

JSON Schema Design Mode

Octavian Nadolu, Syncro Soft
octavian.nadolu@oxygenxml.com
@OctavianNadolu

© 2021 Syncro Soft SRL. All rights reserved.



Agenda

- JSON Schema Support in Oxygen
- Create JSON Schema from Scratch
- Visualize and Edit Complex JSON Schemas
- Generate JSON Schema Documentation



JSON Schema

JSON Schema is a vocabulary that allows you to **annotate** and **validate** JSON documents



<http://json-schema.org>



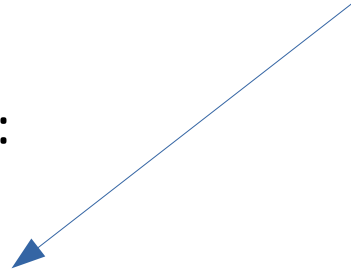
JSON Schema Definition

- It is recommended to have the schema definition on the first level

`"$schema": "http://json-schema.org/draft-07/schema#"`

- JSON Schema used versions:

- draft-04
- draft-06
- draft-07
- draft/2020-12



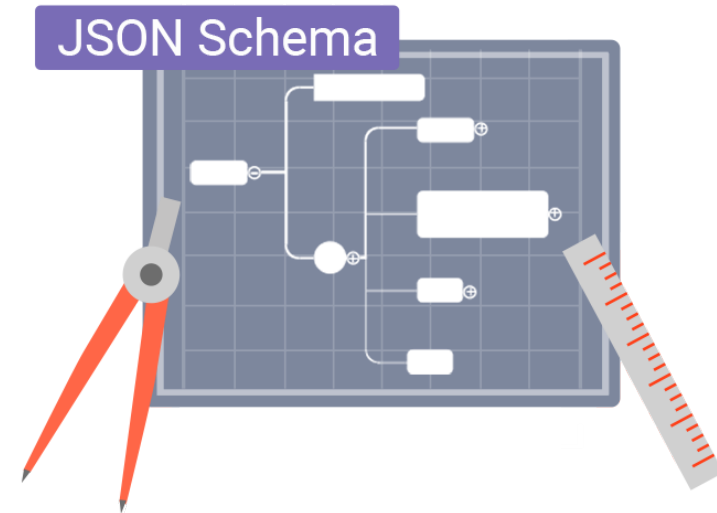
Question: Do you use JSON Schema, what versions?

- No
- Yes. Draft 4, 6 or 7
- Yes. (2020-12)
- Yes. Other (use the Questions pane to provide more details)



JSON Schema Support in Oxygen

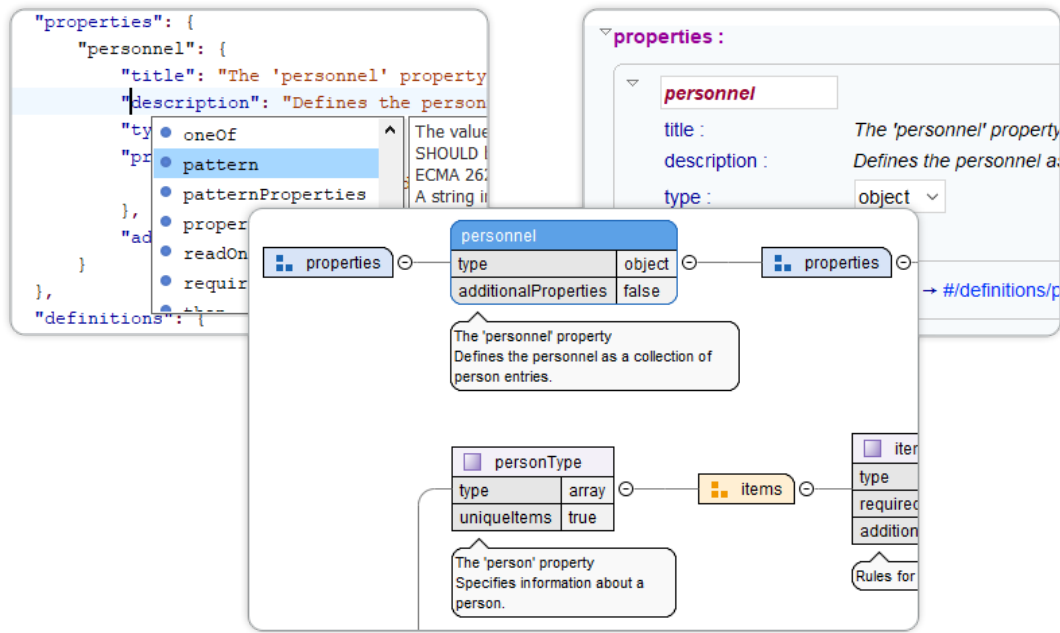
- **JSON Schema Editor** - specialized editor with various editing features
- **Validation** against JSON Schema
- **Editing** based on JSON Schema
- **Tools**
 - Generate **JSON Schema Documentation**
 - **Generate Sample JSON** Files from a JSON Schema
 - **Generate JSON Schema** from a JSON File
 - **XSD to JSON** Schema Converter



JSON Schema Editor

Design, develop, and edit JSON Schemas in:

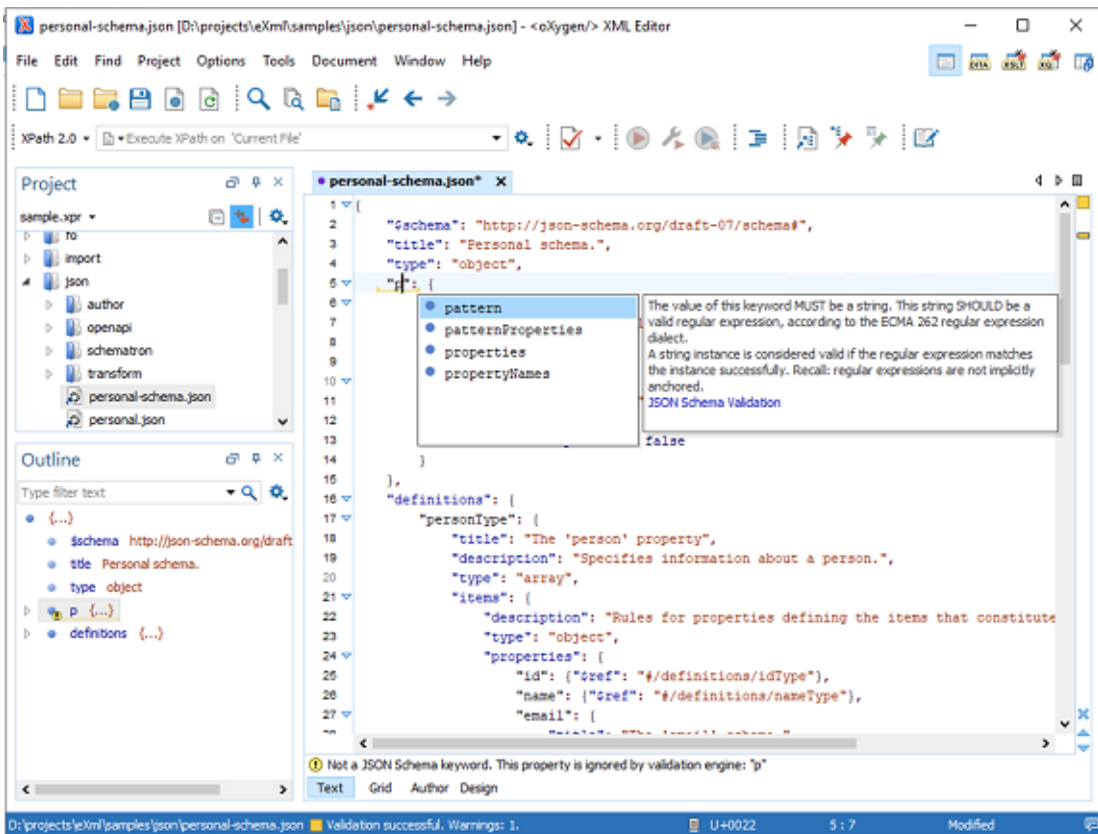
- Text Editing Mode
- Author Editing Mode
- Schema Design Mode



Text Editing Mode

Text editing mode is packed full of editing helpers

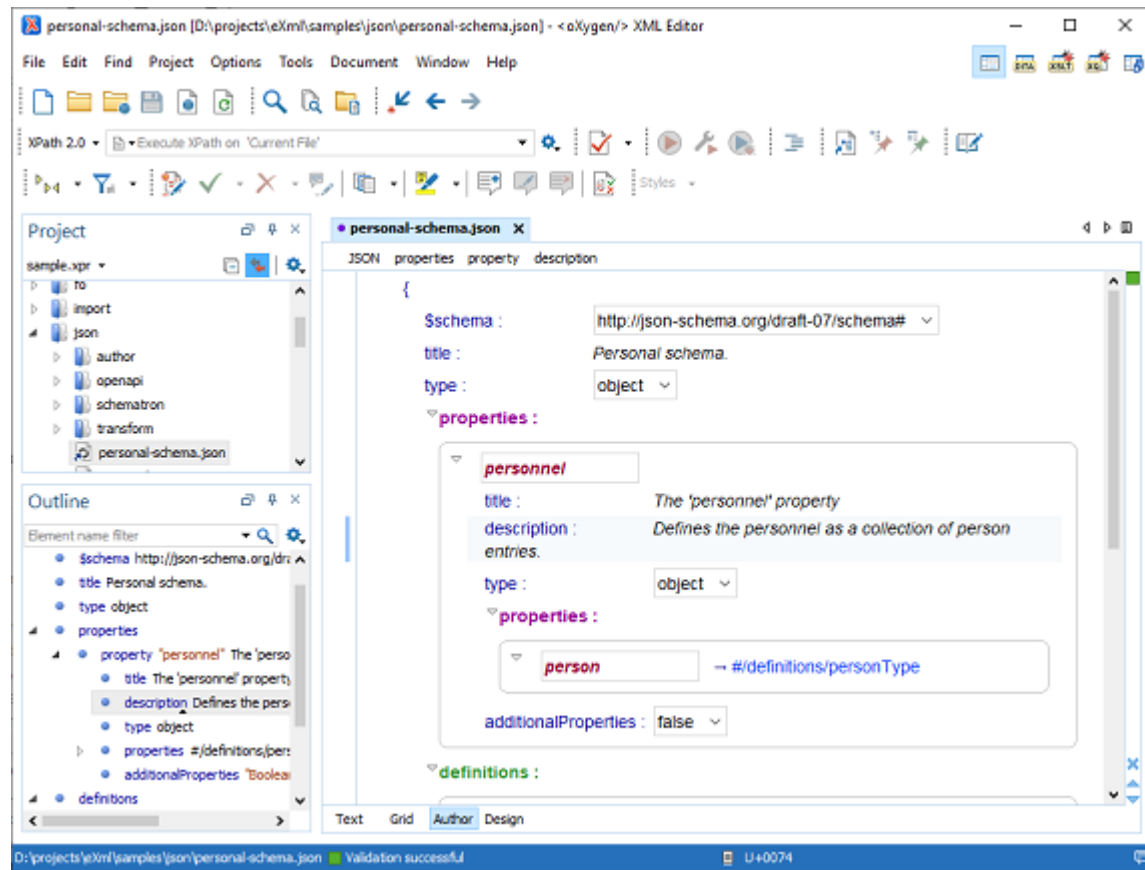
- JSON **Outline View**
- JSON-specific **Syntax Highlighting**
- Search and **Find/Replace**
- **Drag and Drop**
- **Validation**
- **Format and Indent (Pretty Print)**



Author Editing Mode

Visual editing mode for JSON Schema documents:

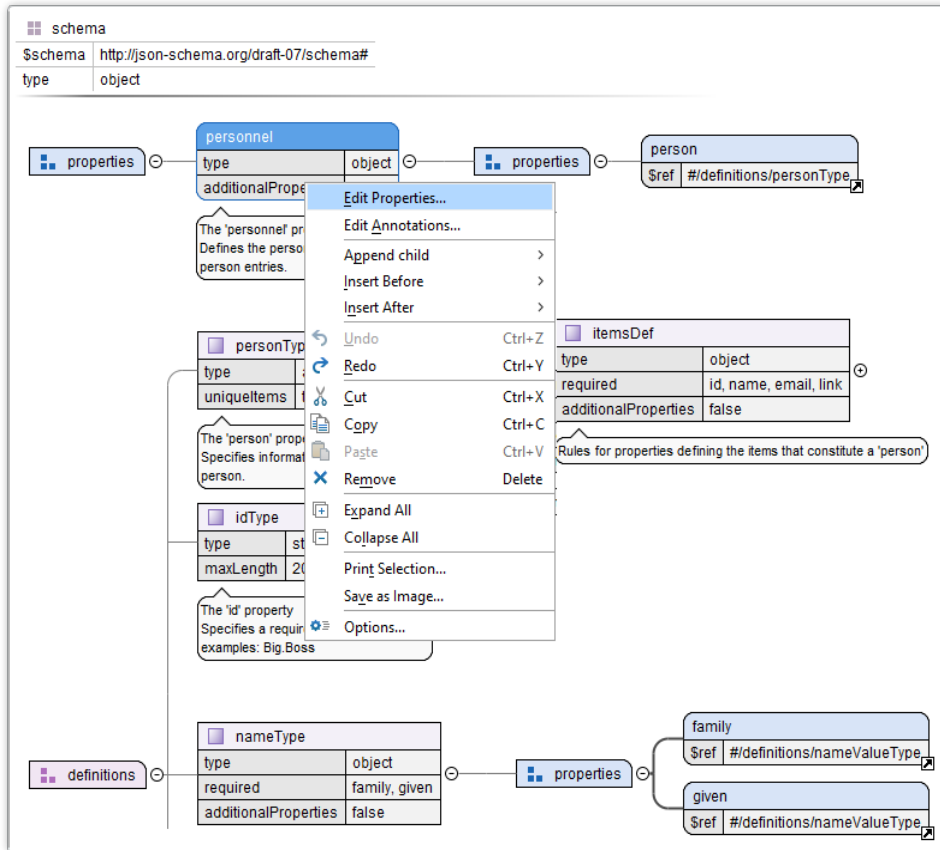
- JSON Schema **framework**
- **Content completion** support
- **Validation**
- Specific **CSS** for rendering
- Create your own **custom JSON framework**



Schema Design Mode

Visualize, edit, and understand JSON Schemas

- In-Place Component Editing
- Edit Properties
- Edit Annotations
- Schema Editing Actions
- Move Component Up/Down Actions
- Print/Save as Image



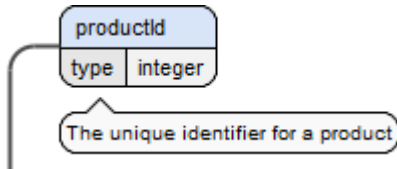
Create JSON Schema from Scratch

- JSON-based product catalog
 - productID – the product identifier
 - productName – the product name
 - price – the selling price of the product
 - tags – an optional set of tags

```
{  
  "productId": 1,  
  "productName": "A green door",  
  "price": 12.50,  
  "tags": ["home", "green"]  
}
```

Create JSON Schema from Scratch

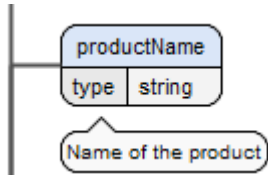
- productID – The identifier for a product
 - Is a numeric value
 - Is required



```
{  
  "productId": 1,  
  "productName": "A green door",  
  "price": 12.50,  
  "tags": ["home", "green"]  
}
```

Create JSON Schema from Scratch

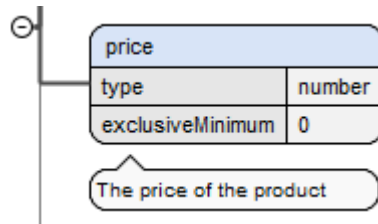
- productName – The name of the product
 - Is a string value
 - Is required



```
{  
  "productId": 1,  
  "productName": "A green door",  
  "price": 12.50,  
  "tags": ["home", "green"]  
}
```

Create JSON Schema from Scratch

- price – The price of the product
 - Is a number value
 - Is required
 - Must be greater than 0

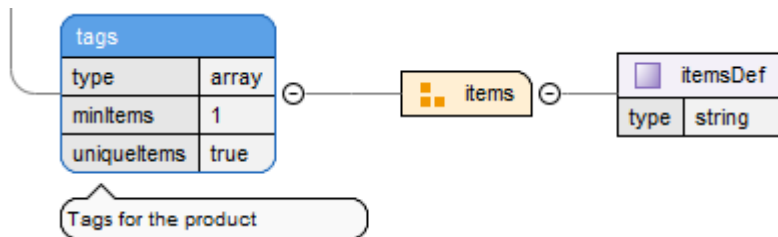


```
{  
  "productId": 1,  
  "productName": "A green door",  
  "price": 12.50,  
  "tags": ["home", "green"]  
}
```

Create JSON Schema from Scratch

- tags – Tags for the product
 - Is an array of string values
 - Is not required
 - At least one item in the array
 - Must be unique relative to one another

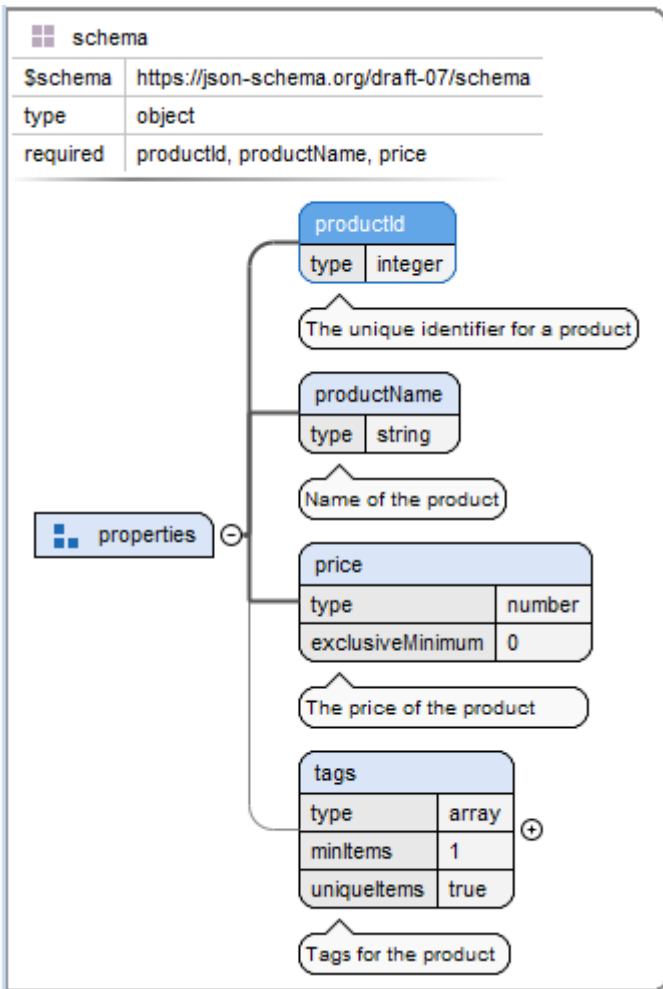
```
{  
  "productId": 1,  
  "productName": "A green door",  
  "price": 12.50,  
  "tags": ["home", "green"]  
}
```



JSON Schema for Product Catalog

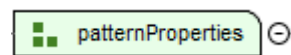
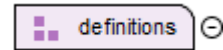
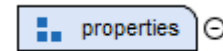
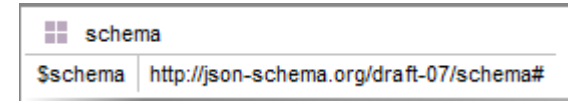
Easy to create a schema from scratch

- Use **document template**
- **Edit** the components **in-place**
- Edit the component **properties** in the in-place **view**
- Add new components using the **append/insert** contextual **actions**
- Change the documentation using the **Annotation dialog** box



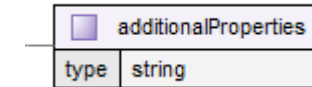
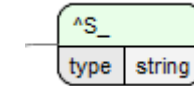
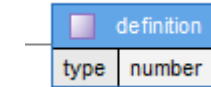
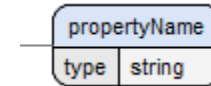
Schema Diagram Components

- **schema** – defines the root schema component
- **properties** – defines a group of *property* components
- **definitions** – contains a group of *definition* components
- **pattern properties** – contains a group of *pattern property* components



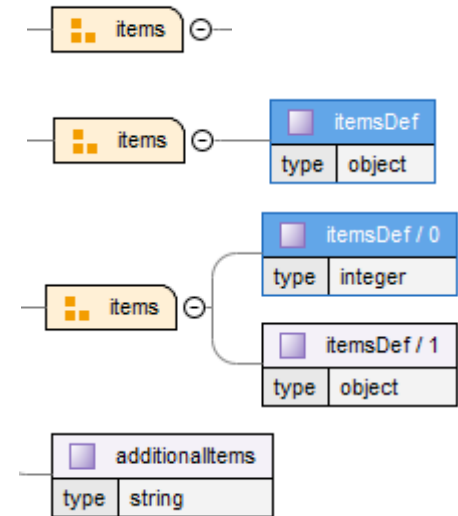
Schema Diagram Components

- **property** – defines a property declaration
- **definition** – contains a declaration of a reusable definition
- **pattern property** – defines a pattern property
- **additional properties** – contains a definition for the additional properties



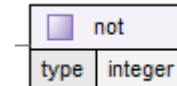
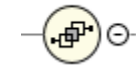
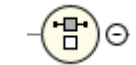
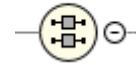
Schema Diagram Components- Arrays

- **items** – contains the array definition or definitions
- **Items definition** – definition for all array items
- **Items array** – an array of definitions, one for each item from the array
- **additional items** – contains a definition for the additional items from an array



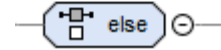
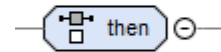
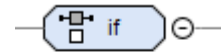
Schema Diagram Components- Composition

- **allOf** – a list of definitions, data must be valid against *all* definitions
- **oneOf** – a list of definitions, data must be valid against exactly *one* of the definitions
- **anyOf** – a list of definitions, data must be valid against *any* definition
- **not** – a definition, data must not be valid against the given definition



Schema Diagram Components- Conditional

- **if** – contains a schema definition for the *if* condition
- **then** – contains a schema definition, data must be valid against it when the *if* condition is *true*
- **else** – contains a schema definition, data must be valid against it when the *if* condition is *false*

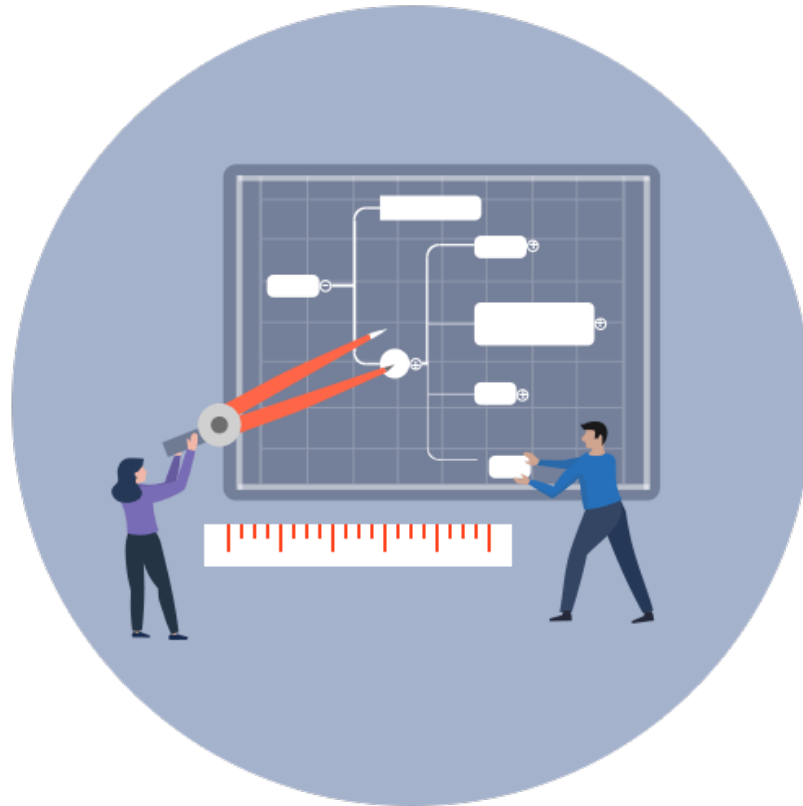


Question: How do you use/intend to use JSON Schema?

- To define an API
- In a database
- To validate JSON documents
- Other (use the Questions pane to provide more details)



Visualize and Edit Complex JSON Schemas

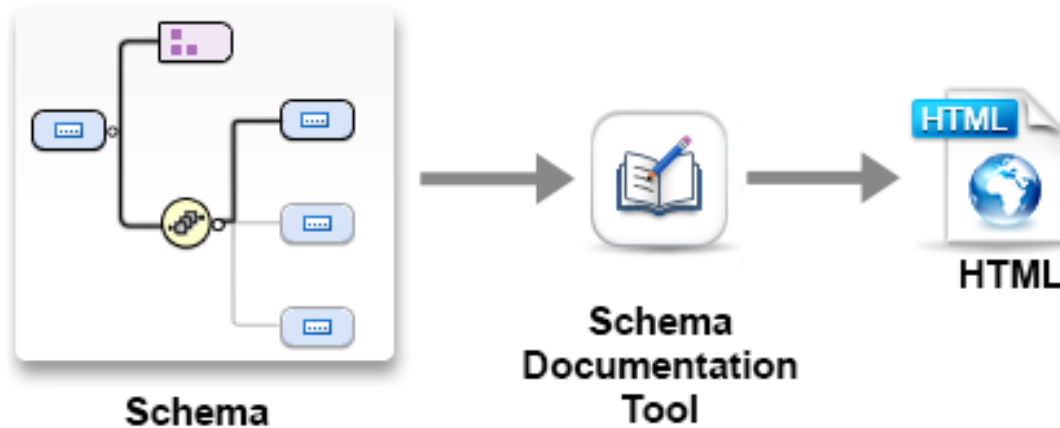


Visualize and Edit Complex JSON Schemas

- Smart navigation
- Zoom in/out
- Expand/Collapse components
- Go to references
- Go back and forward between components viewed or edited
- Validation markers

Generate JSON Schema Documentation

- Tool for generating detailed documentation for a JSON Schema file in HTML format



JSON Schema Documentation

- Generate documentation in one file or split into multiple files
- Option to include components details
- Display the diagram image for each component

The screenshot shows a web browser window displaying the documentation for a JSON Schema component named 'Order'. The browser address bar shows the file path: `D:/Projects/VideoDemonstrations/JSON_Schema_Design_v24/JSON_Schema_Design/Samples/json-schem...`

The main content area is titled 'Property Order' and contains several sections:

- Annotations:** Description: A document used to order goods and services.
- Diagram:** A visual representation of the schema structure. It shows a box for 'Order' with properties: `maxItems: 1`, `minItems: 1`, and `type: array`. An arrow points from 'Order' to a box labeled 'items', which in turn points to a box labeled 'ItemsDef' with the property `$ref: #/definitions/Order`. A tooltip for 'Order' contains the text: 'A document used to order goods and services.'
- Type:** array
- Constraints:** Unique Items: false
- Array Items:** Items
UBL-Order-2.1.json.html#/properties/Order/items
- Additional Items:** true
- Used by:** Schema
#/schema
- Source:**

```

"Order": {
  "type": "array",
  "minItems": 1,
  "maxItems": 1,
  "description": "A document used to order goods and services.",
  "items": {"$ref": "#/definitions/Order"}
}
                    
```

On the right side, there is a 'Showing:' panel with the following options checked:

- Annotations
- Diagram
- Properties
- Constraints
- Used By
- Source

A 'Close' button is located at the bottom of this panel.

Conclusion

- Complete support for JSON Schemas
- JSON Schema Editor
- Validate JSON with JSON Schema
- Editing based on JSON Schema
- Useful JSON Schema Tools

Future Plans

- Improve JSON Schema **Diagram**
- Better **JSON Schema Documentation**
- **Quick fixes** for JSON problems
- **OpenAPI** support
- **YAML** support



Question: What features are the most important for you?

- ❑ JSON Schema Diagram
- ❑ JSON Schema Documentation
- ❑ Quick fixes for JSON problems
- ❑ OpenAPI support
- ❑ YAML support



Resources

- <https://json-schema.org/>
- oxygenxml.com/json_schema_editor.html
- oxygenxml.com/doc/ug-editor/topics/editing-JSON-schema.html

Video Demos:

- oxygenxml.com/demo/json_author.html
- oxygenxml.com/demo/json_tools.html
- oxygenxml.com/demo/json_validation.html
- oxygenxml.com/demo/json_editing.html
- oxygenxml.com/demo/json_query.html



Questions?

Octavian Nadolu
Product Manager at Syncro Soft

octavian.nadolu@oxygenxml.com

Twitter: [@OctavianNadolu](https://twitter.com/OctavianNadolu)

LinkedIn: [octaviannadolu](https://www.linkedin.com/in/octaviannadolu)