Engineer

Structured data

facts about a page, written for machines

What it is

A company's contact page says in a sentence where the office is; structured data says it again as a `PostalAddress` with a street, a postcode and a country, so that a program reading the page has no need to guess which word is the town. It is the practice of repeating what a page states in prose as typed statements in a vocabulary that publishers and consumers agree on, so that search engines and other software can read the facts without interpreting the layout.

The vocabulary is usually schema.org, which Google's documentation names as the one most structured data for its search uses. Google, Microsoft, Yahoo and Yandex founded it, and since April 2015 a W3C community group has been its main forum. It defines types, such as `Organization`, `Person`, `Service`, `Review` or `DefinedTerm`, and the properties each one can carry, such as `name`, `address`, `author` or `price`.

The statements can be written in three syntaxes. Microdata and RDFa add attributes to the visible HTML elements themselves. JSON-LD keeps the statements apart, as a block of JSON in a `script` element of type `application/ld+json`, and is a W3C Recommendation in its 1.1 version, published on 16 July 2020. Google accepts all three equally and recommends JSON-LD as the easiest to maintain at scale.

```
<script type="application/ld+json">
    {
        "@context": "https://schema.org",
        "@type": "Organization",
        "@id": "https://example.com/#organization",
        "name": "Example Ltd",
```

```
        "url": "https://example.com/"
    }
</script>
```

The line that matters most in that block is `@id`. JSON-LD uses it to identify a node, and a node without one is a blank node, with no document-independent identifier. Blank-node identifiers can be shared by blocks within one document. With a globally usable identifier, any other block on any other page can write `"publisher": {"@id": "https://example.com/#organization"}`, which the specification calls a node reference: a pointer to a thing described elsewhere, instead of a fresh description of it. Identifiers are what turn a scatter of blocks into one graph.

The neighbouring term is metadata. The title, the description and the Open Graph tags in a page's head describe the page as a document, while structured data describes the things the page is about: the company behind it, the person who wrote it, the service it offers or the word it defines.

Use the concept in a decision

Put "What is structured data?" to work in a real decision. We can help relate the concept to the system you are building or buying. A site or an app that the AI assistants your customers use can find, read and act on, checked on every release rather than promised once.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Built a multilingual Hugo site in 3 languages — 108 template files serving three languages and four formats

Built a multilingual Hugo site in 3 languages Get in touch →

Why it matters

The decision is how much to say, and the honest answer is usually less than the vocabulary allows, but complete. Nothing in the syntax stops a publisher from filling in every property schema.org offers. Google's own guidance points the other way: supply fewer recommended properties, complete and accurate, rather than every possible one with partial or wrong data, and mark up nothing a reader cannot see on the page, even when it is true.

The reason is that every marked-up fact is a second copy of something the page already says, and a second copy can drift. A price edited in the visible table and forgotten in the JSON, a rating that appears only to machines, an address typed slightly differently on each page: each makes the page say one thing to people

and another to programs. The cost of structured data lies less in writing it than in keeping it true, so the useful questions are which statements the page can stand behind and whether they come from the same source as the visible text.

What it buys is legibility. An engine or an agent that can join a hundred mentions into one company knows more than one reading a hundred unrelated names. Pages that follow the guidelines may become eligible for rich results, listings with extra detail, though eligibility guarantees none. What changes is how well a machine understands what the page is about.

What it looks like in practice

The site's credits commit it to marking only what is true and visible on the page, and the history of its structured data shows why that sentence needed writing down.

Before the July 2026 rework, 963 blocks restated the company 1,671 times, with no stable identifier anywhere, so no consumer had grounds to treat the organisation on one page as the organisation on another. The graph now rests on three identifiers minted from the site's origin, for the organisation, the website and the founder, and every other block refers to them by `@id` instead of describing them again. The identifiers are the same in every language: a comment in the template explains that a separate Danish identifier would declare a second organisation with the same company registration number. Sixteen types were recorded at that stage, and lists named their items by URL alone, because writing out the 107 entries then on the portfolio index would have made the compressed page 52 per cent heavier.

Several facts are read from the same source as the visible page. Service prices come from the same data file as the prices on screen, so a machine and a reader are quoted the same rate. The period a piece of work covers comes from the partial that writes the sentence about it. Breadcrumbs follow the visible trail, and list positions follow the order on screen, never a ranking published only to machines. The page's own node uses the same address as its canonical link, so the markup and the head make one claim.

The refusals are as deliberate as the statements. The company's own write-ups of its work are not marked up as reviews, because that would invent an author for its own words. The testimonial template uses Review nodes only for entries

published with granted consent, without adding a rating. Each glossary page, this one included, carries a `DefinedTerm` whose description is the one-line gloss shown in the list, so the definition a machine reads is the one a reader sees.

A check over the built site asserts that every block parses, that every identifier resolves, and that every URL the graph names was built, so a graph pointing at a page that does not exist fails the gate. The same credits page lists the Schema Markup Validator for reading the data and Google's Rich Results Test for seeing what one engine does with it.

Where it goes wrong

Rich results stay out of reach although every validator passes. The markup is valid JSON-LD and uses real schema.org types, yet the page keeps appearing as a plain result. The data describes things a reader cannot see, such as a rating nobody gave, questions hidden from the page or a price that differs from the one on screen, and Google warns that markup breaking its guidelines may leave a page ineligible, however clean the syntax. Mark up only what the page shows, and generate both from one source so they cannot part.

Hundreds of blocks describe one company, and nothing joins them. Every page repeats the organisation's name, logo and address, sometimes with small differences, and correcting one detail means finding all of them. The nodes are anonymous, or each page and each language mints its own identifier for the same entity. Give every real thing one `@id` built from a stable URL with a fragment, describe it once, and refer to it everywhere else.

One template change breaks the markup on every page at once. A stray quote, a trailing comma or an empty value turns every block invalid, and consumers can use none of it. Google's guide warns that templating can break markup after deployment. The JSON was assembled as text, with values pasted between braces, so a missing field leaves an empty string or a broken structure. Build the statement as a data structure and serialise it, as this site's templates do with Hugo's `jsonify`, omit properties that have no value instead of emitting them empty, and parse every block in the built output.

The graph names pages that are missing. URLs in the markup lead to a 404, a staging host or a mirrored copy. The addresses were typed by hand or built by a different rule from the links and the canonical. Derive every URL in the markup

from the same function that produces the page's canonical address, and check that each one was built.

Further reading

- schema.org — the shared vocabulary of types and properties, with a page for every type.
- JSON-LD 1.1 — the W3C Recommendation for the syntax, including node identifiers and node references.
- schema.org: DefinedTerm — the type each glossary page uses to describe the word it explains.
- Google Search Central: introduction to structured data — the three syntaxes, the visibility rule and how rich results are tested.
- Schema Markup Validator — reads any page's structured data against the schema.org vocabulary.

Glossary

Choose for your actual constraints

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Make the terminology practical

Ask how "What is structured data?" applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/structured-data/