Engineer

Métadonnées

Données décrivant d’autres données pour les retrouver

Ce que c’est

Les métadonnées sont des données portant sur d’autres données : titre, date, langue, auteur, format et taille accompagnant un document pour que personnes et programmes puissent le classer, lui faire confiance et agir avant de le lire. Les normes anciennes écrivent meta-data avec un trait d’union ; au quotidien, elles portent des noms plus précis : champs d’en-tête HTTP, propriétés d’un fichier bureautique, étiquettes d’une photo, en-tête de métadonnées au début d’une page Markdown.

Une répartition courante distingue trois types. Les métadonnées descriptives disent de quoi il s’agit : titre, créateur, sujet, résumé. Les métadonnées structurelles disent comment les parties s’assemblent : chapitres, pages, ordre de lecture. Les métadonnées administratives disent l’origine et les usages autorisés : dates de création et de modification, personnes responsables de ces opérations, licence et format. Le Dublin Core, quinze éléments devenus la norme ISO 15836 en 2003, est la liste partagée de champs descriptifs la plus connue ; titre, créateur, date, langue et droits réapparaissent sous d’autres noms dans presque tous les formats apparus depuis.

Une page web porte plusieurs couches de métadonnées. La réponse HTTP vient d’abord. La RFC 9110 dit qu’un champ d’en-tête « peut modifier ou étendre la sémantique du message, décrire l’expéditeur, définir le contenu ou fournir un contexte supplémentaire » ; `Content-Type`, `Content-Language` et `Last-Modified` décrivent la page avant l’arrivée du premier octet. Dans le HTML se trouvent `<title>`, l’attribut `lang` et les éléments `<meta>` dont la norme définit les noms, comme `description`, `author` et `generator`. Au-dessus viennent les vocabulaires lus par d’autres acteurs : propriétés Open Graph des aperçus de liens et données structurées pour moteurs de recherche et assistants.

Qu’est-ce qui sépare donc métadonnées et données ? Le lecteur, plus que les octets. Une notice de catalogue est une métadonnée sur un livre, mais, pour le catalogue, elle constitue les données elles-mêmes ; l’heure de prise de vue d’une photo est un détail pour qui la regarde et une matière première pour qui cartographie les lieux de prise de vue. La distinction avec les données structurées est plus étroite : celles-ci sont des métadonnées écrites dans un vocabulaire partagé, pour qu’un programme lise un champ sans deviner son sens.

Utilisez le concept pour prendre une décision

Utilisez « Que sont les métadonnées ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Des règles et des contrôles qui détectent les données incorrectes dès leur arrivée, avec une réponse claire sur le responsable et l'origine de chaque chiffre.

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. Un plan d'infrastructure pour 14 départements — 14 départements de recherche sur un même plan d'infrastructure

Un plan d'infrastructure pour 14 départements Contactez-nous →

Pourquoi c’est important

Chaque champ est une petite promesse ; la décision importante est l’origine de sa valeur : saisie humaine, historique des versions, colonne d’une base ou simple information du système de fichiers. Les programmes agissent sur ces promesses sans les vérifier. C’est le but des métadonnées et leur risque.

Caches et robots répartissent leurs efforts à partir d’elles. Un navigateur possédant une copie renvoie la date `Last-Modified` reçue ; si rien n’a changé, il obtient une courte réponse `304 Not Modified` plutôt que la page complète. La RFC 9110 demande au serveur cette date pour toute représentation dont la dernière modification « peut être déterminée raisonnablement et de manière cohérente ». La condition joue dans les deux sens : sans date cohérente, la norme n’en demande pas. La date de modification d’un sitemap remplit le même rôle pour un robot décidant quoi récupérer à nouveau.

Les personnes et technologies d’assistance en dépendent aussi. L’attribut `lang` indique au lecteur d’écran les règles de prononciation, et le titre apparaît dans l’onglet, le favori et le résultat de recherche. Les droits circulent de même : un gestionnaire de paquets connaît la licence d’une dépendance uniquement parce qu’elle l’a déclarée. Un examen des licences fondé sur cette liste vaut donc exactement ce que valent les déclarations.

Chaque champ doit rester vrai : une page qui comporte quarante champs offre quarante occasions d'erreur. Certains ne sont lus par personne : la norme HTML note que de nombreux moteurs de recherche ignorent le nom `keywords`, utilisé pour polluer les résultats. Les métadonnées divulguent aussi des informations : la position d'une photo ou l'historique d'un document peut en dire davantage que le contenu. Moins de champs, chacun alimenté par une source qui ne peut dériver, valent généralement mieux qu'une multitude.

À quoi cela ressemble en pratique

Les métadonnées apparaissent dans trois projets de ce site, chaque fois dans un autre type de fichier : sitemap, ensemble de PDF et graphe de dépendances.

Le sitemap d'un site tirait ses dates de dernière modification des mauvaises métadonnées : 172 de ses 176 pages partageaient le même instant. Ces dates étaient les heures de modification des fichiers, décrivant les octets plutôt que le contenu ; chaque export réécrivait tout. La correction lit une chaîne de repli : champ explicite, historique des commits, puis fichier. Elle consigne aussi le piège : le nom littéral du champ doit figurer dans cette liste, sinon le générateur ne le lit jamais. Les enregistrements des pages distinguent désormais les jours de rédaction et de révision des années du travail décrit ; une date inconnue reste vide, selon la règle « une valeur précise mais fausse est pire qu'une valeur absente ». Au stade d'implémentation documenté, une resynchronisation après un mois de modifications touchait 8 fichiers sur 186.

Au stade de validation consigné, neuf documents PDF générés évalués selon la norme d'accessibilité ont réussi les 106 règles configurées du profil d'accessibilité. Le travail comprenait une langue déclarée et un vrai titre dans les métadonnées, ainsi qu'une structure balisée, des textes alternatifs et un ordre de lecture explicite. Lors de cette exécution consignée, les mêmes documents ont échoué à exactement une règle sur 146 du profil d'archivage. La page parle d'un résultat proche en nommant la règle, car 145 réussites ne constituent pas une conformité.

Et lorsque les licences de 644 crates Rust ont été vérifiées à chaque construction, la limite déclarée de l'audit était celle des métadonnées. Il lit la licence déclarée de chaque crate ; les déclarations peuvent être fausses ou incomplètes, et le contrôle ne prouve rien sur une crate mal déclarée. Il a volontairement été mis en échec deux fois, ce qui montre son fonctionnement ; aucun contrôle ne peut prouver l'honnêteté des déclarations qu'il lit.

Où les problèmes surviennent

Les défauts suivants ont un point commun : tous les contrôles qui ne les recherchent pas réussissent.

Chaque page annonce un changement après chaque déploiement. Un serveur envoyant l'heure de modification d'un fichier statique comme `Last-Modified`, ou un générateur datant ainsi un sitemap, annonce le moment de la copie. Or une extraction du dépôt, un export ou une construction de conteneur copie tous les fichiers. Les caches revalident tout et les robots n'apprennent rien. Tirez les dates de l'historique des versions ou d'un champ saisi par quelqu'un, et n'indiquez pas une date inconnue. Dans Hugo, les sources forment une liste lue dans l'ordre, où le nom du champ lui-même doit figurer :

```
enableGitInfo = true

[frontmatter]
date = ["date", ":git", ":fileModTime"]
lastmod = ["lastmod", ":git", ":fileModTime"]
```

Un titre ou un aperçu de lien affiche des caractères mal décodés. L'encodage décrit tous les autres octets ; un analyseur rencontrant le texte avant sa déclaration doit deviner. La norme HTML exige que l'élément déclaratif tienne entièrement dans les 1 024 premiers octets ; le mettre en tête de `<head>` ne laisse rien à deviner. Sur ce site, l'outil URL Inspection de Bing (inspection d'URL) a lu un jour le titre et les deux chaînes Open Graph en Latin-1, car un tiret cadratin du titre précédait la déclaration d'encodage. L'en-tête HTTP annonçait aussi UTF-8, mais un outil lisant les octets sans l'en-tête n'a que la page. La déclaration d'encodage est désormais le premier élément, et un contrôle le confirme sur chaque page construite.

Le même fait diffère selon l'endroit. Le titre vit dans `<title>`, `og:title`, les données structurées et, pour un PDF, les propriétés du document ; sa langue vit dans `lang`, `Content-Language` et `og:locale`. Saisies séparément, ces valeurs divergent et chaque consommateur en croit une autre. Calculez chaque valeur une seule fois, dans un modèle ou une fonction, puis faites lire cette valeur partout où elle est affichée.

Un fichier révèle davantage que son auteur ne voulait publier. Les photos portent l'appareil, l'heure et souvent la position ; les documents bureautiques, les noms d'auteurs, l'historique des modifications et les chemins des modèles ; une balise `generator` nomme le logiciel créateur de la page. Retirez à l'export tout ce que personne n'a choisi de publier et considérez chaque champ comme un texte qu'un inconnu lira.

Pour aller plus loin

- RFC 9110, section 6.3 : champs d'en-tête — le transport des métadonnées avant le contenu en HTTP et les rôles possibles d'un champ d'en-tête.
- RFC 9110, section 8.8.2 : Last-Modified — quand envoyer une date de modification et pourquoi remplacer une date future.
- Norme HTML : élément meta — les noms normalisés des métadonnées et les règles de déclaration d'un encodage.
- Termes de métadonnées DCMI — les quinze éléments Dublin Core et le vocabulaire construit autour.
- Protocole Open Graph — les quatre propriétés requises pour chaque page et les propriétés facultatives associées.

Glossaire

Choisissez selon vos contraintes réelles

Présentez un jeu de données représentatif, les questions auxquelles il doit répondre et le goulot d'étranglement actuel. Nous pouvons convenir de la manière de mesurer la qualité, le débit et la reprise.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Que sont les métadonnées ? » 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/metadata/