Schema.org Blueprints Module for Drupal
Hello! I am Jacob Rockowitz

I want to make your life easier, by solving complex problems with simple, well thought out solutions.

You can find me at: @jrockowitz
Background

What is the story behind this module?
OVER THE PAST THREE YEARS

Attended Decoupled Drupal in 2019

Attempted to migrate from Drupal to Sitecore

Realized, I was building monolithic applications
WROTE A FEW BLOG POSTS

April 11, 2022
The future of our Drupal CMS and Schema.org: APIs, UI, and UX
https://www.jrockowitz.com/blog/drupal-schema-api-ui-ux

October 06, 2020
Using a Schema.org-First approach to build a single source of truth and a unified Content Management System
https://www.jrockowitz.com/blog/schema-org-first

September 01, 2020
Exploring a Schema-First approach to Drupal and Content Management Systems
https://www.jrockowitz.com/blog/schema-first
Architecting emergent systems requires an evolution from strategic planning to collaborative strategic thinking — everyone seeing the parts through the lens of the whole.

https://decoupleddays.com/session/content-systems-architecture-approaches-decoupled-world
DETERMINED THAT I NEED TO...

- **Switch** to a polythetic approach
- **Explore** headless/decoupled Drupal
- **Audit** existing content models
- **Acknowledge** that Schema.org as the de facto data model for the web
MONOLITHIC VS POLYLITHIC
Leveraging Schema.org within Drupal is not a new idea.

- RDF UI
  https://www.drupal.org/project/rdfui
- Entity Fields Builder
  https://www.drupal.org/project/efb

Schema.org is another way for us to get off our individual content model and API islands.
1

Schema.org

What is Schema.org?
Schema.org is a collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on web pages, in email messages, and beyond.

https://schema.org
Over 10 million sites use Schema.org to markup their web pages and email messages.

https://schema.org
SCHEMA.ORG

The schemas are a set of 'types', each associated with a set of properties. The types are arranged in a hierarchy.

- 797 Types
- 1453 Properties
- 14 Datatypes
- 86 Enumerations
- 462 Enumeration members
Welcome to Schema.org

Schema.org is a collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on web pages, in email messages, and beyond.

Schema.org vocabulary can be used with many different encodings, including RDFa, Microdata and JSON-LD. These vocabularies cover entities, relationships between entities and actions, and can easily be extended through a well-documented extension model. Over 10 million sites use Schema.org to markup their web pages and email messages. Many applications from Google, Microsoft, Pinterest, Yandex and others already use these vocabularies to power rich, extensible experiences.

Founded by Google, Microsoft, Yahoo and Yandex, Schema.org vocabularies are developed by an open community process, using the public-schemaorg@w3.org mailing list and through GitHub.

A shared vocabulary makes it easier for webmasters and developers to decide on a schema and get the maximum benefit for their efforts. It is in this spirit that the founders, together with the larger community have come together - to provide a shared collection of schemas.

We invite you to get started!

View our blog at blog.schema.org or see release history for version 14.0.
2

Schema.org-first

What is Schema.org-first approach?
A Schema.org-first approach leverages Schema.org as the foundation for building content models using structured data, which is API-first, standardized and universal.

This make it easier for organizations to author and distribute content to multiple channels.
Schema.org Metatag module extends the Metatag module to display structured data as JSON-LD. Properties are hard-coded or uses token replacements.
Metatag

Configure global meta tag default values below. Meta tags may be left as the default. Meta tag patterns are passed down from one level to the next unless they are overridden. To view a summary of the individual meta tags and the pattern for a specific configuration, click on its name below.

If the top-level configuration is not specific enough, additional default meta tag configurations can be added for a specific entity type or entity bundle, e.g. for a specific content type.

Meta tags can be further refined on a per-entity basis, e.g. for individual nodes, by adding the “Metatag” field to that entity type through its normal field settings pages.

<table>
<thead>
<tr>
<th>Type</th>
<th>Status</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>Active</td>
<td>Edit</td>
</tr>
<tr>
<td>Front page</td>
<td>Active</td>
<td>Edit</td>
</tr>
</tbody>
</table>
What is the Schema.org Blueprints module?
The Schema.org Blueprints module takes a Schema.org-first approach to building content models and structured data in Drupal.
<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core</strong></td>
<td>Provides base API/services, entities, and configuration for understanding Schema.org and mapping Drupal entities.</td>
</tr>
<tr>
<td><strong>Report</strong></td>
<td>Provides a report for browsing Schema.org types, properties, and naming conventions.</td>
</tr>
<tr>
<td><strong>UI</strong></td>
<td>Allows administrators to associate Schema.org types and properties to fieldable types.</td>
</tr>
</tbody>
</table>
The goal of the next demo to build out a website's base content types and information architecture in 5 minutes instead of a few hours.
Reports

Available updates

Recent log messages

About Schema.org

Configuration inspector

Field list

Metatag plugins
### SCHEMA.ORG BLUEPRINTS: OPTIONAL MODULES

<table>
<thead>
<tr>
<th><strong>Descriptions</strong></th>
<th><strong>JSON:API</strong></th>
<th><strong>RDF</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets entity type and field descriptions to their corresponding Schema.org type and property comments.</td>
<td>Applies Schema.org type and property names to Drupal core's JSON:API.</td>
<td>Integrates Schema.org mappings with Drupal core's RDF(a) mappings.</td>
</tr>
</tbody>
</table>
Content types

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Article</td>
<td>Use <em>articles</em> for time-sensitive content like news, press releases or blog posts.</td>
<td>Manage fields</td>
</tr>
<tr>
<td>Basic page</td>
<td>Use <em>basic pages</em> for your static content, such as an 'About us' page.</td>
<td>Manage fields</td>
</tr>
</tbody>
</table>
What are the benefits of the Schema.org Blueprints module?
SCHEMA.ORG BLUEPRINTS: BENEFITS

- Standardization
- Simplification
- Acceleration
Easy to understand - less thinking

Leveraging Schema.org, an open and established standard, makes it easier for organizations to create a content and information architecture that is simple for API consumers and search engines to understand.
Simpler to build - fewer resources
Having a standardized content, and information architecture removes the challenge of naming and relating things, making it easier to build rich content with fewer resources.
Faster to implement - quicker development

The Schema.org Blueprints module allows your organization's content and information architecture to be built in a few clicks.
Organizations can think less about their data structures and focus more on their content and user experience.
How do you configure and customize your Schema.org Blueprints?
SCHEMA.ORG MAPPING TYPES

Mapping types define what Schema.org types, properties, and data types are mapped to which Drupal entity types, fields and data types.

Mapping types configuration includes

- Recommended types
- Default types, properties, and subtyping
- Default base field mappings
- Default field weights and groups
GLOBAL SETTINGS

Types and Properties
- Field types, prefix, and unlimited

Names
- Prefixes, suffixes, and abbreviations
- Custom names and titles
- Acronyms and minor words

References
- Links to Schema.org and other resources
Default Schema.org type field types

DefinedTerm|string,list_string,field_ui:entity_reference:taxonomy_term
PostalAddress|address,entity_reference,text
URL|link,string
Time|time,string,datetime
DateTime|datetime

Enter one value per line, in the format
SchemaType|field_type_01,field_type_02,field_type_03.
Field types are applied in the order that they are entered.

Save configuration
PRO TIP: EDIT THE CONFIG/INSTALL YAML SOURCE

```yaml
recommended_schema_types:

  common:
    label: Common
    types:
      - Person
      - Place
      - Event
      - Organization

web_content:
  label: Web Content
  types:
    - WebContent
```
PRO TIP: USE DRUSH TO TEST SCHEMA.ORG MAPPINGS

# Import Schema.org configuration.
drush features:import -y schemadotorg

drush features:import -y schemadotorg_report

# Generate Schema.org types.
drush schemadotorg:create-type -y paragraph:ContactPoint paragraph:PostalAddress

drush schemadotorg:create-type -y media:AudioObject media:ImageObject media:VideoObject

drush schemadotorg:create-type -y node:Person node:Organization node:Place node:Event node:CreativeWork

# Generate content.
drush devel-generate:media --kill 50

drush devel-generate:content --kill --skip-fields=menu_link

# Delete Schema.org types.
drush schemadotorg:delete-type -y --delete-fields media:AudioObject media:ImageObject media:VideoObject

drush schemadotorg:delete-type -y --delete-entity paragraph:ContactPoint paragraph:PostalAddress

drush schemadotorg:delete-type -y --delete-entity node:Person node:Organization node:Place node:Event
MORE PRO TIPS

- If two Schema.org properties address similar use-cases, use the more common property. (Use 'image' instead of 'photo')
Next steps

What is next for the Schema.org Blueprints module?
SCHEMA.ORG BLUEPRINTS: NEXT STEPS

Try

Improve

Release
TRY SCHEMA.ORG BLUEPRINTS

- Download
  https://www.drupal.org/project/schemadotorg

- Composer
  composer require drupal/schemadotorg

- Simplytest.me
  https://simplytest.me/configure?project=schemadotorg
IMPROVE SCHEMA.ORG BLUEPRINTS

- Defining Schema.org type and properties
- Determining the proper user experience
- Building sector specific content models
- Integrating with JSON-LD
- Stabilizing codebase with test coverage
SCHEMA.ORG BLUEPRINTS ROADMAP

Alpha
- Define baseline Schema.org types and properties
- Establish sub-modules and feature list

Beta
- Find co-maintainers and sponsoring organizations
- Improve documentation with in-line help
- Determine upgrade path between Schema.org versions

Stable
- Add Schema.org types and properties as needed
- Provide additional enhancements via custom code or contrib modules
What are the pros and cons of the Schema.org Blueprints module?
PROS

- Structured data
- Standard naming convention
- Well-defined relationships
- Ability to extend and grow with Schema.org
CONS

- Overly complex structured data
- Limited naming convention
- Too many relationships
- Tied to the growth and decisions of Schema.org
Thanks!

GET INVOLVED!

You can find me at:
@jrockowitz

Learn more:
https://www.drupal.org/project/schemadotorg