

Schema Markup Checklist

A 7-point Checklist for Marketers to Audit their Markup



Introduction

The rise of semantic SEO and generative AI now requires brands to start leveraging Schema Markup to build a content knowledge graph that aids search engines in understanding and contextualizing web content.

In return, brands can prepare their content for AI search, enhance their visibility on the SERP, and drive more quality traffic to their site.

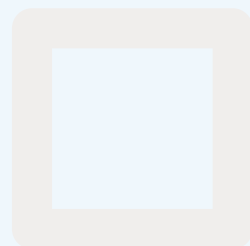
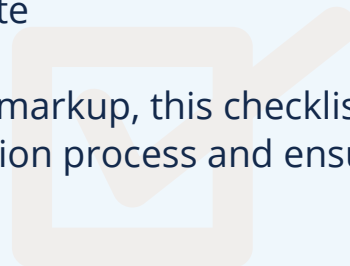
We created this Schema Markup Checklist to help marketers evaluate the current state of their markup and learn the best practices for implementing Schema Markup to build a content knowledge graph.

This checklist is primarily intended for marketers who:

- Want to implement Schema Markup not just for rich results but also for its semantic benefits
- Have a basic understanding of the Schema.org vocabulary
- Have existing markup on their site

That said, if your site does not have markup, this checklist is still a great resource to guide your implementation process and ensure your markup is off to a good start.

Happy Schema-ing!



Schema Markup Checklist

We recommend using this checklist to review the Schema Markup on each page of your site. You can dive deeper into each task for further guidance and best practices to improve your markup.



Item	Priority	Complexity
<input type="checkbox"/> Use the right Schema.org Type to categorize your page	High	Low
<input type="checkbox"/> Use relevant Schema.org properties that describe the content visible on your page	High	Low
<input type="checkbox"/> Ensure your markup is nested	Medium	Medium
<input type="checkbox"/> Include Uniform Resource Identifiers (URI) in your markup	High	Medium
<input type="checkbox"/> Use the appropriate properties to connect the entities across your site	High	High
<input type="checkbox"/> Link the entities on your site to other known entities on external authoritative knowledge bases	Medium	Medium
<input type="checkbox"/> Ensure your markup is aligned with the content on your page at all times	High	Medium



Use the right Schema.org Type to categorize your page

This task requires you to choose the most appropriate Schema.org Type for each page on your website.

Priority: High

Complexity: Low

In Schema.org, a Type is a category or class that can be used to categorize the entities described in your web page's content.

Each Schema.org Type also has a list of available properties to further describe the entity and its relationship to other entities defined on a site.

The primary purpose of implementing Schema Markup is to help search engines clearly understand the content on your page. Therefore, you should choose the most specific Schema.org type that best describes what your page is primarily about.

How to pick the most appropriate Schema.org Type

- 1 Identify the intent of your page**
E.g. Is it selling a product, showcasing the expertise of someone in your organization, or educating the public through a blog post?
- 2 Look at the [Schema.org type hierarchy](#) to see which type best aligns with the intent of your page**
E.g. If your page is selling a product, use [Product](#); if it is describing an employee, use [ProfilePage](#); if it is educating through blog content, use [BlogPosting](#).
- 3 Review the definition and the properties of your chosen Type on Schema.org to ensure it applies to your content**
E.g. [Article](#) has access to different properties than [WebPage](#). While both these types have properties like [author](#) and [datePublished](#), they also have unique properties for describing an entity of that type. Comparing a type's available properties to the content you want to mark up will support you in making the right decision.

Recommended Schema Type for Commonly Used Pages

Page Type	Schema Type
Homepage or About Page	https://schema.org/Organization
Contact Page	https://schema.org/ContactPage
Product Detail Page	https://schema.org/Product
Blog Post	https://schema.org/BlogPosting
Service Page	https://schema.org/Service
Event Overview Page	https://schema.org/Event
Category Page	https://schema.org/CollectionPage
Recipe Post	https://schema.org/Recipe
Physician Profile	https://schema.org/IndividualPhysician
Organization with a Physical Address or Store	https://schema.org/LocalBusiness

Use relevant Schema.org properties that describe the content visible on your page

This task involves reviewing the available properties for the chosen Schema.org Type and determining which properties you can use to describe the entity on your page.

Priority: High

Complexity: Low

In the Schema.org vocabulary, properties serve two functions.

- **They describe the attributes, qualities, or characteristics of a given entity.** For example, a product might have attributes like size and colour.
- **They express the relationships between entities.** Using properties to connect entities both within your content and with other authoritative knowledge bases, such as Wikidata, creates context. Properties do the heavy lifting when it comes to the semantic enrichment that machines need to support their understanding of your content.

Is your markup describing the entity on the page as thoroughly as possible?

There are a few common reasons why websites do not maximize the available properties:

- **The website uses a generic Schema Markup plugin that adds default markup to the page.** Website owners can't customize their markup and are subjected to the rudimentary markup added by the plugin.
- **Website owners are only adding Schema Markup to achieve rich results on Google.** As such, they only add the required properties to be eligible for the rich result and do not actually describe the content on their page using the properties available.

Good quality Schema Markup should include as many relevant properties for the chosen type as possible. This applies regardless of whether you're trying to achieve a rich result or develop a robust content knowledge graph.

If you want search engines and machines to truly understand the content and entities on your page, you need to maximize the relevant properties to clearly describe what your page is about.

That said, you should also avoid going overboard with the properties. Just because a property exists does not mean you need to use it. If you do not have the content for it or the property is not relevant to the entity you are describing, you should not use the property in your markup.

How to pick the most appropriate Schema.org properties

Answer these questions

- **Is there content on the web page that satisfies this property?**
- **Is this property providing useful information about the entity?**
- **Is this property required or recommended for a rich result?**

If you answered yes to any of these questions, you should include that property in your markup.

If a property provides useful information about the entity or is a required/recommended property for a rich result, but your web page does NOT have the content to satisfy the property, consider adding the relevant content to utilize the property. The Schema.org properties are a great way to inform your content on each page.

Ensure your markup is nested

This task requires you to review the structure of your markup and ensure your entities are organized in a meaningful hierarchy.

Priority: Medium **Complexity:** Medium

A common mistake SEOs make when implementing Schema Markup is creating separate markups when multiple entities are present on a page.

For example, a page could contain a product, a video of that product, and reviews and ratings of that product. All of this user-visible content should be marked up with structured data.

But if you specify each of these entities separately, you miss the opportunity to tell search engines what your page is mainly about and how the rest of these entities relate to the main entity. *Are the reviews and ratings about that product or are they random reviews and ratings for the organization that is selling the product?*

Therefore, we recommend nesting the entities on your page if they are relevant.

What is Nesting?

When a page has a main entity, nesting is the grouping of additional relevant entities on the page under the main entity in the markup.

Nesting helps:

- explain the hierarchy of your Schema Markup,
- establish relationships between different defined entities on that web page, and
- build your content knowledge graph.

When you structure your markup properly, search engines can understand the properties of your defined entities and how they relate to other entities.

For example, if you have reviews and ratings about your product on the page, you can nest them under the top-level Product type and use the properties `aggregateRating` and `review` to describe them.

<u>With Nesting</u>	<u>Without Nesting</u>
<pre><script type="application/ld+json"> [{ "@context": "https://schema.org/", "@type": "Product", "name": "White Tshirt", "description": "Soft, cotton white Ts." "aggregateRating": { "@type": "AggregateRating", "bestRating": "100", "ratingCount": "70", "ratingValue": "81", } }]</pre>	<pre><script type="application/ld+json"> [{ "@context": "https://schema.org/", "@type": "Product", "name": "White Tshirt", "description": "Soft, cotton white Ts." }, { "@context": "https://schema.org", "@type": "AggregateRating", "bestRating": "100", "ratingCount": "70", "ratingValue": "81", }]</pre>

When should you NOT nest your Schema Markup?

You should not nest the entities on a page if they are not related.

For example, [BreadcrumbList](#) markup tells search engines about the breadcrumb trail of the page. While [BreadcrumbList](#) markup can be used to achieve a rich result, its purpose is purely navigational and not semantic.

The other markup on your page will generally describe the content on your page to support [semantic SEO](#). Therefore, [BreadcrumbList](#) is usually not nested.

Include Uniform Resource Identifiers in your markup

This task involves including Uniform Resource Identifiers (URIs) in your markup to enable search engines to identify the unique entities on your site.

Priority: High

Complexity: Medium

A Uniform Resource Identifier is a string of characters that identifies a resource. It provides a consistent way to identify resources across different systems and protocols. URIs are important if you want to utilize your Schema Markup to build your content knowledge graph.

For your entity to be identifiable and retrievable, it must have a distinct URI so that machines and search engines can:

- Identify the unique entities on your site and
- Know which entities you are referring to when you connect your entities in your knowledge graph.

In JSON-LD, URIs are expressed with the '@id' attribute. By adding the '@id' attribute to the entities in your Schema Markup, you can easily connect and refer back to other entities on your site so that search engines can clearly understand their relationship.

Google uses @ids for some rich results like [Product Variant](#), [Book](#), and [Profile Page](#). In Google's documentation, they call this an identifier.

How to include URIs in your markup

You can do this manually by following these steps.

1 Enter the URL of the page that represents the entity

E.g. www.schemaapp.com

2 Add # followed by the Schema.org Type (URL + #Type)

E.g. www.schemaapp.com/#Organization/

Example of URIs in JSON-LD markup

Organization	
ID:	https://www.schemaapp.com/#Organization
@type	Organization
@id	https://www.schemaapp.com/#Organization
description	Schema App is an end-to-end Schema Markup solution that helps enterprise SEO teams develop a knowledge graph and drive search performance.
name	Schema App
url	https://www.schemaapp.com/
address	
@type	PostalAddress
@id	https://www.schemaapp.com/#PostalAddress
streetAddress	201 - 412 Laird Road
postalCode	N1G 3X7
addressRegion	Ontario
addressLocality	Guelph
name	Schema App Address
founder	
@type	Person
@id	https://www.schemaapp.com/author/vberkel/#Person
sameAs	https://www.linkedin.com/in/markvanberkel/

The @id for the entity, Schema App, which is the organization described on www.schemaapp.com.

The @id for the entity, Schema App Address, which is the postal address of Schema App as described on www.schemaapp.com.

The @id for the entity, Mark van Berkel, who is the co-founder of Schema App. His entity has been defined on his author page.

Every entity in your markup should have a unique @id. [Google recommends](#) using only hashtags when referring to entities that exist on the same page.

However, this can be confusing if you are connecting entities across multiple web pages. Therefore, we recommend using the URL+#Type URI structure. Both options are fine. However, using just the # method will prevent you from connecting entities across your site.

How we create URIs at Schema App

When you publish your Schema Markup using the [Schema App Highlighter](#) or [Editor](#), our tool automatically generates HTTPS URIs for the entities defined in it.

That said, not every schema authoring tool will automatically generate URIs for your entities. This is because they're usually only generating Schema Markup for rich results rather than building a reusable knowledge graph.

Use the appropriate properties to connect the entities across your site

This task involves reviewing the entities across your site and connecting them through the properties that best describe their relationship.

Priority: High

Complexity: High

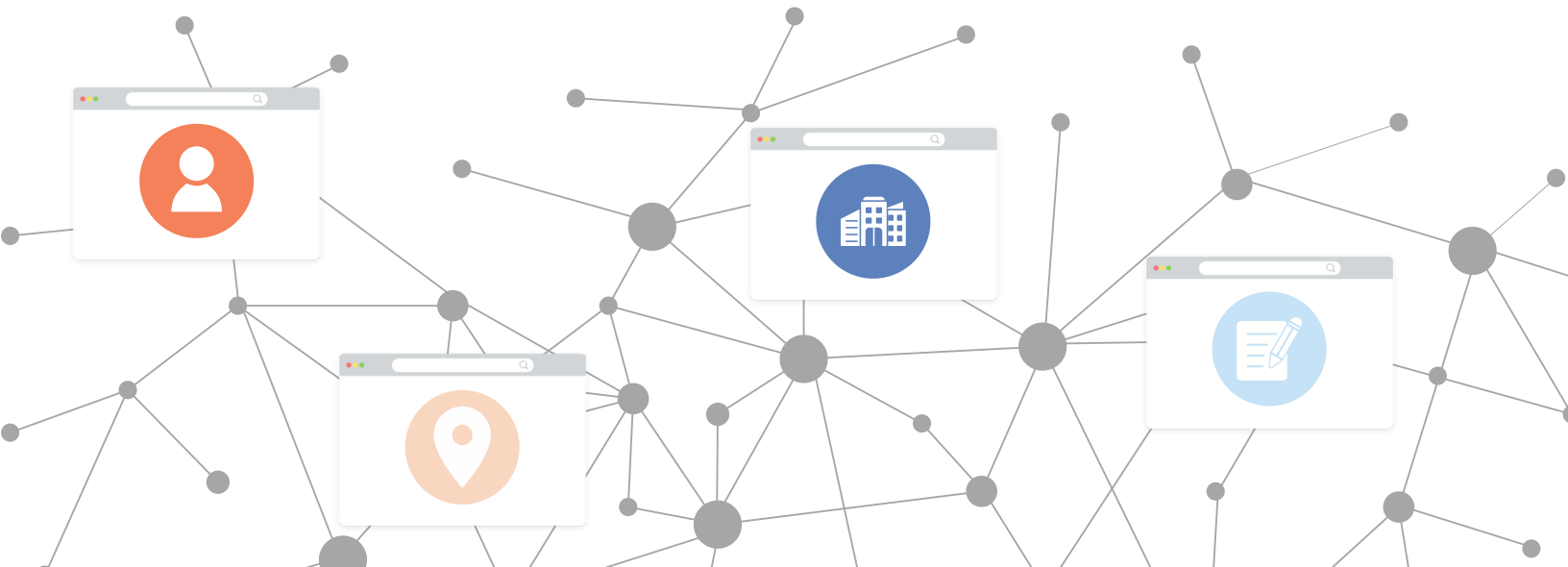
In previous sections of this checklist, we've discussed nesting entities on a single page to explain their relationships. However, it is equally important to consider the connections between the entities across your entire website.

When marking up the entities on a page, ask yourself if any of these entities have been previously defined on other pages of your site.

If they have been defined on other pages of your site, you should be referencing that entity's identifier instead of creating a new duplicate entity.

This task is more complicated because it requires you to:

- Create @ids for the entities on your site
- Customize your markup to leverage the appropriate properties to connect these entities
- Know all the entities on your site to recognize when and how to connect them.



Here's an example

You have a blog post written by your CTO, Mark van Berkel, and you want to markup that blog post and utilize the author property under the BlogPosting type.

If you have an author page on your site for Mark van Berkel and you've defined him as an entity on that page with a URI, the blog post should use the author property to reference the URI created for him on his author page.

If you do not do this and just create another URI for Mark based on the URL of the blog post, you are creating a duplicate entity which inhibits the creation of a well-connected knowledge graph.

Well Connected Entities

Since Mark has already been defined on his author page, we can connect his entity to this article.

```
<script type="application/ld+json">
[ {
  "@context": "https://schema.org/",
  "@type": "BlogPosting",
  "@id":
  "https://www.schemaapp.com/blog/abc-by-
mark/#BlogPosting"
  "headline": "ABC by Mark",
  "author": {
    "@type": "Person",
    "@id":
    "https://www.schemaapp.com/author/mark-van-
berkel",
    "name": "Mark van Berkel",
  }
}
]
```

Duplicate Entities

If we created a new entity for Mark on this page despite his entity already being created on his author page, we are creating a duplicate entity for him.

```
<script type="application/ld+json">
[ {
  "@context": "https://schema.org/",
  "@type": "BlogPosting",
  "@id":
  "https://www.schemaapp.com/blog/abc-by-
mark/#BlogPosting"
  "headline": "ABC by Mark",
  "author": {
    "@type": "Person",
    "@id":
    "https://www.schemaapp.com/blog/abc-by-
mark/#Person",
    "name": "Mark van Berkel",
  }
}
]
```

When you connect the entities across your site using appropriate properties, this helps you develop an accurate content knowledge graph and provides search engines with contextual information about the entities on your site.

How to connect the entities on your site

1 Pick the property that best describes the relationship between both entities

Most types in the Schema.org vocabulary have forty or more properties available, giving you many options to define the relationship. More descriptive properties mean a more informative knowledge graph, which is the point of [semantic SEO](#)!

For example, an Article might [mention](#) an Organization, but it may be more accurate to say the Article is [published by](#) that Organization.



Pro Tip: Use the free [Schema Paths tool](#) to see which [Schema.org](#) properties can be leveraged to connect your entities.

2 Include the @id of the entity you want to connect to using the chosen property

Remember, the same entity probably exists on multiple pages of your website, but that entity is represented by a single @id.

If you want to connect two entities, make sure you reference the entity by using the right @id to avoid creating duplicate entities.

Less Descriptive:
The Article mentions Organization
with a link to Organization's @id

```
<script type="application/ld+json">
[
  {
    "@context": "https://schema.org/",
    "@type": "Article",
    "headline": "Title of the article",
    "mentions": {
      "@type": "Organization",
      "@id": "https://schemaapp.com/#Organization",
      "name": "Schema App",
    }
  }
]
```

More Descriptive:
The Article's publisher is Organization
with a link to Organization's @id

```
<script type="application/ld+json">
[
  {
    "@context": "https://schema.org/",
    "@type": "Article",
    "headline": "Title of the article",
    "publisher": {
      "@type": "Organization",
      "@id": "https://schemaapp.com/#Organization",
      "name": "Schema App",
    }
  }
]
```



Link the entities on your site to other known entities on external authoritative knowledge bases

This task involves identifying the entities mentioned in the text on your website and linking them to the corresponding known entities on external authoritative knowledge bases to disambiguate your entities.

Priority: Medium **Complexity:** Medium

When you think about the word 'Apple,' what comes to mind? The fruit? The company? Your childhood friend with that first name?

This example is a simple illustration of how words can be ambiguous because words can have multiple meanings. As search becomes more semantic, you can help search engines disambiguate the entities on your site by doing entity linking.

What is Entity Linking?

In the context of Schema Markup, entity linking is the act of identifying entities mentioned in the text on your website and linking them to the corresponding known entities on external authoritative knowledge bases such as Wikipedia, Wikidata, and Google's Knowledge Graph.

Entity linking can help websites:

- Define the terms and entities mentioned in their content more explicitly
- Build a more descriptive content knowledge graph

This enables search engines to disambiguate the entities identified on your site with greater confidence and provide users with more accurate and relevant search results, which drives more quality traffic to your site.

How to do Entity Linking

1 Identify an entity mentioned in your text

Example: Your page mentions the entity Paris because your organization is located in Paris.

2 Find the URI of the corresponding known entity on Google's Knowledge Graph, Wikipedia, or Wikidata

Example: The 'Paris' you are referring to is the capital of France, and these are the URIs for the entity Paris on...

- **Google's KG:** kg:/m/05qtj
- **Wikipedia:** <https://en.wikipedia.org/wiki/Paris>
- **Wikidata:** <https://www.wikidata.org/entity/Q90>

3 Use the appropriate Schema.org property to link your entity to these linked entities

Example: Paris is your organization's postal address, so you can use the sameAs property to disambiguate the entity.

```
"address":
  "@type": "PostalAddress",
  "@id": "https://www.schemaapp.com/#PostalAddress ",
  "name": "Schema App Address",
  "addressCountry":
    "@type": "Country",
    "name": "Paris",
    "sameAs": "kg:/m/05qtj",
    "sameAs": "https://en.wikipedia.org/wiki/Paris",
    "sameAs": "https://www.wikidata.org/entity/Q90",
```



If you are doing entity linking manually, keeping the data updated and accurate can be resource-intensive and time-consuming, especially if you have a huge site.

Pro Tip: If you're looking for a scalable way to do entity linking, Schema App's Omni Linked Entity Recognition (LER) feature can help. Learn more about Omni LER [here](#).



Ensure your markup is aligned with the content on your page at all times

This task involves reviewing your markup to ensure it matches the visible content on your web page.

Priority: High

Complexity: Medium

Google requires websites to mark up content that is visible to users, which might sound easy. But Schema Drift is a common issue that plagues sites that have markup hardcoded into their HTML.

What is Schema Drift?

Schema Drift is the divergence between the markup and the content on your page. It can happen when Google updates their structured data requirements, Schema.org removes certain properties, content changes are made, websites migrate to a new CMS, and more.

Schema drift is ultimately a data quality issue that can affect your rich result eligibility and the search engine's ability to understand your content.

How to identify Schema Drift

- 1 Enter a page on your site in the [Schema.org validator](#)
- 2 Review the markup in the Schema.org validator against the visible content on your page

This can be laborious if your site has many pages. When you choose the manual method of adding Schema Markup to your site, you have to review and update your markup whenever content changes are made to prevent schema drift.



Pro Tip: Use programmatic solutions like the [Schema App Highlighter](#) to generate your Schema Markup dynamically based on your page's content.

Good quality Schema Markup starts here.

Like everything else in SEO, Schema Markup is a tactic that requires ongoing maintenance and improvements for long-term success.

This checklist should be repeated across all the key pages on your site to help you understand the true state of your website's Schema Markup and inch you closer to building a robust content knowledge graph that supports your SEO and AI efforts.

Recommended Resources

- **Schema Markup 101** Online Course
- **Content Knowledge Graph Fundamentals** Online Course
- **Guide to Entities & Knowledge Graphs for SEO** eBook
- **How to Manage Your Schema Markup**
- **Knowledge Graph: Value of Schema Markup Beyond Rich Results**



Schema App is an end-to-end Schema Markup solution for enterprise SEO teams. From developing a tailored Schema Markup strategy to implementing and managing it at scale, we help teams leverage the power of Schema Markup to stand out in search and drive AI performance.

[Book a Demo](#)



www.SchemaApp.com