Engineer

Données structurées

les faits d'une page, écrits pour les machines

Ce que c’est

La page de contact d’une entreprise indique dans une phrase où se trouve son bureau ; les données structurées le répètent sous la forme d’une `PostalAddress` avec une rue, un code postal et un pays, pour qu’un programme lisant la page n’ait pas à deviner quel mot désigne la ville. Cette pratique consiste à reprendre ce qu’une page énonce en prose sous forme d’énoncés typés, dans un vocabulaire convenu entre les éditeurs et ceux qui exploitent ces données, afin que les moteurs de recherche et les autres logiciels puissent lire les faits sans interpréter la mise en page.

Ce vocabulaire est généralement schema.org, que la documentation de Google désigne comme celui qu’utilisent la plupart des données structurées destinées à son moteur de recherche. Google, Microsoft, Yahoo et Yandex l’ont fondé, et un groupe communautaire du W3C en est le principal lieu de discussion depuis avril 2015. Il définit des types, comme `Organization`, `Person`, `Service`, `Review` ou `DefinedTerm`, ainsi que les propriétés que chacun peut porter, comme `name`, `address`, `author` ou `price`.

Les énoncés peuvent s’écrire dans trois syntaxes. Microdata et RDFa ajoutent des attributs directement aux éléments HTML visibles. JSON-LD garde les énoncés à part, dans un bloc JSON à l’intérieur d’un élément `script` de type `application/ld+json` ; sa version 1.1 est une recommandation du W3C publiée le 16 juillet 2020. Google accepte les trois à égalité et recommande JSON-LD comme le plus facile à maintenir à grande échelle.

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

```
        "name": "Example Ltd",
        "url": "https://example.com/"
    }
</script>
```

La ligne la plus importante de ce bloc est `@id`. JSON-LD l'utilise pour identifier un nœud ; un nœud qui n'en possède pas est un nœud anonyme, sans identifiant indépendant du document. Les identifiants de nœuds anonymes peuvent être partagés entre des blocs d'un même document. Avec un identifiant utilisable globalement, n'importe quel autre bloc de n'importe quelle autre page peut écrire `"publisher": {"@id": "https://example.com/#organization"}`, ce que la spécification appelle une référence de nœud : un renvoi vers une chose décrite ailleurs, plutôt qu'une nouvelle description. Ce sont les identifiants qui transforment des blocs épars en un seul graphe.

Le terme voisin est « métadonnées ». Le titre, la description et les balises Open Graph dans l'en-tête d'une page décrivent la page en tant que document, tandis que les données structurées décrivent les choses dont elle parle : l'entreprise qui la publie, la personne qui l'a écrite, la prestation qu'elle propose ou le mot qu'elle définit.

Utilisez le concept pour prendre une décision

Utilisez « Que sont les données structurées ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Un site ou une application que les assistants d'IA de vos clients peuvent trouver, lire et utiliser : une capacité vérifiée à chaque livraison, plutôt qu'une promesse ponctuelle.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Site Hugo multilingue en 3 langues — 108 fichiers de modèles pour trois langues et quatre formats

Site Hugo multilingue en 3 langues Contactez-nous →

Pourquoi c'est important

La décision porte sur la quantité d'informations à fournir ; la réponse honnête est généralement d'en fournir moins que le vocabulaire ne le permet, mais de façon complète. Rien dans la syntaxe n'empêche un éditeur de remplir toutes les propriétés proposées par schema.org. Les consignes de Google vont dans l'autre sens : fournir moins de propriétés recommandées, mais complètes et exactes, plutôt que toutes les propriétés possibles avec des données partielles ou fausses, et ne rien baliser qu'un lecteur ne puisse voir sur la page, même si c'est vrai.

La raison est que chaque fait balisé est une seconde copie de ce que la page dit déjà, et qu'une seconde copie peut diverger. Un prix corrigé dans le tableau visible mais oublié dans le JSON, une note qui n'apparaît que pour les machines, une adresse saisie un peu différemment sur chaque page : chacun de ces cas fait dire à la page une chose aux personnes et une autre aux programmes. Le coût des données structurées tient moins à leur rédaction qu'au maintien de leur exactitude. Les questions utiles sont donc de savoir quels énoncés la page peut assumer et s'ils viennent de la même source que le texte visible.

Le bénéfice est la lisibilité. Un moteur ou un agent capable de réunir cent mentions en une seule entreprise en sait davantage qu'un autre qui lit cent noms sans lien. Les pages qui suivent les consignes peuvent devenir admissibles aux résultats enrichis, des résultats de recherche comportant des détails supplémentaires, sans que cette admissibilité garantisse leur apparition. Ce qui change, c'est la compréhension par la machine du sujet de la page.

À quoi cela ressemble en pratique

Les crédits du site l'engagent à ne baliser que ce qui est vrai et visible sur la page, et l'histoire de ses données structurées explique pourquoi il fallait écrire cette phrase.

Avant la refonte de juillet 2026, 963 blocs redécrivaient l'entreprise 1 671 fois, sans aucun identifiant stable : aucun consommateur de ces données n'avait donc de raison de considérer l'organisation d'une page comme celle d'une autre. Le graphe repose désormais sur trois identifiants créés à partir de l'origine du site, pour l'organisation, le site web et la fondatrice ; tous les autres blocs y font référence par `@id` au lieu de les décrire à nouveau. Les identifiants sont les mêmes dans toutes les langues : un commentaire du modèle explique qu'un identifiant danois distinct déclarerait une seconde organisation portant le même numéro d'immatriculation. Seize types étaient consignés à cette étape, et les listes ne nommaient leurs éléments que par leur URL, car détailler les 107 entrées alors présentes sur l'index du portfolio aurait alourdi la page compressée de 52 pour cent.

Plusieurs faits sont lus dans la même source que la page visible. Les prix des prestations viennent du même fichier de données que les prix à l'écran : la machine et le lecteur se voient ainsi annoncer le même tarif. La période couverte par un travail vient du modèle partiel qui écrit la phrase à son sujet. Les fils d'Ariane suivent le parcours visible, et les positions dans les listes suivent l'ordre à

l'écran, jamais un classement publié uniquement pour les machines. Le nœud de la page elle-même utilise la même adresse que son lien canonique : le balisage et l'en-tête expriment ainsi la même chose.

Les refus sont aussi délibérés que les énoncés. Les récits que l'entreprise fait de son propre travail ne sont pas balisés comme des avis, car cela reviendrait à inventer un auteur pour ses propres mots. Le modèle de témoignages n'utilise des nœuds Review que pour les entrées publiées avec un consentement accordé, sans ajouter de note. Chaque page du glossaire, celle-ci comprise, porte un `DefinedTerm` dont la description est la définition en une ligne affichée dans la liste : la définition lue par la machine est donc celle que voit le lecteur.

Un contrôle du site généré vérifie que chaque bloc peut être analysé, que chaque identifiant se résout et que chaque URL nommée par le graphe a été générée. Un graphe renvoyant vers une page inexistante fait ainsi échouer le contrôle qualité obligatoire. La même page de crédits cite le Schema Markup Validator pour lire les données et le Rich Results Test de Google pour voir ce qu'un moteur en fait.

Où les problèmes surviennent

Les résultats enrichis restent hors de portée alors que tous les contrôles de validation réussissent. Le balisage est du JSON-LD valide et utilise de vrais types schema.org, mais la page continue d'apparaître comme un résultat ordinaire. Les données décrivent des choses que le lecteur ne peut pas voir, comme une note que personne n'a donnée, des questions absentes de la page ou un prix différent de celui affiché. Google prévient qu'un balisage contraire à ses consignes peut rendre une page inadmissible, quelle que soit la qualité de la syntaxe. Ne baliser que ce que la page montre et générer les deux à partir d'une seule source pour qu'ils ne puissent pas diverger.

Des centaines de blocs décrivent une entreprise, sans rien qui les relie. Chaque page répète le nom, le logo et l'adresse de l'organisation, parfois avec de petites différences, et corriger un détail impose de les retrouver tous. Les nœuds sont anonymes, ou chaque page et chaque langue créent leur propre identifiant pour la même entité. Donner à chaque chose réelle un seul `@id` construit à partir d'une URL stable avec un fragment, la décrire une fois et y faire référence partout ailleurs.

Une modification de modèle casse le balisage de toutes les pages à la fois. Un guillemet égaré, une virgule finale ou une valeur vide invalide tous les blocs, que les consommateurs ne peuvent plus exploiter. Le guide de Google avertit que les modèles peuvent casser le balisage après le déploiement. Le JSON a été assemblé comme du texte, avec des valeurs collées entre des accolades : un champ manquant laisse donc une chaîne vide ou une structure cassée. Construire l'énoncé comme une structure de données puis le sérialiser, comme le font les modèles de ce site avec `jsonify` de Hugo ; omettre les propriétés sans valeur plutôt que de les émettre vides, et analyser chaque bloc de la sortie générée.

Le graphe nomme des pages absentes. Les URL du balisage aboutissent à une erreur 404, à un hôte de préproduction ou à une copie miroir. Les adresses ont été saisies à la main ou construites selon une règle différente de celle des liens et de l'adresse canonique. Déduire chaque URL du balisage de la même fonction que celle qui produit l'adresse canonique de la page, et vérifier que chacune a été générée.

Pour aller plus loin

- schema.org — le vocabulaire partagé de types et de propriétés, avec une page pour chaque type.
- JSON-LD 1.1 — la recommandation du W3C pour la syntaxe, notamment les identifiants et les références de nœuds.
- schema.org : DefinedTerm — le type utilisé par chaque page du glossaire pour décrire le mot qu'elle explique.
- Google Search Central : introduction aux données structurées — les trois syntaxes, la règle de visibilité et la manière de tester les résultats enrichis.
- Schema Markup Validator — lit les données structurées de toute page à l'aide du vocabulaire schema.org.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Que sont les données structurées ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/structured-data/