



The Global Language of Business

---

# Technical guidance for GS1 SmartSearch pilots

February 2016

## 1. Introduction

### 1.1. Purpose of this document

This paper provides guidance about how to test the effectiveness of GS1 SmartSearch on retailer websites.

The focus of this document is to:

- Document three technical methods for implementing GS1 SmartSearch on e-commerce platforms.
- Suggest key performance indicators for testing the effectiveness of GS1 SmartSearch with retailers.

This guidance originates from ongoing research and prototyping work performed by the Digital team at GS1 UK from December 2015.

This document should be read alongside the following resources

- GS1 SmartSearch Implementation Guideline (<http://www.gs1.org/gs1-smartsearch/guideline/gtin-web-implementation-guideline>)
- GS1 SmartSearch Vocabulary standard (<http://www.gs1.org/gs1-smartsearch/latest>)
- GS1 SmartSearch starter kit (<http://mozone.gs1.org/gs1-smartsearch>)

### 1.2. Who is this document aimed at?

Representatives from GS1 member organisations, who have agreed, or are in the process of negotiating, a pilot of GS1 SmartSearch with a retailer on their e-commerce platform.

## 2. Technical Overview

### 2.1. What is GS1 SmartSearch?

GS1 SmartSearch is GS1 standard that uses a text-based data format to describe a product comprehensively using structured data. This data format uses a syntax standard called *JavaScript Object Notation* (JSON). In this syntax, data becomes machine-readable; that is, it can be understood by computer applications such as search engine 'spiders' that surf the web reading and indexing web pages.

In addition, GS1 SmartSearch extends schema.org data tagging notation by using an in-depth vocabulary to describe each product in great detail. A vocabulary is an agreed set of product attribute names that should be used, where relevant, in every product. There are specialist vocabularies for food and drink products, and for apparel (clothing and fashion) as part of this GS1 standard. There is also an automated mechanism for describing all products, mainly general merchandise, outside of these two categories.

## 2.2. What is JSON Structured Data, and why is it useful?

JSON structured data is a format that uses *name/value pairs* to describe product attributes. For example:

```
``colour``: ``purple``
```

When testing the effectiveness of GS1 SmartSearch, the following three capabilities have been tested through prototyping work conducted by the GS1 UK Digital team:

- **Search engine indexing applications can be certain about the data.** Compare this format to standard HTML, which is a *markup language* that tells a web browser how to display a web page's words and images for humans to view. In the example above, the words "colour" and its value "purple" may appear close to each other on the web page. In HTML language, however, this name and value may be in completely different sections of the document describing the page. A search engine indexer may not be sure that "colour" and "purple" are even connected to each other in any way. Using structured data as shown above allows the indexing application to be certain about this connection.
- **Web developers can use JavaScript to read the data and display it in the web page.** This technique allows the data to be separated from the page HTML markup language rather than being mixed up together and duplicated. In the example of GS1 SmartSearch, the data can be imported from another location on the web and the JavaScript programming language used to create the finished page. JSON is the native data format of JavaScript.
- **JSON is a lightweight and compact data format.** It can store data with less syntax overhead and in fewer data bytes than most other popular data formats such as *eXtensible Markup Language* (XML). It is universally readable data format in all modern web browsers.

## 3. Implementing GS1 SmartSearch for Pilot Purposes

### 3.1. Overview of Approach to Piloting

GS1 SmartSearch is a new standard that consists of ongoing work to enable a product in the digital world to be described as fully as if you were in the presence of the physical product.

We wish to describe every product in the world that has a GTIN identification, and ensure that the vocabulary is largely future-proof. In the case of grocery products that have a known set of descriptive and legal terms, the work by GS1 is both mature and well documented. Conversely, the world of all general merchandise products will have a diverse descriptive vocabulary – some of which are general and common (such as weight and colour) and some of which are specialist to a particular category (such as spin-speed for washing machines and the focal-length of an SLR camera lens).

For the purposes of piloting, we recommend that retailers use the product data descriptions used by their current e-commerce application and displayed already in their particular products' description pages. This recommendation makes it easier, simpler, and low risk.

However there may be instances when a pilot involves a retailer and a brand working together, and the brand wishes to author GS1 SmartSearch structured data for their products. In this case we have a recommended approach that can be included in early piloting and documented as "Method 3" further along this document.

### 3.2. Technical Assumptions for a Pilot

This section describes a set of technical assumptions for a pilot that are aligned with the workflows for each method described in section 3.3.

#### 3.2.1. Consumers using the website generally have JavaScript enabled.

JavaScript is a programming language used by web browsers to enhance the customer's experience of a web page by making it more interactive. Most e-commerce websites require JavaScript to be enabled in the consumers' web browsers. If a retailer insists that their site must work for *JavaScript-not-enabled* web browsers than only Method 1 can be piloted.

#### 3.2.2. Location of GS1 SmartSearch in a web page

The GS1 SmartSearch JSON structured data must be written between a <SCRIPT> and </SCRIPT> tag which can be anywhere in the <HEAD> or <BODY> part of the HTML web page. Here is a simple example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Product name here</title>
<script type="application/ld+json" id="datablock1">
{
  (GS1 SmartSearch structured data here)
}
</script>
</head>
<body>
```



```
(Usual Body HTML tags and content here)
</body>
</html>
```

Of course, a web page can have other `<script></script>` tags without effecting the GS1 SmartSearch data. Often pages will have JavaScript, CSS styles and other additional tags.

### 3.3. Build Methods 1, 2, and 3

There are three different ways to build GS1 SmartSearch pages. These methods have resulted from ongoing research by GS1 UK to assess how search engines react to each implementation and what level of change is required to a retailer's e-commerce application.

The methods are:

1. **Embedded JSON and 'hard-coded' HTML both describe the product.** Embed GS1 SmartSearch inside a product details web page alongside the existing HTML describing the product. This is the simplest, easiest and lowest risk method, but product information is sourced from the retailer's own data source, and the information is duplicated.
2. **Embedded JSON describes the product and JavaScript writes the HTML.** Embed GS1 SmartSearch inside a product details web page, then use a JavaScript routine to read the JSON structured data and use it to populate 'empty' places in the HTML section where product information should be. This method is a little more complex but has the advantage of stopping duplication. It still uses the retailer's own data source.
3. **Imported JSON describes the product and JavaScript writes the HTML.** Import a GS1 SmartSearch document from elsewhere on the web, then use a JavaScript routine to read the JSON structured data and use it to populate 'empty' places in the HTML section where product information should be. This is the highest risk method because the web server serving the GS1 SmartSearch document might be unavailable. However, the advantage is that the document could be held at the GTIN licensee's (brand owner's) web domain, or on the servers of a Product Library or GDSN service, giving them ownership of (and liability for) product data quality.

These build methods are now described in detail.

### 3.3.1. Method 1: Embed GS1 SmartSearch alongside existing HTML product information

Implementing this method involves adjusting an existing web page template for the product. Web templates are web pages with placeholder variables, which are replaced by actual product data when e-commerce web server renders the page and sends it to the customer's web browser. A placeholder variable is text that represents a product attribute value. The format of a placeholder depends on the web server or e-commerce application but often take a form similar to "\$attribute name\$". For example if web template has the line:

```
<p>${productname$ price is &pound;${productprice$}</p>
```

...then, when the product page containing a particular product (for example, a Kettle costing £23.50) is sent to the customer's web browser, it will look like this:

```
<p>Kettle price is &pound;23.50</p>
```

..and on the web page the customer will see:

**Kettle price is £23.50**

Each placeholder variable can be used more than once, for example:

```
<p>${productname$ price is &pound;${productprice$, what a great price for a  
>${productname$!</p>
```

**Kettle price is £23.50, what a great price for a Kettle!**

We can take advantage of the ability to use placeholder variables multiple times by adding them to JSON structured data in GS1 SmartSearch format. Here is an example in this snippet from the full syntax, which is contained within a <SCRIPT> tag in the <HEAD> part of the HTML document:

```
`gs1:productName": "${productname$",  
"gs1:price": "${productprice$",  
`gs1:gtin13": "${gtin13$", ...
```

...which, in the rendered page becomes:

```
`gs1:productName": "Kettle",  
"gs1:price": "23.50",  
`gs1:gtin13": "5705548031312", ...
```

The resulting page sent to the customer contains the product information in both structured data and HTML.

### 3.3.2. Method 2: Embed GS1 SmartSearch and use a JavaScript routine to render empty HTML data

In this method, the web template is changed to include the placeholder variables only once – and that is within the structured data. In order for the customer to see the data, some JavaScript code needs to be added to the page template that will execute within the customer's web browser to write the additional HTML that will display that product information.

An important reason for choosing JavaScript Object Notation (JSON) for GS1 SmartSearch is that the web browser's most popular programming language, JavaScript, can read this data format easily.

However, if a customer has decided to disable JavaScript, or uses a web browser without JavaScript capability, no product information will be displayed using this method.

**Step 1: Use the placeholder variables to populate data into the structured data section of the web page.** Make sure that the structured data is between a `<SCRIPT>` and `</SCRIPT>` tag in the `<HEAD>` section of the web document. In this example, we have added two attributes to the opening `<SCRIPT>` tag which tells web browsers (and search engine 'spiders') that the data within the script tag is JSON data, with an name of "datablock1" which can be accessed by JavaScript code:

```
<script type="application/ld+json" id="datablock1">
```

**Step 2: Write the JavaScript code to access the structured data.** This code must be between separate `<SCRIPT>``</SCRIPT>` tags away from the structured data:

```
var data = document.getElementById("datablock1").innerHTML;
```

..or, if the web page has access to the popular *jQuery* library:

```
var data = $("#datablock1").html();
```

**Step 3: Convert the structured data into JavaScript's internal data format:**

```
var json = JSON.parse(data);
```

**Step 4: Now you can access any of the values in the data, like this:**

```
var productName = json['gs1:productName'];  
var currentPrice = parseFloat(json['gs1:price']);
```

..or loop through all the values like this example, where a for() loop is used to add rows to an existing <TABLE> with ID "productstable" in the HTML page:

```
<script>  
  for (var key in json)  
  {  
    if (json.hasOwnProperty(key))  
    {  
      propertyName = key.toString().replace('gs1:', '');  
      propertyValue = json[key].toString();  
      propertyValue = json[key].toString();  
      $('#productstable').append('<tr><td>' + propertyName + '</td>');  
      $('#productstable').append('<td>' + propertyValue + '</td></tr>');  
    }  
  }  
</script>  
<script src="jquery214.js"></script>  
<table id="productstable"></table>
```

Note the use of *jQuery* to simplify the code to access the table in JavaScript so that the code refers to the table as:

```
$('#productstable')
```

Most retailers are using *jQuery*, a fast, small, and feature-rich JavaScript library to simplify JavaScript programming.

### 3.3.3. Method 3: Import a GS1 SmartSearch document then render HTML using JavaScript

This method speaks to the strengths of GS1 SmartSearch, in that the product data source is under greater ownership and control of the GTIN licensee – the brand owner. Rather than the retailer having all the extensive data stored inside their ecommerce system, the web page is able to pull the GS1 SmartSearch data directly from the web servers of a brand owner, or from a product library / GDSN data source. The retailer then only has to store basic data about their product range such as GTIN13, and the data that represents their own unique offer for their consumers – such as price, loyalty points, offer information and their own descriptive 'voice' for the product.

Notwithstanding the high standards of availability and performance from these original data sources – especially GDSN services – there is a risk that customer's web browser cannot access the GS1 SmartSearch document as well as the retailer's web page, resulting in a mostly blank display! However, from a pilot perspective, seeing how well this third method works would be a useful exercise. One pragmatic approach might be that the retailer creates their own GS1 SmartSearch data server which stores the GS1 SmartSearch documents for their product range, with a name such as <http://id.retailername.com> - and to access any SmartSearch document the URL would be (for GTIN 5012345678901):

<http://id.retailername.com/5012345678901.gs1ss.js>



... which would return the structured data in the form of a text document.

### **Step 1: Build the set of GS1 SmartSearch product data files.**

You can use the toolset available at <https://www.gs1uk.org/web/webmarkuptool.html> to enter product information manually, or use a web template to convert many products into their structured data versions, as described in Method 1.

The result should be a set of text files, each containing the GS1 Smartsearch structured data for a single product. The file name should be <gtin13 value>.gs1ss.js. For example: [5012345678901.gs1ss.js](#)

### **Step 2: Save GS1 Smartsearch files to a public-accessible web location.**

For the purpose of piloting, a web server can be created to store these files. Usefully, since there is little server-side processing required of these files, a small and lower-powered web server will suffice. The server's work is simply to transfer the file to a customer's web browser. However the server will need to support *JSONP*, a variation of JSON where the server 'wraps' the contents of inside a JavaScript function. Building on the above example, the URL call would be:

<http://id.retailername.com/5012345678901.gs1ss.js?callback=gs1ss>

..and the data returned would be everything inside the file 5012345678901.gs1ss.js wrapped like this:

```
gs1ss(<contents of 5012345678901.gs1ss.js>);
```

It looks as if a function called `gs1ss()` has been created with the JSON structured data inside it. The name came from the `callback=` part of the URL.

If your brand doesn't have "JSONP callbacks" enabled on their servers, then contact the Digital team at GS1 UK team who can set up a JSONP forwarding server that will do the wrapping of the brand's document for you (for pilot use only!).

### Step 3: Use JavaScript to import the GS1 SmartSearch file into the web page for that product.

The method used by JavaScript to import the file is called *AJAX* (if the document is on the retailer's own web site) and *JSONP* if on a completely separate web site). The code, which uses the jQuery library, looks like this (assuming that the retailer's web page has already set a variable – in this example **gtin13value** -which describes the gtin13 of the product) if the GS1 SmartSearch documents are in the same domain as the retailer's e-commerce website:

```
$.ajax(  
{  
    url: 'http://id.retailer.com/' + gtin13value + '.gslss.js',  
    dataType: 'json',  
    async: false,  
    success: function(data)  
    {  
        window.ssJson = data;  
    }  
});  
json = JSON.parse(window.ssJson);
```

If importing the GS1 SmartSearch document from a completely different web domain (such as the brand's website) use this example:

```
$('#document').ready(function()  
{  
    var brandUrl = 'http://id.brand.com/' + gtin13value +  
    '.gslss.js&callback=gslss';  
    var gslss = function (data)  
    {  
        window.ssJson = data;  
    };  
    $.ajax({  
        url: brandUrl,  
        dataType: 'jsonp',  
        jsonpCallback: gslss,  
        jsonp: false,  
    });  
});
```

After this code is run, the data in the GS1 SmartSearch structured data can be accessed as demonstrated in Method 2; for example:

```
var productName = json['gs1:productName'];
```

## 4. Authoring Key Performance Indicators for a Pilot

### 4.1. KPI Authoring for a pilot – Important Background

Member companies who wish to take part in the pilot of GS1 SmartSearch are likely to consider key performance indicators and success factors when it comes to verifying that this new standard works for their consumers and helps their business.

The most likely indicator you will hear is that a member wishes to improve search ranking for their product pages on search engine services such as Google and Bing, a process known as Search Engine Optimisation (SEO).

When engaging with members, it is important **not** to say that GS1 SmartSearch will improve their search ranking. In our research, Google have told us that they are sensitive to third parties making statements about SEO techniques.

Fortunately, Google have produced an online guide called ‘Google Search Engine Optimization Guide’ (see Further Reading section), which summarises the steps web page designers should consider when making sure their pages are found in the search engine index. Web designers are likely to have read this document and followed its recommendations. If they have not done so, your first recommendation to ask them is to do so.

Adding GS1 SmartSearch structured data to a product details page is likely to cause some improvement in page ranking but, as Google warn in their SEO Guide:

*“Focusing too hard on specific tweaks to gain ranking in the organic results of search engines may not deliver the desired results. Search engine optimization is about putting your site’s best foot forward when it comes to visibility in search engines, but your ultimate consumers are your users, not search engines”.*

What we can tell our members is that Google are strong advocates of adding structured data, of which GS1 SmartSearch is an example, onto web pages. We know this because:

- Google and other search engine companies are founder members of *Schema.org* which has a data structure that can be used to tag important information on a web page to help search indexers understand that page better. GS1 SmartSearch extends the Schema.org definitions for products to allow the descriptions to be much more detailed and comprehensive.
- Google has started a Developers’ knowledgebase web site called “Promote Your Content with Structured Data Markup” (see *Further Reading*). On this site they state (emphasis mine):

*“‘Structured data markup’ is a standard way to annotate your content so machines can understand it. When your web pages include structured data markup, **Google (and other search engines) can use that data to index your content better**, present it more prominently in search results, and surface it in new experiences like voice answers, maps, and Google Now.”*

So we can say to members wishing to pilot GS1 Smartsearch that using this particular structured data format will allow search engines to **read product information data more easily** and **index this content better**. Google wishes to deliver relevant search results for user queries, so the better that it understands your page then the more likely it is to consider your page when presenting results to those queries.

With this background in mind, here are some recommended key performance indicators.

## 4.2. KPI 1: Relevant Search Improvements

- a. Agree on a search phrase with your member, which causes a particular product page on their site to appear in Google search results but ideally not on page 1. You should consider some special properties of the product to include in the search request. In this example it is the name of the product plus a couple of its properties: **Clair De Lune Changing Mat cotton candy blue**
- b. Ask the member to change their web template to include GS1 SmartSearch using method 1 described in this document (chosen as it is simple, easy and lowest risk).
- c. Every day for at least 28 days, test the search phrase and make a note of the page ranking position.
  - *Note:* We are not promising *general* page rank improvements; we are saying that pages with structured data are easier to index and thus are likely to appear higher on search results for *relevant* searches. This is why the KPI insists on adding extra search words to a basic product name. For general or vague product searches, e.g. **Changing Mat** there is no reason to believe that GS1 SmartSearch will improve page ranking, since these words are clearly visible on standard product pages. GS1 SmartSearch should make a difference for more focussed search requests because it makes available more of the product attributes in a machine-readable format.

## 4.3. KPI 2: Increased Sales Per Click-Through From Search Engine

- a. Ask the member for an average count, by day, of people landing on a particular product page (ideally one used in KPI 1) from Google. This should be available in web server logs and through web performance and analysis tools.
- b. Ask the member to calculate what percentage of those consumers landing on the page goes on to purchase the product (or, if the journey is not clear; add to basket, add to wish-list or other positive affirmation).
- c. The member should then apply the page template to provide the GS1 SmartSearch structured data.
- d. On a regular basis, ask the member to see if they achieve an improvement in sales as a percentage of consumers landing from Google.
  - *Note:* By improving page relevance for search engines, consumers arriving on a product page are more likely to find that the product is the one they are looking for, and thus proceed to purchase it. By keeping the KPI a sales increase as a *percentage* of total landings from Google, we rule out promising a member that they will have increased landings from Google. The member should also keep other factors as stable as possible, such as maintaining stock levels and not changing the price or adding / withdrawing an offer.

#### 4.4. KPI 3: Reduced Returns

- a. Ask the member to provide a percentage value for the return of a product as a percentage of sales of that product. Ideally they should be able to provide a breakdown of why a product was returned.
- b. The member should then apply the page template to provide the GS1 SmartSearch structured data.
- c. On a regular basis, ask the member to see if they have found that the returns are reduced as a percentage of sales.
  - *Note:* GS1 SmartSearch seeks to improve the quality of product data such that consumers trust what they are going to have delivered. So returns which have a reason 'Product not what I wanted' or even 'Not what I ordered' should be reduced. However faulty goods, damaged products and other return reasons are unlikely to be affected. This does depend on the level of comprehensiveness of the data used within the GS1 SmartSearch structured data.

#### 4.5. Notes on testing technologies

**Multi-variant (also known as A/B) Testing (MVT)** is a testing technique used to provide different pages to different consumers to see if certain key performance indicators can be improved. You can use MVT to switch the web template of a group of products to use the template containing GS1 SmartSearch. **However, once you have 'switched' a page, make it a permanent switch until the end of the pilot unless the retailer reports a problem.** MVT is the low-risk mechanism to switch pages in and out of GS1 SmartSearch mode; don't use MVT to change a particular product's page format frequently, nor use it to present different versions to different customer demographics, or by time of day, or any other reason. Search engines are slow to respond to changes of this nature and you may inadvertently report an improvement or deterioration of a KPI when the search engine is actually responding to an earlier version of the same page.

### 5. Further Reading:

- Google Search Engine Optimization Guide (PDF)  
<http://static.googleusercontent.com/media/www.google.co.uk/en/uk/webmasters/docs/search-engine-optimization-starter-guide.pdf>
- Google Developers – Promote Your Content With Structured Data Markup  
<https://developers.google.com/structured-data>
- GS1 SmartSearch Implementation Guideline  
<http://www.gs1.org/gs1-smartsearch/guideline/gtin-web-implementation-guideline>
- GS1 SmartSearch Vocabulary standard (for Grocery and Apparel)  
<http://www.gs1.org/gs1-smartsearch/latest>
- GS1 SmartSearch starter Kit  
<http://mozone.gs1.org/gs1-smartsearch>

Many thanks to Nick Lansley (GS1 UK) for writing this guide.

