

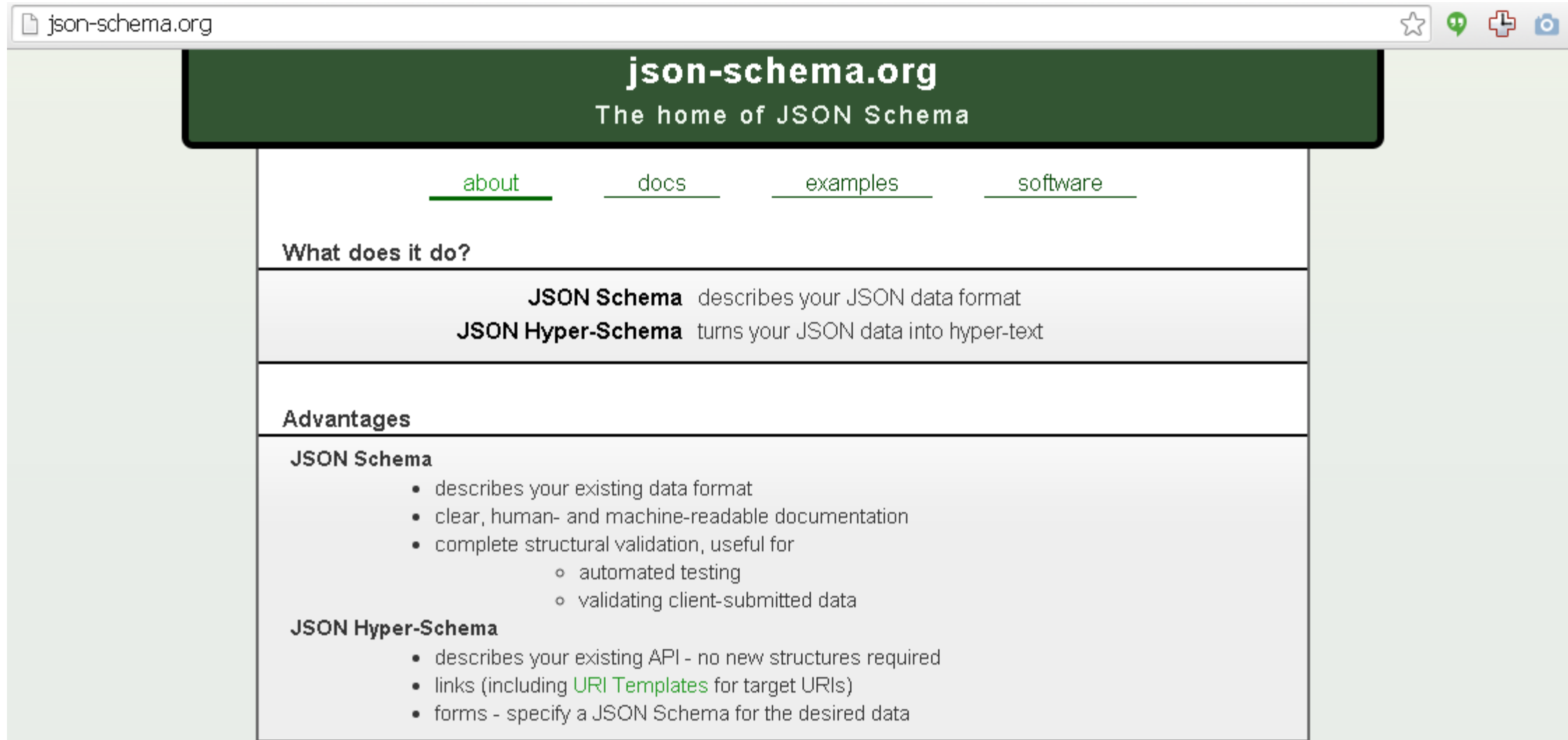
Json.NET Schema



Xavier Morera

@xmorera | www.xaviermorera.com

JSON Schema



The screenshot shows the homepage of json-schema.org. The browser's address bar displays 'json-schema.org'. The website has a dark green header with the title 'json-schema.org' and the tagline 'The home of JSON Schema'. Below the header is a navigation bar with four links: 'about', 'docs', 'examples', and 'software', each underlined. The main content area is divided into sections. The first section is titled 'What does it do?' and contains two items: 'JSON Schema describes your JSON data format' and 'JSON Hyper-Schema turns your JSON data into hyper-text'. The second section is titled 'Advantages' and contains two sub-sections. The first sub-section is 'JSON Schema' with a bulleted list of advantages: 'describes your existing data format', 'clear, human- and machine-readable documentation', and 'complete structural validation, useful for' followed by two sub-bullets: 'automated testing' and 'validating client-submitted data'. The second sub-section is 'JSON Hyper-Schema' with a bulleted list: 'describes your existing API - no new structures required', 'links (including URI Templates for target URIs)', and 'forms - specify a JSON Schema for the desired data'.

json-schema.org
The home of JSON Schema

[about](#) [docs](#) [examples](#) [software](#)

What does it do?

JSON Schema describes your JSON data format
JSON Hyper-Schema turns your JSON data into hyper-text

Advantages

JSON Schema

- describes your existing data format
- clear, human- and machine-readable documentation
- complete structural validation, useful for
 - automated testing
 - validating client-submitted data

JSON Hyper-Schema

- describes your existing API - no new structures required
- links (including [URI Templates](#) for target URIs)
- forms - specify a JSON Schema for the desired data

JSON Schema

- Specification for defining JSON structure

Specification

The latest draft is v4. The specification is split into three parts:

JSON Schema Core defines the basic foundation of JSON Schema
JSON Schema Validation defines the validation keywords of JSON Schema
JSON Hyper-Schema defines the hyper-media keywords of JSON Schema

They are also available as Internet Drafts on the IETF site: [core](#), [validation](#) and [hyper-schema](#).

Json.NET Schema

- Json.NET's functionality for describing and validating JSON
- Create schemas
- Validate JSON
- Separate package
- Licensed

Licensing & Pricing

Perpetual, royalty-free licenses are available for commercial use.

Edition	Price
AGPL 3.0	Free with limitations (1000 validations per hour)
1 Developer Indie *	\$99 USD
1 Developer Business	\$149 USD
Site License	\$999 USD

* Companies or incorporated entities with **more than 10 employees** must purchase a Business plan.

Json.NET Schema

- Available in NuGet



The screenshot displays the NuGet package manager interface. At the top, there are filters for 'Stable Only' and 'Sort by: Relevance'. The search bar contains the text 'json schema'. Below the search bar, two packages are listed:

- Json.NET Schema**: A complete and easy-to-use JSON Schema framework for .NET. An 'Install' button is visible next to this package.
- Json.NET**: A popular high-performance JSON framework for .NET. A green checkmark icon is visible next to this package.

On the right side of the interface, the details for the selected package, 'Json.NET Schema', are displayed:

- Created by:** Newtonsoft
- Id:** [Newtonsoft.Json.Schema](#)
- Version:** 1.0.10
- Last Published:** 8/16/2015
- Downloads:** 6278
- License:** [View License](#)
- [Project Information](#)
- [Report Abuse](#)
- Description:** Json.NET Schema is a complete and easy-to-use JSON Schema framework for .NET
- Tags:** json schema
- Dependencies:** Newtonsoft.Json (≥ 6.0.8)

A note at the bottom states: *Each item above may have sub-dependencies subject to additional license agreements.*

Creating Schemas

- Describe your JSON
- JSchema
- Create manually

```
JSchema schema = new JSchema
{
    Type = JSchemaType.Object,
    Properties = {
        { "name", new JSchema { Type = JSchemaType.String } },
        { "courses", new JSchema{
            Type = JSchemaType.Array,
            Items = { new JSchema { Type = JSchemaType.String } }
        }
    },
    { "since", new JSchema { Type = JSchemaType.String } },
    { "happy", new JSchema { Type = JSchemaType.Boolean } },
    { "issues", new JSchema { Type = JSchemaType.Object } },
};
```

- Generate from .NET objects

```
JSchemaGenerator jsonSchemaGenerator = new JSchemaGenerator();
JSchema schema = jsonSchemaGenerator.Generate(typeof(Author));
```

Validate JSON

- Validating a JSON object is easy with Json.NET Schema
- Extension method for Json.NET LINQ objects
- JObject.IsValid()
 - Overload with error message list

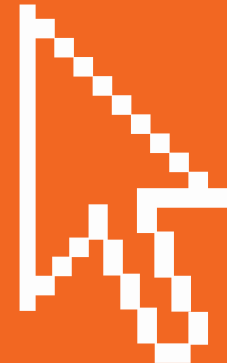
```
IList<string> why;  
bool validateManual = author.IsValid(manualSchema, out why);
```

- JSchemaValidatingReader/JSchemaValidatingWriter
 - Validating while reading/writing

Demo 27: Json.NET Schema

Create Schema

Validate JSON



Takeaway

JSON Schema

Json.NET Schema

Create
Generate

Validate