user-defined rules in a query language such as [SPARQL](#) (see the SPARQL CONSTRUCT mechanism).

### Use URIs instead of words

When we write facts in RDF, instead of using simple text string or words to identify things and relationships, we use HTTP URIs where possible. The exception to this is for simple literal values

1503 such as numbers, dates or when we want to use a text string to provide a label, description or  
1504 definition for something.

1505 The advantage of using HTTP URIs is that they are globally unambiguous and can be created in a  
1506 very decentralised manner. Anybody can create HTTP URIs by first obtaining an Internet domain  
1507 name (or using one you already have, such as the domain of your website) and then using this in  
1508 the “authority” portion of a URI. Because the domain name is unique, the HTTP URIs you create will  
1509 not accidentally clash with HTTP URIs created by others.

1510 While an HTTP URI does not have to be an actual web address to be usable in Linked Data, in  
1511 practice it is helpful if an HTTP URI appearing in Linked Data can actually be used to make a web  
1512 request (“dereferenced”) that returns some useful information about the thing the HTTP URI  
1513 represents. That way, if you want to find more information about a thing that is identified by an  
1514 HTTP URI, or find the definition of a property or predicate identified by an HTTP URI, you can try  
1515 making a web request for it.

1516 Although you might be redirected to an alternative URI that delivers the information, you can  
1517 typically expect to receive some useful information as a result of such a web request. This can  
1518 include multi-lingual labels, descriptions and definitions, as well as links to other related things (also  
1519 identified by HTTP URIs), and where the relationship of each link indicates a specific property or  
1520 relationship.

### 1521 **What are vocabularies and what are ontologies?**

1522 Vocabularies and ontologies provide lists of concepts, classes (types of thing) and properties or  
1523 predicates (relationships, attributes), together with their definitions. Examples of vocabularies  
1524 include schema.org, GoodRelations, vCard, [Friend Of A Friend \(FOAF\)](#), [Dublin Core](#) and the new GS1  
1525 vocabulary that is being developed in the GS1 SmartSearch work group.

1526 Ontologies go a step further than vocabularies because they typically also include some very precise  
1527 logical statements about the classes and properties that allow computer software to do some  
1528 automated logical reasoning. For example, an ontology can make use of W3C technical standards  
1529 such as [RDF Schema \(RDFS\)](#) and the [Web Ontology Language \(OWL\)](#) to make such statements. For  
1530 example, we can define a father as being a sub-property of a parent. We can say that  
1531 ‘hasDateOfBirth’ is only allowed to have one value for any specified thing. We can say that  
1532 ‘hasAncestor’ is transitive, which means that if computer software sees a ‘hasAncestor’ relationship  
1533 between you and one of your parents, and between one of your parents and one of your  
1534 grandparents, it can use the transitive property to reason or infer that there is also a ‘hasAncestor’  
1535 property between you and your grandparents and your great-grandparents, etc. Classes can also be  
1536 marked as being mutually disjoint (e.g. letters and digits are both subclasses of characters but have  
1537 no overlapping members).

### 1538 **How is Linked Data published and made available by the publisher?**

1539 Linked Data can be embedded within existing web pages, either as inline markup using RDFa or  
1540 Microdata or as a block of structured data, using JSON-LD markup. See Sections 3.7 and 3.8 for  
1541 examples of how to embed a block of JSON-LD within an existing web page.

1542 Linked Data can also be served directly, using a web server, provided that the appropriate [Internet  
1543 Media Type \(MIME\)](#) headers are emitted before the data is served. See Section 3.9 for guidance  
1544 about serving a block of JSON-LD directly, without embedding in a web page.

1545 Another approach for serving Linked Data on the web is via the use of [SPARQL](#) endpoints. These  
1546 provide an online query interface using the W3C SPARQL query protocol standard. In this situation,  
1547 data need not be provided as a complete dump of Linked Data; instead the SPARQL endpoint can  
1548 respond to SPARQL queries, perform the appropriate matches on its data graph and return the  
1549 results on demand.

### 1550 **How can consumers of Linked Data request a particular format (e.g. JSON-LD)?**

1551 Software that wishes to retrieve Linked Data can use HTTP Content Negotiation to request the  
1552 preferred format, by specifying a sequence of MIME types and associated preferences.

1553 If the Linked Data is not available in the requested format, a number of tools exist, which can  
1554 convert Linked Data from one format into another format, without loss of information or meaning.

1555 Such tools include <http://rdf-translator.appspot.com/> and software libraries in various programming  
1556 languages, e.g. <http://rdflib.net>.

1557 Consumers of Linked Data can also make use of [SPARQL](#) endpoints to request Linked Data that  
1558 matches their SPARQL queries.

### 1559 **Why are we advocating the use of JSON-LD?**

1560 In this document, we recommend using JSON-LD because:

- 1561 ■ Use of JSON-LD requires less detailed analysis / knowledge of the structure and layout of the  
1562 human-readable web page, such as the nesting of <div> elements within the page.
- 1563 ■ The block of JSON-LD is largely decoupled from the rest of the web page and as such, it is  
1564 possible to modify the visible layout of the page without needing to make changes to the JSON-  
1565 LD block, so long as the structured data in the JSON-LD block still accurately corresponds to the  
1566 human-readable information in the page.
- 1567 ■ It is much easier and more scalable for GS1 work group to provide some worked examples of  
1568 JSON-LD markup in a way that can easily be adapted by various user companies and their  
1569 solution providers, rather than trying to develop individual customised examples using inline  
1570 markup, for which users might have more difficulty in relating to the necessary modifications to  
1571 their existing web pages.
- 1572 ■ JSON-LD is considered to be less brittle than inline markup such as RDFa or Microdata. For  
1573 example, using RDFa or Microdata, an image of a product appearing within the web page might  
1574 be annotated with a property such as `http://schema.org/image` using the following markup  
1575 example.

```
1576 <div about="http://example.com/id/gtin/05011476100885" typeof="http://schema.org/Product">  
1577     
1578 </div>
```

1579 However, if someone then wraps a hyperlink around an image and changes the markup to be as  
1580 shown below, the interpretation of the RDFa markup changes.

```
1581 <div about="http://example.com/id/gtin/05011476100885" typeof="http://schema.org/Product">  
1582   <a href="promotion.html"></a>  
1584 </div>
```

1585 In the example above, the image then becomes an image for the hyperlinked promotion page,  
1586 instead of an image of the product, because the href value of the hyperlink overrides the  
1587 subject specified in the about attribute of the parent <div> container.

1588 JSON-LD is not susceptible to this brittleness caused by the addition of hyperlinks.

### 1589 **Which frameworks are available for serving Linked Data?**

1590 Linked Data can be served using an existing web server or it can be served using a dedicated Linked  
1591 Data framework. These include commercial implementations as well as free or open source  
1592 implementations, including those based on the [W3C Linked Data Platform \(LDP\)](#). Further information  
1593 about implementations is available at [http://www.w3.org/wiki/LDP\\_Implementations](http://www.w3.org/wiki/LDP_Implementations)

### 1594 **What is Product Master Data?**

1595 Product master data typically consists of the specifications or attributes of a product that are stable  
1596 over time and apply to all instances of that product class, i.e. every individual product package  
1597 having that same GTIN barcode number can be expected to share the same characteristics that are  
1598 described through master data. Master data can include information about material composition or  
1599 ingredients, nutritional information, power consumption and technical specifications, as well as  
1600 information about accreditations (e.g. environmental, ethical or dietary claims), as well as  
1601 information about allergens that might be contained in the product - or other consumer safety  
1602 information.

1603 Typically master data is created by the brand owner or manufacturer and shared with the retailer of  
1604 the product, so that consumers have access to this information even when they buy products  
1605 online. For some kinds of product master data involving accreditations, independent accreditation

1606 agencies might also play a role in contributing their assertions about the product, which can be  
 1607 embedded or referenced from the product master data.

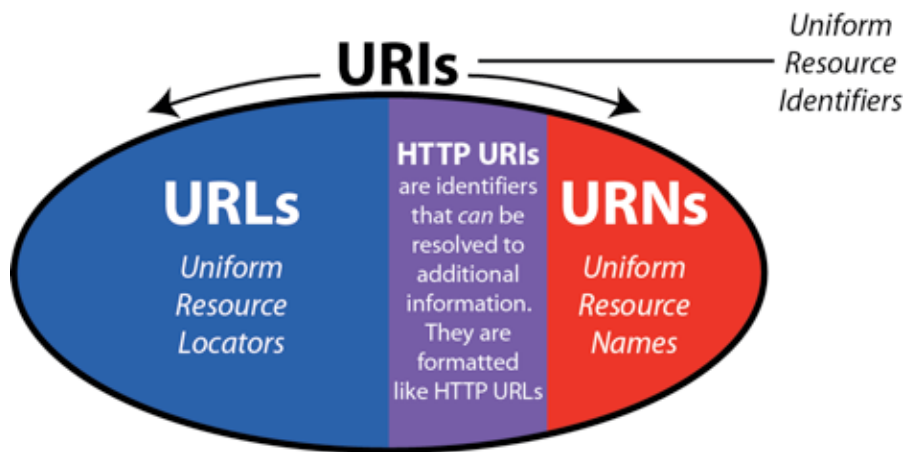
1608 In the GS1 SmartSearch group, our initial focus is on the use of Linked Data technology to enable  
 1609 brand owners, manufacturers and retailers to publish product master data openly on the web, so  
 1610 that it is available for use by search engines, smartphone apps. We expect that the result of this will  
 1611 be to enable enhanced search listings for products and services, enable new business-to-consumer  
 1612 and consumer-to-business interactions, help consumers find the products they are really looking for,  
 1613 and help business understand how consumers are searching for products and the selection criteria  
 1614 that matter to them in their decisions.

1615 **Is the semantic web reliant on the Search Engine Optimisation (SEO) of websites?**

1616 No. Linked Data can be served within websites, using either inline markup (e.g. Microdata or RDFa)  
 1617 or a single block of structured data (e.g. JSON-LD) but it is also possible to serve the structured  
 1618 data directly using web server technology or other dedicated Linked Data mechanisms (such as RDF  
 1619 triple stores with SPARQL endpoints), so that search engines, smartphone apps and other software  
 1620 can make a request just for the structured data, without requesting the web page. There are some  
 1621 existing conventions and best practices for how to do this.

1622 Having said that, some websites still make use of JavaScript or Flash for the navigation within the  
 1623 website. This can mean that search engines and other software is unable to interpret the JavaScript  
 1624 or Flash and therefore unable to discover all of the pages in the website that might contain  
 1625 structured data. It is therefore recommended to check whether someone can still navigate through  
 1626 the website even when their web browser has JavaScript and Flash switched off. If this is not  
 1627 possible, then it may be better to re-think how the navigation is done and to use more modern  
 1628 approaches to navigation toolbars and sidebars using HTML5 and CSS, rather than relying upon  
 1629 using JavaScript to do image rollovers and highlighting. If the HTML source code for your website  
 1630 includes many JavaScript 'onClick' handlers, then it might be a good idea to think about replacing  
 1631 these with regular hyperlinks (e.g. `<a href="...">`) since these will be more accessible to search  
 1632 engine crawlers and other software, without needing to understand any of the customised  
 1633 JavaScript code that was written for the individual website.

1634 **What is the difference between a URI and a URL?**



1635  
 1636  
 1637 URIs are Uniform Resource Identifiers.

1638 URLs are Uniform Resource Locators (e.g. web addresses) and are used for retrieving information.

1639 URNs are Uniform Resource Names (e.g. EPCs are canonically expressed as Pure Identity URIs using  
 1640 URN notation) and are used for globally uniquely naming things - but they have no obvious  
 1641 mechanism for retrieving information.

1642 All URLs are URIs. All URNs are URIs. URIs are the 'union' of URLs and URNs.

1643

### Why do we use URIs?

1644

Linked Data / Semantic Web technology makes extensive use of HTTP URIs, which can function either as names (like URNs) or as locators (like URLs).

1645

1646

You cannot always tell from looking at an HTTP URI whether it is serving the role of a name or a locator. However, you can make an HTTP GET request for that HTTP URI.

1647

1648

If it is serving a role as a name for a real-world thing or place, then that real-world thing or place cannot be delivered to you via the web, so the web server that handles that HTTP URI does the next best thing and returns an HTTP 303 'See Other' response code, together with a corresponding HTTP URI that works as a locator. Your web browser will then make a second request for that locator HTTP URI and obtain an information representation (e.g. web page) of the real-world thing or place.

1649

1650

1651

1652

1653

Linked Data usually consists of RDF triples consisting of a Subject, Predicate and an Object. The HTTP URIs that work like names can be used in the Subject, Predicate or Object positions of RDF triples. The HTTP URIs that work like locators are used to retrieve a collection of RDF triples. So for example,

1654

1655

1656

1657

- `http://dbpedia.org/resource/Brussels`

1658

is an HTTP URI that serves as a name and is used in the subject of many RDF triples at DBpedia for facts about Brussels, capital of Belgium.

1659

1660

- `http://dbpedia.org/page/Brussels`

1661

is an HTTP URI that serves as a locator and is used to retrieve a web page containing a collection of those RDF triples from DBpedia.

1662

1663

When you type `http://dbpedia.org/resource/Brussels` into your web browser, DBpedia returns an HTTP 303 'See Other' response code and suggests that your web browser request `http://dbpedia.org/page/Brussels` instead. By doing so, DBpedia is saying "Sorry - I can't deliver Brussels to you via the web - try requesting this page of information about Brussels instead."

1664

1665

1666

1667

### Does the URI replace GTIN?

1668

No. We expect that for many years, the Global Trade Item Number (GTIN) will provide the primary key for identifying products at Point of Sale systems or accessing information via GDSN. However, the GTIN is simply a numeric string. Unlike a URI, it does not indicate any obvious or trivial mechanism for retrieving data about the object identified by a GTIN. In contrast, an HTTP URI looks like a web address or URL and can be configured to behave like one, returning data in response to a web request. HTTP URIs complement GTINs.

1669

1670

1671

1672

1673

1674

### How do we develop a structured data mark-up template?

1675

From our experience, it will be best to do this using a single block of JSON-LD either within the `<head>` block of the HTML page or as the last child element within the `<body>` block of the HTML page, instead of using inline markup such as RDFa or Microdata.

1676

1677

1678

The reason is that it is very easy for RDFa markup to become broken. For example, an HTML image tag `<img>` might contain an RDFa attribute such as

1679

1680

```
property="schema:image"
```

1681

and this is understood as meaning that this image is being said to be representative of something in an enclosing block of HTML that was identified using an RDFa attribute such as

1682

1683

```
about="http://nestle.com/id/05011476100885"
```

1684

That all works fine until somebody else puts a hyperlink around the image, perhaps to link to a promotion or even open a pop-up window with additional views of the product.

1685

1686

When they do that, if they are not very careful to have asserted an explicit attribute of `about="http://nestle.com/id/05011476100885"` within the `<img>` tag, the image will be considered to be a `schema:image` of the new hyperlink, rather than a `schema:image` of the product.

1687

1688

1689

It's also much more tricky to do RDFa correctly in the first place because the structure of the web page and its various nested `<div>` blocks needs to be carefully considered.

1690



1691 It is much easier for GS1 to provide a JSON-LD template (perhaps one for food products, one for  
1692 textiles, etc.) that can then simply be populated with the actual values (e.g. weight, colour, size,  
1693 nutritional info, etc.) without either GS1 nor the retailer or brand owner being concerned about  
1694 where that information appears within the web page.

1695 However, it is important that the information that appears in the JSON-LD block is consistent with  
1696 the information appearing in the web page.

1697 At present, Google are still in the process of fully recognising JSON-LD as acceptable markup, but  
1698 we have had discussions with them to explain why JSON-LD will be much more practical to deploy  
1699 than inline markup such as RDFa or Microdata. We are also encouraging them to improve their  
1700 support for JSON-LD in their Google Structured Data Testing Tool at  
1701 <http://www.google.com/webmasters/tools/richsnippets> although other tools are available for  
1702 testing, including <http://linter.structured-data.org/>

### 1703 **What is JSON?**

1704 JSON is an abbreviation for JavaScript Object Notation. JSON is a compact way of exchanging  
1705 structured data objects (lists, key-value pairs). JSON is already used in websites and smartphone  
1706 apps for exchanging fragments of data between the web browser or app and the backend server  
1707 (e.g. for auto-suggest, auto-complete etc.) without reloading the page.

### 1708 **What is JSON-LD?**

1709 JSON-LD is an abbreviation for JavaScript Object Notation for **Linked Data**. JSON-LD provides a  
1710 way to make pieces of JSON interoperable with each other, by mapping locally-defined keys  
1711 (properties) to global URIs. JSON-LD also provides a way to include structured data in a web page  
1712 as a single block. JSON-LD is less fiddly and less brittle than inline markup such as RDFa or  
1713 Microdata. Unlike the latter, JSON-LD appears within a `<script>` element within the `<head>` or  
1714 `<body>` element of a web page, but does not require inline annotations interspersed with the visible  
1715 content.

### 1716 **What are JSON-LD Templates?**

1717 A JSON-LD Template is a single block of machine-interpretable structured data that can be placed  
1718 within the `<head>` or `<body>` element of a web page or served as standalone structured data.

### 1719 **Where can I learn more about JSON-LD?**

1720 For more information about JSON-LD, please see:

- 1721 ■ <http://json-ld.org/> JSON-LD site & playground
- 1722 ■ <http://www.w3.org/TR/json-ld/> W3C standard
- 1723 ■ <http://youtu.be/vioCbTo3C-4> video intro
- 1724 ■ <http://www.slideshare.net/gkellogg1/json-for-linked-data>

### 1725 **What do we mean by a 'trusted source of data'?**

1726 Trusted source of data refers to techniques that provide an assurance that the data was provided by  
1727 an organisation that had the authority to provide that data, such as the brand owner or  
1728 manufacturer of a product. This is in contrast with data from untrusted or non-authoritative sources,  
1729 such as crowd-sourced data about products. Because most brand owners who apply barcodes to  
1730 products lease a GS1 Company Prefix from a national GS1 member organisation (such as GS1 UK),  
1731 GS1 is in a unique position to know which brand owner is associated with a given product barcode  
1732 (GTIN – Global Trade Item Number) – and to be able to confirm whether data about a product came  
1733 from a trusted source, typically the brand owner.

### 1734 **What role does standardisation have to play in the semantic web?**

1735 Standardisation plays a critical role in promoting interoperability and reducing ambiguities and  
1736 incompatibilities in the web of data. The World Wide Web Consortium (W3C) has overseen the  
1737 development of many of the fundamental technical standards that provide the framework for  
1738 exchanging structured data using semantic technologies. Most of these standards are supported by  
1739 commercial and free or open source implementations. For web vocabularies, there are some

1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751

standardised ontologies such as Dublin Core [<http://dublincore.org> - see also IETF RFC 5013, <http://tools.ietf.org/html/rfc5013>, ISO 15836:2009 ], as well as web vocabularies (such as [schema.org](http://schema.org)) that were initially developed outside of a standards process, but which are now being further developed and extended within a collaborative community process, with involvement from the W3C. [<http://www.w3.org/wiki/WebSchemas>]. For over 40 years, GS1 has brought together a community of brand owners, manufacturers, distributors and retailers in a number of industry sectors to work together on common standards for exchanging information within supply chains. The GS1 community has already developed extensive detailed data models and data dictionaries for describing products, services and organisations and the GS1 Digital initiative and GS1 SmartSearch work group is now making these available as a web vocabulary for use with Linked Data technologies.