

# GitHub pages with Bioschemas markup

---

These GitHub pages include a page describing a [software](#) and a page describing a [dataset](#) that was created as an output from the software at hand.

## Overview

---

- **Name:** Tutorial on adding Bioschemas markup to GitHub pages
- **Description:** This tutorial shows how to add Bioschemas markup to GitHub pages. It uses a simple GitHub page hosted in the gh-pages branch to create a sample project page, i.e., as learners could do with their own GitHub projects. As an example, it uses the software and data corresponding to the released project [TREC-doc-2-doc-relevance](#), a web-based interface to add document-to-document relevance assessments to pairs of documents retrieved from [TREC 2005 Genomics Track](#).
- **Keywords:** Bioschemas, [schema.org](#), structured markup, GitHub pages

### ❓ Questions

- How can I add Bioschemas markup to GitHub pages?
- Can I also use [schema.org](#) types not included in Bioschemas?
- How can I use Bioschemas profiles? In particular [ComputationalTool](#) and [Dataset](#) (bonus, we also use [TrainingMaterial](#) to describe the tutorial itself)

### 🎯 Learning outcomes

- Describe how [schema.org](#) and Bioschemas markup can be embedded to GitHub pages
- Use [schema.org](#) and Bioschemas profiles on GitHub pages
- Use schema and Bioschemas validators

### ✅ Requirements

- Understanding of what is Bioschemas and what are Bioschemas types and profiles
- Familiarity on how to use GitHub
- Basic knowledge on how to use GitHub pages. More information at [GitHub Pages](#)
- Familiarity with JSON-LD

- Basic knowledge on Markdown
- Knowledge of develop tools on a browser

 **Time estimation** 30 minutes

 **Level** Beginner / Introductory


 **Published** 2024-02-05

 **Latest modification** 2024-09-20

 **License** [CC-BY 4.0](#)

 **Version** 1.1.0

 **Identifier** [DOI:10.5281/zenodo.13799121](https://doi.org/10.5281/zenodo.13799121)

 **Citation** Castro, L.J. (2024). Adding Bioschemas Dataset and ComputationalTool markup to GitHub pages. Zenodo.  
<https://doi.org/10.5281/zenodo.13799121>

## Learning experience

---

### Agenda

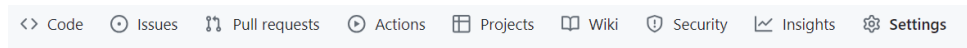
In this tutorial we will cover:

- [GitHub pages with Bioschemas markup](#)
  - [Overview](#)
  - [Learning experience](#)
    - [Agenda](#)
    - [Creating this GitHub Page](#)
    - [Adding schema.org and Bioschemas markup](#)
      - [Using schema.org types](#)
      - [Using Bioschemas profiles](#)
    - [Visualizing the structured markup](#)
    - [Validating the pages against the Schema Validator](#)
    - [Validating the pages against the Rich Results Test](#)
    - [Validating the pages against FAIR-Checker/Bioschemas validator](#)
    - [Creating a Sitemap](#)
    - [Try it out](#)
  - [What is next?](#)

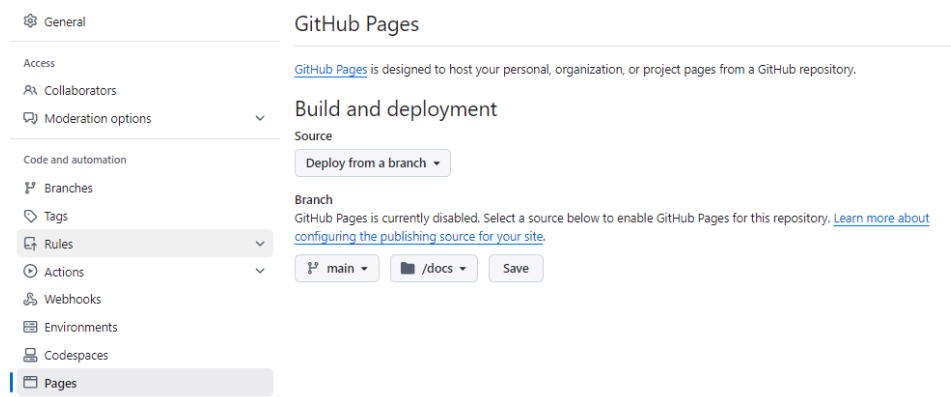
- [Acknowledgements](#)

## Creating this GitHub Page

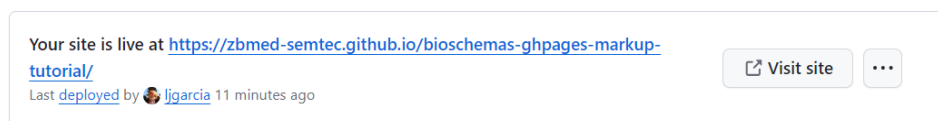
Let's start by forking [this repository](#) for your own purposes. Once forked, go to settings



You will need to enable "Pages" on your forked repository, and select under Source Deploy from a branch. Use the main branch and the folder /docs. Save your changes.



In a matter of minutes, your site will be live. The pages corresponding to the examples used in this tutorial are available at <https://zbmed-semtec.github.io/bioschemas-ghpages-markup-tutorial/>. They will have no markup to start with, you will add it in a moment.



Do not forget to get a local copy of your fork so you can make changes.

## Adding [schema.org](#) and Bioschemas markup

We will add [schema.org](#) and Bioschemas markup corresponding to our example, code (software page) and data (dataset page) for our repo [TREC-doc-2-doc-relevance](#). Right now those pages only have text, with the following steps you will get [schema.org/Bioschemas](#) markup embedded in your pages.

With GitHub pages, same as with regular HTML (static or generated), the trick is adding the JSON-LD describing those elements that you want. To do so, you need to add a block `<script type="application/ld+json">` with the Bioschemas markup as a JSON-LD inside the block.

The Bioschemas markup will start with the context "`@context`": "`https://schema.org`", i.e., where all types and profiles are defined.

Then you add the type and id of what you are describing, for instance a `bioschemas:ComputationalTool`. The "`http://purl.org/dc/terms/conformsTo`" is the bit that indicates that you are using Bioschemas.

```
"@type": "SoftwareApplication", "@id": "https://doi.org/10
```

For the rest of the markup, you need to see what the Bioschemas profile recommends ([keep reading, more information below](#)).

To get the markup added to your pages, copy the content of [software](#) to the [software page under /docs folder](#). Then copy the content of [dataset](#) to the [dataset page under /docs folder](#)

Have a look to the pages that you just created on your own GitHub pages, one for the [code and corresponding release](#), another for a [dataset created from data collected by the software](#). To see them working for your repo (that will now have the [schemas.org/Bioschemas](#) markup), you need to use `<your_user>/<your_repo>` instead of `zbmed-semtec.github.io/bioschemas-ghpages-markup-tutorial` on the html link `https://zbmed-semtec.github.io/bioschemas-ghpages-markup-tutorial/dataset`.

The [software page](#) in your repo should now have markup for two elements, `schema:SoftwareSourceCode` and `bioschemas:ComputationalTool`.

The markup corresponding to `schema:SoftwareSourceCode` is shown below. Note that the *source code* is linked to the *release* via the property `schema:targetProduct`. In the code below, we are using the *release @id* to refer to it.

```
<script type="application/ld+json"> { "@context": "https
```

Note that we use a trick for the `@id` and the `schema:url` in our markup. Identifiers should be unique, global and permanent, like e.g., DOIs, and clearly identify the object they describe (in this case a source code). But, we do not have such a PID for our source code, we only have the GitHub URL for our repo. Now, if the `@id` is meant to identify a source code, we cannot use the exact same URL for the source code and for a "regular" url. We add # to the "regular" URL. If we do not, validators will see that the exact same URL is used to identify a source code and a "regular" URL but source codes and URLs are not the same thing so the validators might get confused. Using the exact same URL would be like saying that an animal is a cat at the same time that it is a dog, it cannot be. The best way to go would be getting an actual PID for your repo, e.g., via [w3id.org](https://w3id.org)

And here you have the markup corresponding to the `bioschemas:ComputationalTool`

```
<script type="application/ld+json"> {   "@context": "https
```

The [dataset page](#) in your repo should now have markup only for one element, a dataset.

```
<script type="application/ld+json">   {   "@context": "http
```

### Using [schema.org](https://schema.org) types

If you use a [schema.org](https://schema.org) type that has a corresponding Bioschemas profile, we recommend using the Bioschemas profile as it will help you focus on the most common and useful properties. However, if there is a type in [schema.org](https://schema.org) that does not have a Bioschemas profile, we suggest finding examples on how others have used them for a case similar to yours.

In this case we have used `schema:SoftwareSourceCode` with some properties that are also present for `bioschemas:ComputationalTool` and that will provide some minimum information to anyone willing to reuse our source code, including citation information.






### Using Bioschemas profiles

Whenever you use a Bioschemas profile, go to the corresponding page, we have used:

- [bioschemas:ComputationalTool](#), and
- [bioschemas:Dataset](#)

The minimum information that you should provide are the "minimum" properties, and, whenever possible, it is a good idea to also provide the "recommended" ones. Optional properties are up to you. Another important piece of information is the cardinality, whether or not you could use a single *object* or an *array*.

You can find examples and other useful links for each Bioschemas profile

Description	Contributors	Links			
Group	Use Cases	Cross Walk	Task & Issues	Examples	Live Deploys
<a href="#">Tools</a>					

And also [live deploys](#) using the profile

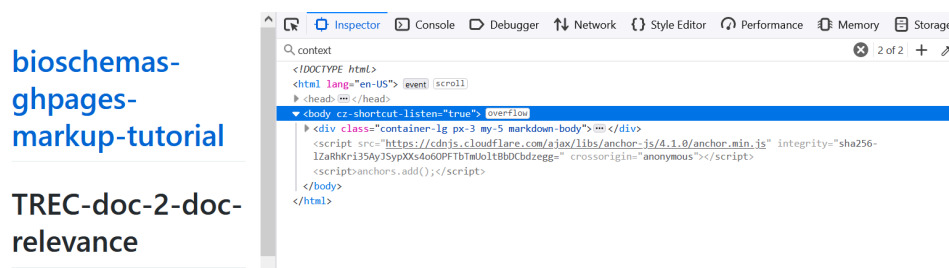
Sites: 8	ComputationalTool (1.0-RELEASE)
BAR 3.0 <b>v0.1</b>	Page SMV JSON-LD FAIR-Checker
bio.tools <b>v1.0-RELEASE</b> 22,687 tools	Page SMV JSON-LD FAIR-Checker
BridgeDb <b>v1.0-RELEASE</b>	Page SMV JSON-LD FAIR-Checker
Cscan <b>v0.1</b>	Page SMV JSON-LD FAIR-Checker
PathVisio <b>v1.0-RELEASE</b>	Page SMV JSON-LD FAIR-Checker
Pscan <b>v0.1</b>	Page SMV JSON-LD FAIR-Checker
PeranChIP <b>v0.1</b>	Page SMV JSON-LD FAIR-Checker

If you use Bioschemas markup on your website, you can also add it to the "live deploy" list. TODO

## Visualizing the structured markup

Once the markup is in place and the page is live, you can use the developers tools in a browser to visualize the markup. You can use the inspect tool in e.g., Chrome, Firefox or Edge.

Go to the [software page](#) and use the inspect tool for any element. You should see the HTML.



With CTRL+F(or command key + F on Macs) to search and find context. The first one is added by GitHub pages, and describes the page as a `schema:WebPage`.

```
<script type="application/ld+json">
  { "@context": "https://schema.org", "@type": "WebPage", "description": "A tutorial on how to add Bioschemas markup to GitHub pages", "headline": "TREC-doc-2-doc-relevance", "url": "https://zbmed-semtec.github.io/bioschemas-ghpages-markup-tutorial/software.html" }
</script>
```

The second occurrence will be the markup corresponding to the `schema:SoftwareSourceCode` and `bioschemas:ComputationalTool` manually added to the block `<script type="application/ld+json">`.

```
<script type="application/ld+json">
  {
    "@context": "https://schema.org",
    "@type": "SoftwareSourceCode",
    "@id": "https://github.com/zbmed-semtec/TREC-doc-2-doc-relevance",
    "citation": "Talha M, Geist L, Fellerhof T, Ravinder R, Giraldo O, Rebholz-Schuhmann D, et al. TREC-doc-2-doc-relevance [Software source code]. GitHub; 2022.",
    "name": "TREC-doc-2-doc-relevance",
    "description": "This is the software source code facilitating the creation of a doc-2-doc relevance assessment on PMIDs used in the TREC 2005 Genomics track along with its metadata.",
    "url": "https://github.com/zbmed-semtec/TREC-doc-2-doc-relevance#",
    "targetProduct": {
      "@type": "SoftwareApplication",
      "@id": "https://doi.org/10.5281/zenodo.7341391",
      "http://purl.org/dc/terms/conformsTo": "https://bioschemas.org/profiles/ComputationalTool/1.0-RELEASE",
      "identifier": "DOI:10.5281/zenodo.7341391",
      "citation": "Talha M, Geist L, Fellerhof T, Ravinder R, Giraldo O, Rebholz-Schuhmann D, et al. TREC-doc-2-doc-relevance assessment interface. Zenodo; 2022. doi:10.5281/zenodo.7341391",
      "name": "TREC-doc-2-doc-relevance assessment interface",
      "description": "The code, data and docs at this release aim at facilitating the creation of a doc-2-doc relevance assessment on PMIDs used in the TREC 2005 Genomics track. A doc-2-doc relevance assessment takes one document as reference and assess a second document regarding its relevance to the reference one. This doc-2-doc collection will be used to evaluate the doc-2-doc recommendations approaches that we are working on.",
      "url": "https://zenodo.org/records/7341391",
      "softwareVersion": "1.0.0",
      "datePublished": "2022-11-21",
      "license": {
        "@type": "CreativeWork",
        "@id": "https://spdx.org/licenses/MIT.html",
        "name": "MIT License",
        "url": "https://opensource.org/licenses/mit/"
      },
      "author": [
        {
          "@id": "https://zbmed-semtec.github.io/previous_members/#muhammad-talha",
          "@id": "http://orcid.org/0000-0002-2910-7982"
        },
        {
          "@id": "https://orcid.org/0000-0002-8725-1317",
          "@id": "https://orcid.org/0009-0004-4484-6283"
        },
        {
          "@id": "https://orcid.org/0000-0003-2978-8922",
          "@id": "https://orcid.org/0000-0002-1018-0370"
        },
        {
          "@id": "https://orcid.org/0000-0003-3986-0510"
        }
      ],
      "license": {
        "@type": "CreativeWork",
        "@id": "https://spdx.org/licenses/MIT.html",
        "name": "MIT License",
        "url": "https://opensource.org/licenses/mit/"
      },
      "author": [
        {
          "@id": "https://zbmed-semtec.github.io/previous_members/#muhammad-talha",
          "@id": "http://orcid.org/0000-0002-2910-7982"
        },
        {
          "@id": "https://orcid.org/0000-0002-8725-1317",
          "@id": "https://orcid.org/0009-0004-4484-6283"
        },
        {
          "@id": "https://orcid.org/0000-0003-2978-8922",
          "@id": "https://orcid.org/0000-0002-1018-0370"
        },
        {
          "@id": "https://orcid.org/0000-0003-3986-0510"
        }
      ]
    }
  }
</script>
```

If you look closely, you will notice that we have the element `bioschemas:ComputationalTool` fully inside the element `schema:SoftwareSourceCode` rather than a reference via the `"@id": "https://doi.org/10.5281/zenodo.7341391"` as we did above. Both ways are valid.

This is how we have embedded the Bioschemas markup on the page

```
<script type="application/ld+json"> { "@context": "https
```

## Validating the pages against the Schema Validator


Go to [Schema validator](#). Add the URL for either the [software page](#) or the [dataset page](#) and click on "Run Test"

Test your structured data

×

FETCH URL

CODE SNIPPET


<https://zbmed-semtec.github.io/bioschemas-ghpages-markup-tutorial/software>

RUN TEST

In the case of the [software page](#), it identifies and validates the two upper objects corresponding to `schema:WebPage` and `schema:SoftwareSourceCode` but it does not find `bioschemas:ComputationalTool` (which corresponds to a `schema:SoftwareApplication`)

You can run the test again with a "Code Snippet" by copying and pasting the code embedded on the page, the results will be the same. However, if you try and validate a code snippet containing only one element, the inner element `bioschemas:ComputationalTool`, the Schema Validator will identify and validate it (tested on 2024-02-05).

```

1 {
2   "@context": "https://schema.org",
3   "@type": "SoftwareApplication",
4   "@id": "https://doi.org/10.5281/zenodo.7341391",
5   "http://purl.org/dc/terms/conformsTo": "https://bioschemas.org/profiles/ComputationalTool/1.0-RELEASE",
6   "identifier": "DOI:10.5281/zenodo.7341391",
7   "citation": "Talha M, Geist L, Fellerhof T, Ravinder R, Giraldo O, Rebholz-Schuhmann D, et al. TREC-doc-2-doc-relevance assessment interface. Zenodo; 2022. doi:10.5281/zenodo.7341391",
8   "name": "TREC-doc-2-doc-relevance assessment interface",
9   "description": "The code, data and docs at this release aim at facilitating the development of a tool for the TREC-doc-2-doc-relevance assessment interface.",
10  "url": "https://zenodo.org/records/7341391",
11  "softwareVersion": "1.0.0",
12  "datePublished": "2022-11-21",
13  "license": {
14    "@type": "CreativeWork",
15    "@id": "https://spdx.org/licenses/MIT.html",
16    "name": "MIT License",
17    "url": "https://opensource.org/licenses/mit/"
18  },
19  "author": [
20    {
21      "@id": "https://zbmed-semtec.github.io/previous_members/#muhammad-talha",
22      "@type": "Person",
23      "name": "Muhammad Talha",
24      "url": "https://zbmed-semtec.github.io/previous_members/#muhammad-talha"
25    },
26    {
27      "@id": "https://orcid.org/0000-0002-2910-7982",
28      "@type": "Person",
29      "name": "Lukas Geist",
30      "url": "https://orcid.org/0000-0002-2910-7982"
31    },
32    {
33      "@id": "https://orcid.org/0000-0002-8725-1317",
34      "@type": "Person",
35      "name": "Talha M. Geist",
36      "url": "https://orcid.org/0000-0002-8725-1317"
37    },
38    {
39      "@id": "https://orcid.org/0009-0004-4484-6283",
40      "@type": "Person",
41      "name": "Ravinder R. Giraldo",
42      "url": "https://orcid.org/0009-0004-4484-6283"
43    },
44    {
45      "@id": "https://orcid.org/0000-0001-9058-9136",
46      "@type": "Person",
47      "name": "Dorothea Rebholz-Schuhmann",
48      "url": "https://orcid.org/0000-0001-9058-9136"
49    },
50    {
51      "@id": "https://orcid.org/0000-0001-9058-9136",
52      "@type": "Person",
53      "name": "Dorothea Rebholz-Schuhmann",
54      "url": "https://orcid.org/0000-0001-9058-9136"
55    }
56  ]
57 }

```

SoftwareApplication

All (1)

SoftwareApplication

0 ERRORS 0 WARNINGS

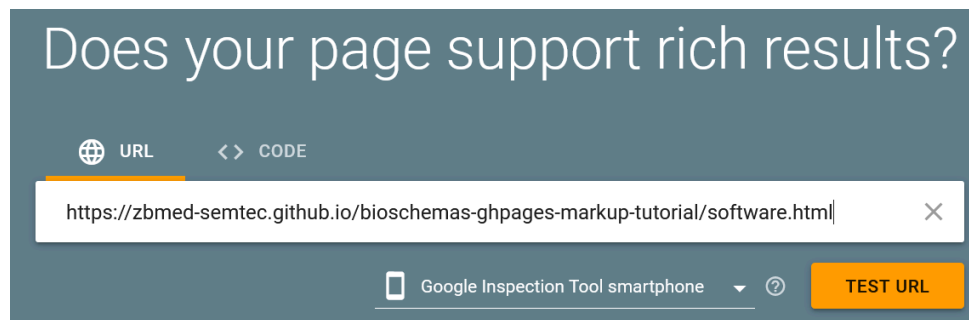
ID: https://doi.org/10.5281/zenodo.7341391

@type	SoftwareApplication
@id	https://doi.org/10.5281/zenodo.7341391
http://purl.org/dc/terms/conformsTo	https://bioschemas.org/profiles/ComputationalTool/1.0-RELEASE
identifier	DOI:10.5281/zenodo.7341391
citation	Talha M, Geist L, Fellerhof T, Ravinder R, Giraldo O, Rebholz-Schuhmann D, et al. TREC-doc-2-doc-relevance assessment interface. Zenodo; 2022. doi:10.5281/zenodo.7341391

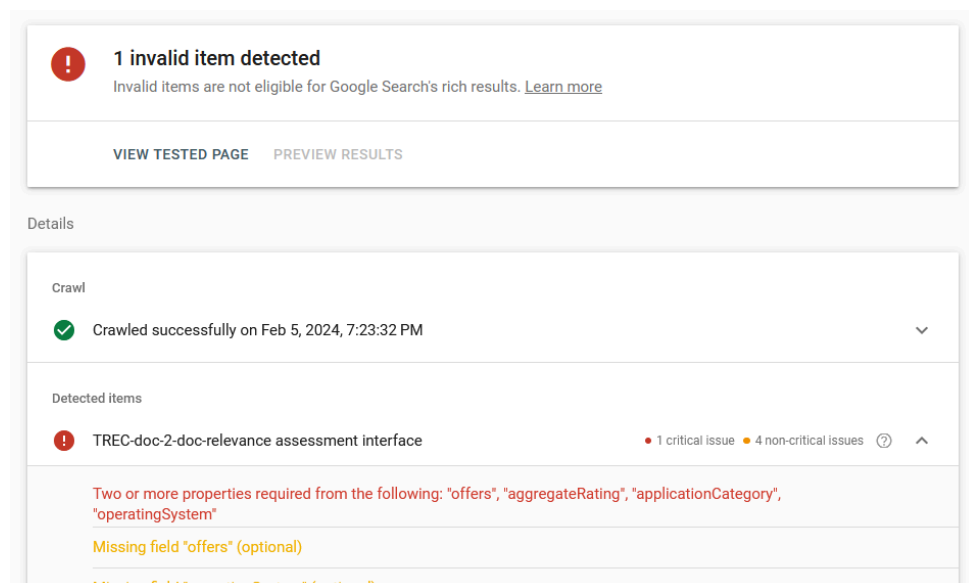
## Validating the pages against the Rich Results Test

There is another [schema.org](#) validator, currently the one recommended by Google. Go to [Rich Results Test](#) and use the same URL as before, and click "Test URL".

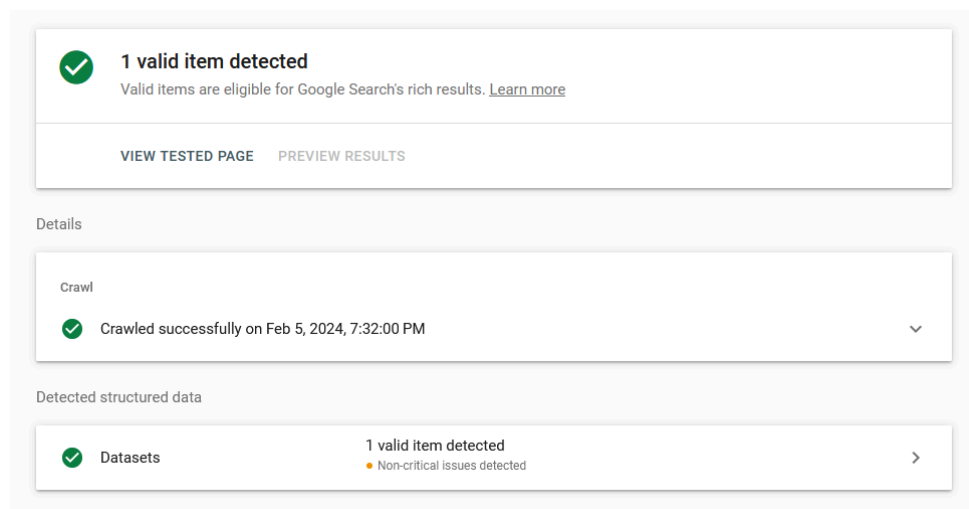




If you try the [software page](#), it will identify some errors with `schema:SoftwareApplication`. Yes, it does identify the inner element! As there is an error, only the error will be shown, nothing regarding the other elements. It looks like our `bioschemas:ComputationalTool` is missing some important elements (for Google, and you will see also for Bioschemas).



If you try the [dataset page](#), one correct element will be identified.



Did you notice that neither the Schema Validator nor the Rich Results Test pay attention to the `schema:WebPage` element?


## Validating the pages against FAIR-Checker/Bioschemas validator

Let's try now a validator specific for Bioschemas. Go to [FAIR-Checker](#) and click on the blue button "Inspect".

On Step 1, add the URL for either the [software page](#) or the [dataset page](#) and click on "Build Knowledge Graph"

**Step 1: fetch RDF metadata from the resource URL**

Examples: [Dataset Dataserve](#) [Workflow](#) [Publication Dataserve](#) [Dataset](#) [Tool](#)

  [Build Knowledge Graph](#)

Once you can see that the Knowledge Graph is ready in Step 2

**Step 2: Enrich Graph**

[... from Wikidata](#) [... from OpenCitations](#) [... from OpenAIRE](#)


```
@prefix dct: <http://purl.org/dc/terms/> .
@prefix ns1: <https://zbmed-semtec.github.io/bioschemas-ghpages-markup-tutorial/software.html#> .
@prefix ns2: <http://ogp.me/ns#> .
@prefix ns3: <https://dev.twitter.com/cards#> .
@prefix sc: <http://schema.org/> .

ns1:html {
  <https://github.com/zbmed-semtec/TREC-doc-2-doc-relevance> a sc:SoftwareSourceCode ;
  scauthor <http://orcid.org/0000-0002-2910-7982>,
    <https://orcid.org/0000-0002-1018-0370>,
    <https://orcid.org/0000-0002-8725-1317>,
    <https://orcid.org/0000-0003-2978-8922>,
    <https://orcid.org/0000-0003-3986-0510>,
    <https://orcid.org/0009-0004-4484-6283>,
    <https://zbmed-semtec.github.io/previous_members/#muhammad-talha> ;
  sc:citation "Talha M, Geist L, Fellerhof T, Ravinder R, Giraldo O, Rebholz-Schuhmann D, et al. TREC-doc-2-doc-relevance [Software source code].
  GitHub. 2023." ;
}
```

<https://zbmed-semtec.github.io/bioschemas-ghpages-markup-tutorial/software.html#html: 45 Triples>

**Jump directly to step 3**, select "Bioschemas" and click "Check Bioschemas"

### Step 3: Metadata quality checks

← Controlled vocabularies     Bioschemas

Bioschemas is a community effort aimed at reusing and extending Schema.org for better life science digital resource findability. Several profiles are defined for each kind of Life Science resources, specifying minimal, recommended or optional information. Are minimal information missing ? Should other information be provided for better findability ?

Check BioSchemas

Only Bioschemas profiles will be checked so, in this case `bioschemas:ComputationalTool` and `bioschemas:Dataset` but not `schema:SoftwareSourceCode`.

`https://doi.org/10.5281/zenodo.7341391` has type `http://schema.org/SoftwareApplication`  
Using `https://bioschemas.org/profiles/ComputationalTool/1.0-RELEASE` for validation, specified from the `dct:conformsTo` property.

#### Required missing properties

Congratulation,

`https://doi.org/10.5281/zenodo.7341391`

has a **valid** Bioschemas profile !

#### Improvements

`https://schema.org/applicationCategory` **should be** provided

`https://schema.org/applicationSubCategory` **should be** provided

`https://schema.org/featureList` **should be** provided

We are missing some of the *recommended* properties and the validator indicates this as *improvements*.

## Creating a Sitemap

If you want to make it easier for machines to crawl your website (and this could be a good idea when you have markup and you want to enable easy integration into aggregators), consider adding a `sitemap.xml` to your website.

We have used [XML-Sitemaps](#) to create [the one for the GitHub pages used in this tutorial](#).

Our sitemap looks like

```
<?xml version="1.0" encoding="UTF-8"?><urlset xmlns="htt
```

## Try it out

Try and modify the markup to describe a source code of your own. If you have not created a release yet, use only `schema:SoftwareSourceCode`. If you have a release or if you prefer, you can omit the markup for the source code and use only `bioschemas:ComputationalTool` for your release. Or you can provide some markup describing a dataset you have created for this you would use `bioschemas:Dataset`.

Are you interested in using Bioschemas markup for real on your web pages? [Join the community](#), you can send questions to the mailing list or more informally on the Slack channel.

## What is next?

---

We covered a manual addition of [schema.org](#) and Bioschemas markup to GitHub pages. However, websites many times are created by a production pipeline. In those cases, markup creation should be integrated to it.

We also briefly mentioned sitemaps with much of a context. We invite to get more information on what they are and why they are useful.

## Acknowledgements

---

This tutorial is based on [bioschemas-github-markup-example](#) and [Adding schema.org to a GitHub Pages site](#).

LJC has received fundings from the [German Research Foundation \(DFG\)](#) via the grant for NFDI4DataScience No. [460234259](#)

We use free SVG icons from [Font Awesone](#)