Structuring Page and Media Details for Better Search Visibility

Why structured data matters for media-rich pages

When you publish a page that contains images, videos, or interactive elements, search engines need more than the raw file name to understand what each asset represents. Structured data, built on formats like JSON-LD and the Schema.org vocabulary, gives you a reliable way to label every piece of content so that a search engine can interpret its role, its relationship to the page, and its relevance to a query. Without that layer of organisation, even the best-produced media can remain invisible to the systems that decide which results to show.

A well-planned structured data strategy begins with identifying the key entities on a page. For a typical article that includes a hero image and an embedded video, those entities might be Article, ImageObject, and VideoObject. Each one needs a set of properties that describes what it is, where it lives, and how it relates to the other items on the page. Getting this right means the difference between a search engine seeing a random collection of files and seeing a coherent, ranked page that deserves a rich result.

Step 1: Map the page's entity relationship

Before you write a line of JSON-LD, map out the relationship between the page and its media. Think of the page as a container. The Article entity holds the main content. The ImageObject and VideoObject entities are nested inside that article, each with its own properties. This entity relationship is what tells a search engine that the video belongs to the article and not to some other page.

To map it, list every media asset that appears on the page. For each one, note its file type, its purpose on the page, and any text that describes it. A product video that demonstrates a feature is a different entity from a background video that runs automatically. Treat each one separately. Once you have the list, decide which Schema.org types fit each asset. The core guide is the Schema.org documentation, but a few common types cover most cases: ImageObject for still images, VideoObject for video files, and AudioObject for sound files. For navigation elements such as breadcrumbs, use BreadcrumbList.

What to watch: It is tempting to reuse the same entity for two similar assets. If an image and a video appear in the same section, they are separate entities. Each needs its own node in the JSON-LD graph. Also watch for assets that are decorative rather than informative. A purely decorative background image can be marked with a null or omitted alt attribute, but it still benefits from a minimal ImageObject entry so the search engine knows it exists.

Step 2: Write the JSON-LD block for each entity

JSON-LD is the recommended format because it keeps the structured data separate from the HTML body. You place a single script tag in the head or body of the page, and inside it you write a JSON object that describes all the entities on the page. Each entity gets its own node, and the nodes are linked through the @graph array or through explicit references.

Start with the Article node. Give it a @type of Article, a name, a headline, a description, and a url. Then add the image and video nodes. For ImageObject, include properties such as url, caption, and contentUrl. For VideoObject, include url, name, description, thumbnailUrl, and contentUrl. If the video has a transcript, add a transcript property that points to a separate file or to a text node.

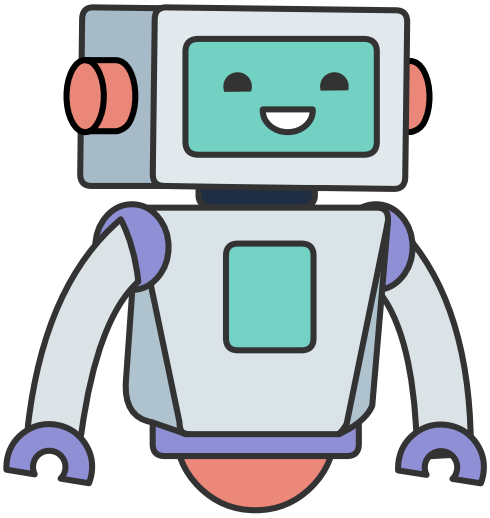
Here is the critical part: each media node must be linked to the article. In JSON-LD, you do this by setting the article's image property to the ImageObject node's @id, and its video property to the VideoObject node's @id. This creates the entity relationship that the search engine uses to understand the page structure.

What to watch: Validate the JSON-LD before publishing. Use a tool such as Google's Rich Results Test or Schema.org's validator. A single missing comma or an unescaped quote can break the entire block. Also watch for incorrect @id references. If the article node refers to a video node with a different @id, the connection is lost. Keep the @id values consistent across the whole block.

Step 3: Test for rich result eligibility

Once the JSON-LD is in place, test the page to see which rich results it qualifies for. Google's search algorithms use structured data to generate enhanced snippets, carousels, and knowledge panels. An article with a properly marked-up VideoObject can appear in a video carousel, while an article with ImageObject can show an image thumbnail in the search result. BreadcrumbList can produce a breadcrumb trail in the snippet, which helps users understand where the page sits in the site hierarchy.

To test, submit the page URL to the Rich Results Test. The tool shows which entities it detected and whether any errors exist. If the tool does not recognise a media entity, check the JSON-LD syntax and the entity relationship. Sometimes the issue is a missing property that the algorithm requires. For VideoObject, for example, Google expects a contentUrl or a embedUrl. Without one, the video is not considered eligible for a rich result.



What to watch: Rich result eligibility changes over time. What works today might not work next month. Keep an eye on Google's developer documentation for updates to the required properties. Also watch for false positives: a page that passes validation might still not get a rich result if the content is thin or if the site's overall authority is low. Structured data is one signal among many.

For a deeper look at how structured data fits into a broader media strategy, the team at Animation & Whatnot provides technical search and structured data approaches that help brands turn their media assets into organic acquisition channels. Their work covers the same entity relationship mapping and validation steps described here, applied across different types of media.

Step 4: Maintain the structured data as the page evolves

A page's media assets change over time. You might replace a video, add a new image, or update the article text. Each change means the JSON-LD block needs a corresponding update. If you add a video but forget to add its VideoObject node, the search engine will not know it exists. If you remove an image but leave its ImageObject node, the search engine might show a broken thumbnail.

Build a maintenance routine. Every time you publish a page update, run the Rich Results Test again. Check that all current media assets have matching entities and that the entity relationships still point to the correct nodes. A simple checklist can help: after any content change, verify that the number of media entities matches the number of visible media assets, that all @id references are valid, and that no orphaned nodes remain.

What to watch: Automated content management systems sometimes strip script tags during updates. If your JSON-LD block disappears after a CMS update, the page loses all its structured data. Store the JSON-LD in a separate file or in a custom field that the CMS does not touch. Also watch for duplicate entities. If two editors add structured data independently, you can end up with two Article nodes that conflict. Set a single source of truth for each page.

Bringing it all together

A structured data strategy that covers page and media details is not a one-time setup. It is a continuous process of mapping, writing, testing, and maintaining. Each step builds on the previous one. Without a clear entity relationship map, the JSON-LD will be inconsistent. Without validation, errors will slip through. Without maintenance, the data will drift out of sync with the page content.

The result of getting it right is a page that search engines can read with confidence. They see which image is the primary visual, which video is the main media asset, and how the breadcrumbs connect the page to the rest of the site. That clarity can lead to better rankings, more rich results, and a stronger presence in search results. The same approach applies to any site that relies on media to communicate its message, from news publishers to e-commerce stores to portfolio sites.

Animation & Whatnot, for example, applies this kind of structured data work to help brands make their video and image assets more discoverable. Their focus on technical search and visual search aligns with the principles outlined here: treat every media asset as a first-class entity, connect it clearly to the page, and verify the output regularly. The effort pays off in the form of organic traffic that comes from users who find exactly the media they are looking for.