

This document describes the logic used for structured data inside Rocket Stack Rank reviews. We use the structured defined by Schema.org as used by Google.

Quick Overview of Structured Data

The whole point of structured data is to provide more information to search engines so they can do a better job of reporting results. More prosaically, RSR's goal is to get Google to respect our ratings and show our star rankings in search results.

Structured data is much like the data in a C structure. We don't get to define new types of structure; we have to use the ones already defined in schema.org, which serves a bit like a header file that defines structures. The structured data parasitizes the HTML hierarchy. That is, we add it as elements inside of tags, but it pays almost no attention to the tags themselves. Where there isn't already HTML to use, we add either SPAN tags or META tags. **NOTE:** Due to a bug in the blogger editor, all META tags in a post are deleted whenever you switch to compose mode. Accordingly, empty span tags must be used instead. E.g. `` instead of `<meta itemprop="height" content="768">`.

With few exceptions, Google ignores structured data that the user cannot see, so while it's possible to use META tags to specify structured data, it's not usually a good idea. It's even possible that Google counts that against you, on the theory that you're trying to deceive them and the user.

Whenever we use a structure, most fields have defaults, but there are certain fields that we have to fill in. Google often has stricter requirements than Schema.org does.

To indicate we're using a particular schema (aka structure), we add two things to a tag:

```
itemscope=""
```

Itemscope is very important because it says "you can't use any of the fields inside this HTML to satisfy schemas external to it." Without this, instead of getting nested schemas, you get something more like blended schemas, and outer schemas can "steal" fields from inner ones.

```
itemtype="NAME OF SCHEMA"
```

This simply gives the name (from schema.org) of the schema we're going to use.

To indicate that we're providing a value to a field of the schema we're already inside of, we add

```
Itemprop="NAME OF FIELD"
```

For example, all of our reviews start with something like this:

```
<div itemref="blogPubDate blogAuthor blogPub" itemprop="mainEntity" itemscope=""  
itemtype="http://schema.org/Review">
```

Ignoring the itemref for the moment, this says that we are filling in the field "mainEntity" in whatever structure encloses us (Article, as it happens), that we're filling it with a schema of type Review, and that we are *not* filling in any *other* fields of Article.

External Structure

By modifying the blogger template, we've arranged that all of our blog posts are of type Article and, further, they are content for the articleBody field. Article requires fields for author, publisher, publication date, and image be filled in, and publisher requires that logo be filled in. All of this is accomplished inside the template. Further, the template gives names to those fields, so they can be reused inside posts. This is what the itemref field is doing in the example above: it imports definitions for key fields. (Anything with a unique id="..." field can be imported this way.) This saves a lot of work!

A more subtle point is that articleBody does not have itemScope, so even though our articles are inside of a text field, we can still define values for fields that are in the Article schema.

MainEntityOfPage

Because blog pages (or fragments of them) can turn up in search results (search results within the blog, not Google search results), there is value that identifies the canonical URL of the post. To accomplish this, put the following at the top of every post:

```
<span itemprop="mainEntityOfPage" content="BLOG_PAGE_URL"></span>
```

This is optional (Google generates a warning) and too tedious to do by hand, but as we generate the review pages programmatically, we might as well do this.

Blog Posts

Any blog post with zero structured data in it will automatically be marked up as containing the article body of a properly formatted Article. If you markup one or more images inside that post, they will be attached to the Article. (See the image schema below).

Review Structure

Review Schema

All reviews, whatever their topic, are wrapped as follows:

```
<div itemprop="blogPubDate blogAuthor blogPub" itemprop="mainEntity" itemscope=""  
itemtype="http://schema.org/Review">
```

```
.  
. .  
. . .
```

```
</div><!--mainEntity-->
```

Notes

- The post should contain nothing whatsoever outside the scope of this div.
- The itemref imports definitions for fields like publication date, author, publisher, etc. which are automatically computed inside the template from values that are actually displayed to the user.

The review schema requires that we define an image, a reviewRating, an itemReviewed, and a reviewBody. We'll take these in order.

Image Schema

Immediately after the div that starts the review, we have a figure tag that defines the image. To simplify things, I've elided other formatting information unrelated to the structure.

```
<figure itemprop="image" itemscope="" itemType="http://schema.org/ImageObject" ... >
.
.
.
</figure><!--image-->
```

Everything related to the image exists inside this figure.

The itemprop tells us that this figure is populating the image field of the Review schema, and the itemType says that we're filling it with a schema of type imageObject.

ImageObject Schema

This schema wants a URL for the big image, a URL for the thumbnail image, the text of the caption, and an indication of whether this image is worth showing with search results. The extra HTML for lightboxing and sizing obscures the formatting, so I've elided that to make the structured form more clear.

```
<span itemprop="representativeOfPage" content="true"></span>
<a itemprop="url" href="IMAGE.jpg" ... >
  
</a>
<figcaption itemprop="caption" ... >
  <span itemscope="" itemType="http://schema.org/Thing">
    <a itemprop="url" itemref="captionURL" href="...">Caption Text</a>.
  </span>
</figcaption>
```

Points to note

- This comprises everything contained inside the figure.
- The parsers for structured data are smart enough to figure out that the url in an A tag comes from the href element, while the url from an IMG tag comes from the src element. These are the only cases I know of where that shortcut works.
- It's okay to use an empty span here because there's no way to show the user that the image is representative of the page.

- Inside the caption, the SPAN with itemscope protects the caption URL from being assigned to the image. It's either the URL of a story or the URL of a book or issue.

Description Schema

The description goes inside the same div as the wordcount and time estimates. It fills in the *description* field of Review.

```
<div>
  <span itemprop="description">Description of Story</span>
  (WORDCOUNT AND TIME ESTIMATE)
</div>
```

- The word count and time estimates are not structured. (They apply to the thing reviewed, not to the review itself.)
- It's arguable that "description" is wrong because it's also a property of the thing reviewed, not the review itself. We could add the element id='storyDescription' to this tag and then import it into itemReviewed (below) via itemref, at the expense of applying the same description to both the review and the thing reviewed.

ReviewRating Schema

This is where the star ratings come from. We define two fields inside reviewRating: ratingValue and description.

```
<div itemprop="reviewRating" itemscope="" itemType="http://schema.org/Rating">Rating:
  <span itemprop="ratingValue" content="2">★★☆☆☆</span>,
  <span itemprop="description">Not Recommended</span>
</div><!--reviewRating-->
```

- By using the content field for ratingValue, we set the numeric value that Google requires. Although this violates their rule about scores being visible, we've determined that it doesn't impact their willingness to display stars for us. Likely this is a case they check for, since displaying a number is suboptimal.
- Because reviewRating has itemscope, this description can only be a description of what the review score means; there's no chance of it being used in some outer scope.

ItemReviewed Schema

This specifies the thing we're actually reviewing. Usually a story, but possibly an anthology or even a magazine or a book.

```
<div itemprop="itemReviewed" itemscope="" itemType="http://schema.org/CreativeWork">
  <a itemprop="url" href="STORY URL IF IT EXISTS">
    <span itemprop="name">TITLE OF THE STORY</span>
```


Notes:

- Even though name is defined inside the scope of the A tag that defines url, both name and url are fields of itemReviewed. This, again, is because the A tag doesn't have itemscope.
- Instead of an empty span for the author's homepage, we could use a real hyperlink.
- Multiple editors and translators are handled the same as multiple authors.

IsPartOf schema NOT used

There are four different ways to do the IsPartOf markup. In some cases, such as tor.com and an anthology review, the thing reviewed is an entity whole; it's not a part of anything larger. In other cases, the publisher is the same as the magazine, and we don't want to be saying "Clarksword is published by Clarksword."

1 Not Used

When the thing reviewed isn't part of anything larger, we just use:

```
<span itemref="captionURL" itemprop="isPartOf" itemscope=""
itemtype="http://schema.org/PublicationIssue">appeared in
  <i itemprop="name">MAGAZINE_NAME</i> issue
  <span itemprop="issueNumber">ISSUE_NUMBER</span>, published on
  <span itemprop="datePublished">PUBLICATION_DATE</span>
  <span itemprop="publisher" itemscope="" itemtype="http://schema.org/Organization"> by
    <a itemprop="url" href="PUBLISHER_URL"><span
  itemprop="name">PUBLISHER_NAME</span></a>.
  </span><!--publisher-->
</span><!--isPartOf-->
```

- As above, the publisher doesn't have to be a hyperlink; it can be an empty span.
- The datePublished and publisher fields of itemReviewed are filled in here.

2 IsPartOfSchema: Publication Title <> Publisher Title

For a publication whose publisher is different from the title of the publication, we format it like so:

```
<span itemprop="isPartOf" itemscope="" itemtype="http://schema.org/PublicationIssue">appeared
in
  <i itemprop="name">MAGAZINE_NAME</i> issue
  <span itemprop="url" content="ISSUE_URL"></span>
  <span itemprop="issueNumber">ISSUE_NUMBER</span>, published on
  <span itemprop="datePublished">PUBLICATION_DATE</span>
  <span itemprop="publisher" itemscope="" itemtype="http://schema.org/Organization"> by
```

```
<a itemprop="url" href="PUBLISHER_URL" ><span
itemprop="name">PUBLISHER_NAME</span></a>.
```

```
</span><!--publisher-->
```

```
</span><!--isPartOf-->
```

3 IsPartOfSchema: Publication Title = Publisher Title

When the publication and publisher have the same name, we want to avoid saying “X was published by X” so we use this form instead:

appeared in

```
<i id="pubname" itemprop="name">MAGAZINE_NAME</i> issue
```

```
<span itemprop="url" content="ISSUE_URL"></span>
```

```
<span itemprop="issueNumber">ISSUE_NUMBER</span>, published on
```

```
<span itemprop="datePublished">PUBLICATION_DATE</span>
```

```
<span itemprop="publisher" itempropref="pubname" itemscope=""
itemtype="http://schema.org/Organization">
```

```
<span itemprop="url" content="PUBLISHER_URL"></span>
```

```
</span><!--publisher-->
```

```
</span><!--isPartOf-->
```

The portions in red are what differ between the two

- Even when the story is part of an anthology we still say it belongs to a publication issue, owing to deficiencies in schema.org and Google’s interpretation of it.
- IssueNumber can be “issue name”. E.g. “April 2016”.
- The issue url duplicates the URL used with the caption of the image. We cannot import it via itemprop because there’s no way to mark it up without it getting confused for the url of the image itself.
- In the case where the publisher and magazine have the same name, there’s no alternative to using an empty span.
- A story in an anthology is the same as case 1 except there is no issueNumber. It’s a defect that we have to call an anthology a publication issue, but schema.org/Google give us no choice.

4 IsPartOfSchema: Publication does not have an Issue

When the publication such as an anthology does not have an issue, we leave out the issue data and add an RSR review link (if available):

appeared in

```

<i itemprop="name">MAGAZINE_NAME</i> (<a href="RSR_REVIEW_URL">RSR review</a>)
issue
<span itemprop="url" content="ISSUE_URL"></span>
<span itemprop="issueNumber">ISSUE_NUMBER</span>, published on
<span itemprop="datePublished">PUBLICATION_DATE</span>
<span itemprop="publisher" itemscope="" itemType="http://schema.org/Organization"> by
  <a itemprop="url" href="PUBLISHER_URL"><span
  itemprop="name">PUBLISHER_NAME</span></a>.
</span><!--publisher-->
</span><!--isPartOf-->

```

Note: Scenario #4 could also be scenarios #2 or #3.

ReviewBody

The last item is the body of the review itself.

```

<div class="spoilers" id="content" itemprop="reviewBody">
  <b>Pro</b>: <br /><br />
  <b>Con</b>: <br /><br />
  Other Reviews: . . .
</div><!--reviewBody-->

```

- Any special notes and any specific recommendations from other reviewers end up outside the markup entirely. A case could be made for wrapping reviewBody around the entire page. Since it doesn't have itemscope, the other fields would still be filled in "through" it.
- The spoilers class is what tells the browsers to display the minireview upside down.

Putting it all Together

Here's what it all looks like, including some of the "glue" HTML code omitted above. Items in red are generated inside the template and don't go into the review directly.

```
<body itemscope="" itemType='http://schema.org/Article'>
```

Template sets headline, image, url, datePublished, dateUpdated, publisher, and author fields for the Article schema.

```

<div itemprop='articleBody'>
  <span itemprop="mainEntityOfPage" content="BLOG_PAGE_URL"></span>
  <div itemref="blogPubDate blogAuthor blogPub" itemprop="mainEntity" itemscope=""
  itemType="http://schema.org/Review">

```


The image figure element

```
<div>
```

The description span. This is the description for the Review schema

```
</div>
```

The reviewRating div element

A possible note that's not structured.

A possible reviewer recommendation that's not structured.

```
<br style="clear: left;" />
```

```
<!--more-->
```

```
<div itemprop="itemReviewed" itemscope=""  
itemtype="http://schema.org/CreativeWork">
```

*The story, author(s), translator(s), editor(s), magazine/anthology, issue, publish date,
publisher*

```
</div><!--itemReviewed-->
```

```
<h3 style="margin-top: 1em;">
```

```
<a href="javascript:expandcollapse()">Mini-Review (click to view--possible spoilers)</a>
```

```
</h3>
```

```
<div class="spoilers" id="content" itemprop="reviewBody">
```

The reviewBody div

```
</div><!--reviewBody-->
```

```
</div><!--mainEntity-->
```

```
</div><!--articleBody-->
```

```
</body>
```

And that's it!

Examples

[The Warrior Boy Who Would Not Suffer](#), by Abhinav Bhat, is a good example of a review where the publisher and title are the same, and I've fully updated the HTML.

[Cold-Forged Flame](#), by Marie Brennan, is a good example of one where there is no IsPartOf.