Create and Design JSON Schema

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

syncro soft

<oXygen/> XML Editor

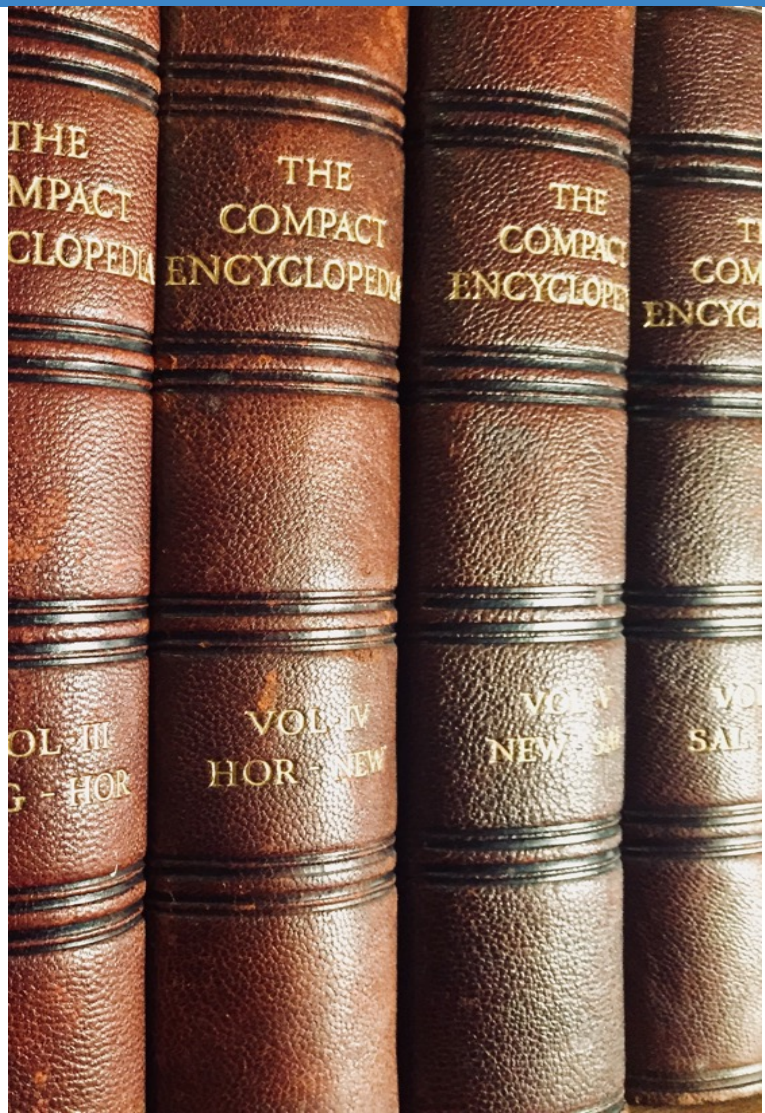
Agenda

- JSON Schema Specification
- Create JSON Schema From Scratch
- Add New Components Using Pallet View
- Design Schema Using Drag-and-Drop Support
- Refactoring Actions
- Visualize and Edit Complex JSON Schemas

JSON Schema

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

http://json-schema.org

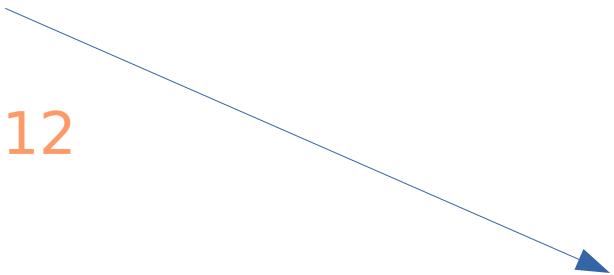


JSON Schema Benefits

- Describe your data format
- Provide human and machine readable documentation
- Validate data
- Automated testing

JSON Schema Definition

- JSON Schema used versions:
  - draft-04
  - draft-06
  - draft-07
  - Draft/2020-12



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

It is recommended to have the schema definition on the first level in the JSON document

Question: Do you use JSON Schema, what versions?

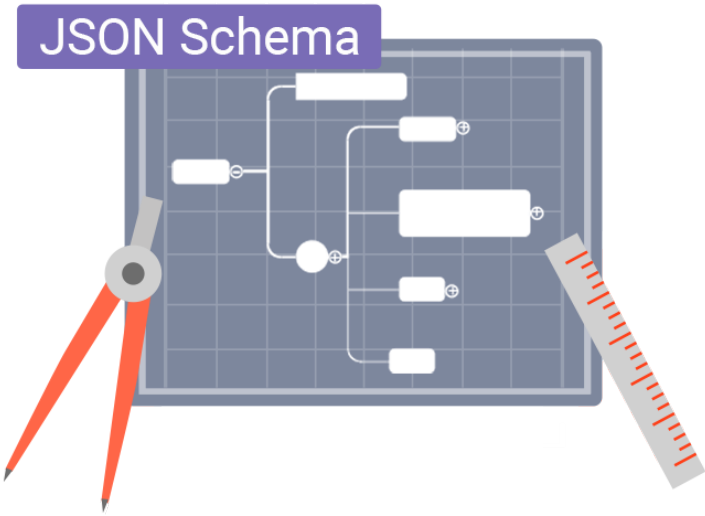
- No
- Yes. Draft 4, 6 or 7
- Yes. (2020-12)
- Yes. Other (use the Questions pane to provide more details)

JSON Schema Usage

- Describe the structure and validation constraints of JSON documents
- JSON Schema is used in OpenAPI spcification
- JSON Schema partially used is AsyncAPI specification
- JSON Schema is used in databases

JSON Schema Support in Oxygen

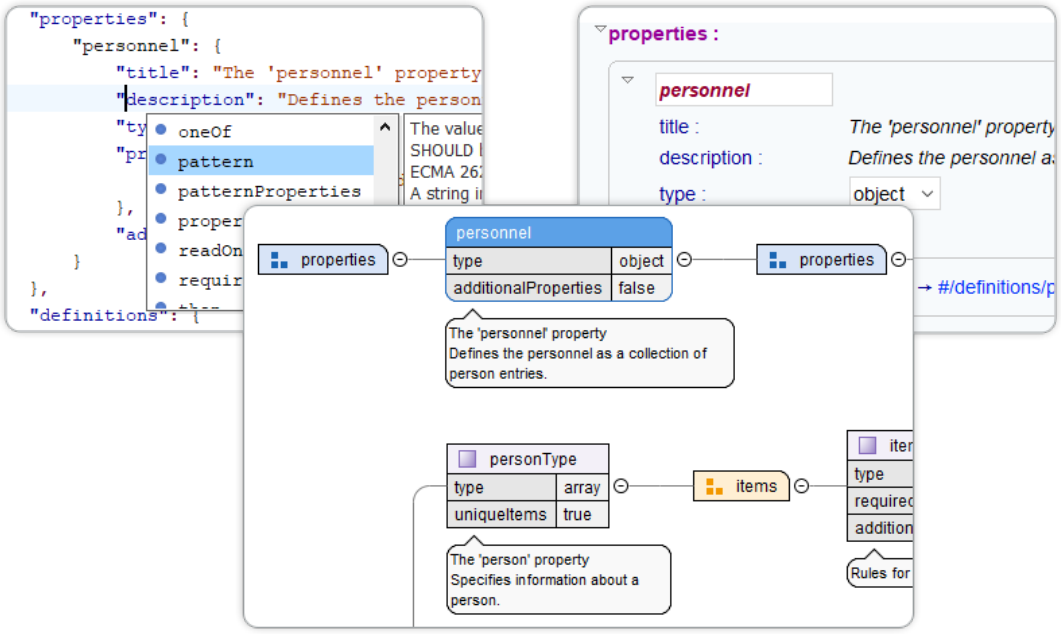
- JSON Schema Editor - specialized editor with various editing features
- Validation against JSON Schema
- Editing based on JSON Schema
- Tools
  - Generate JSON Schema Documentation
  - Generate Sample JSON Files from a JSON Schema
  - Generate JSON Schema from a JSON File
  - XSD to JSON Schema Converter



JSON Schema Editor

Design, develop, and edit JSON Schemas in:

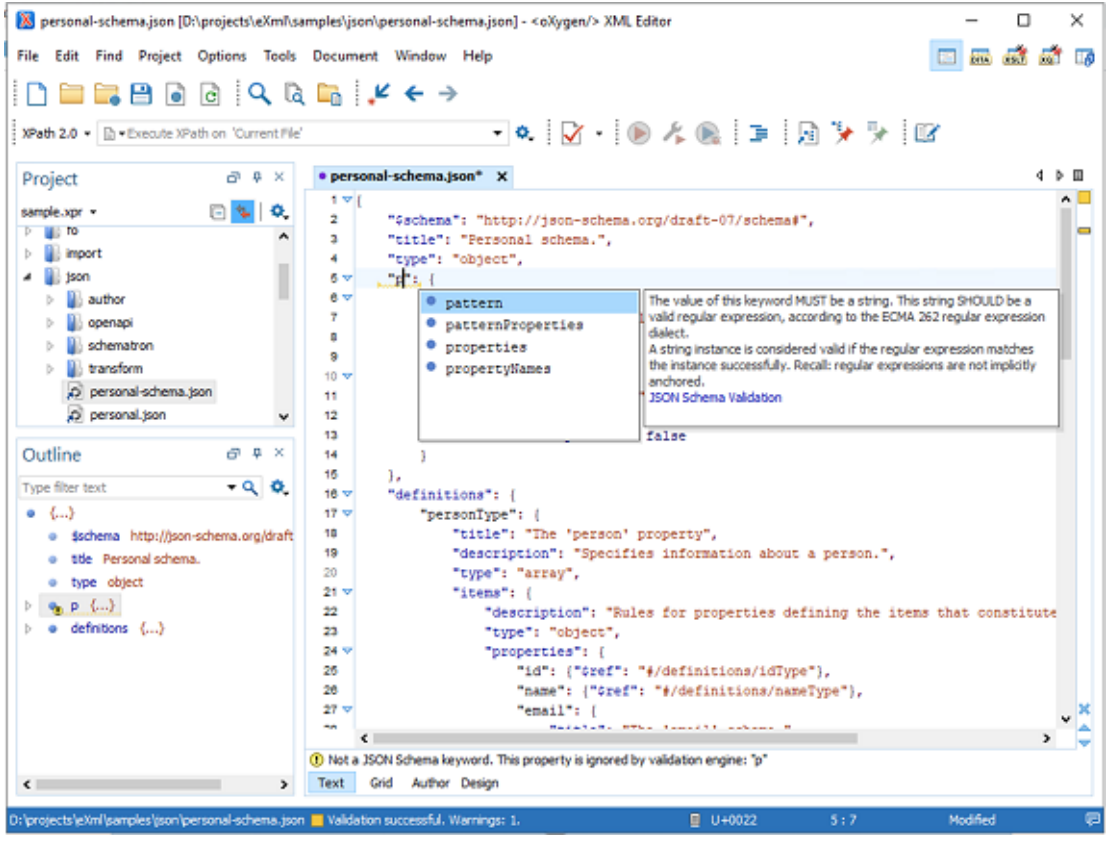
- Text Editing Mode
- Author Editing Mode
- Schema Design Mode



Text Editing Mode

Text editing mode is packed full of editing helpers

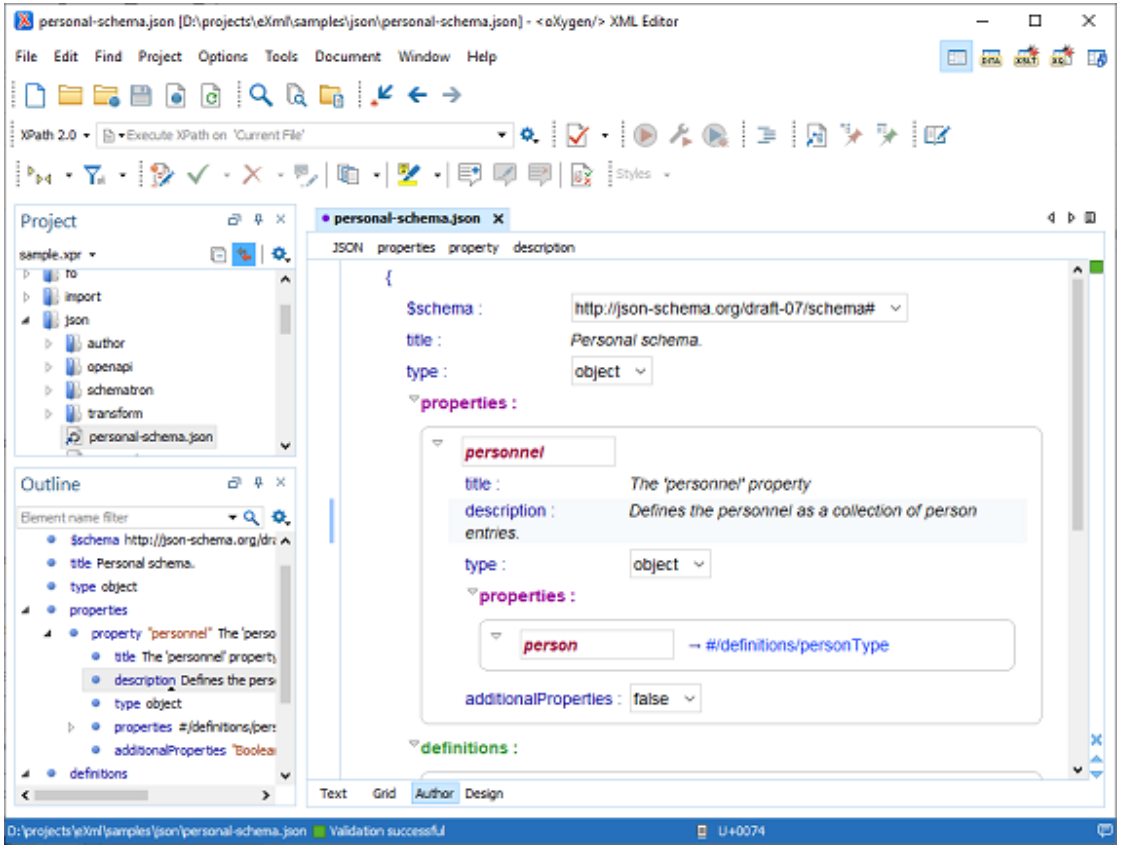
- JSON Outline View
- JSON-specific Syntax Highlighting
- Search and Find/Replace
- Drag and Drop
- Validation
- Format and Indent (Pretty Print)



Author Editing Mode

Visual editing mode for JSON Schema documents:

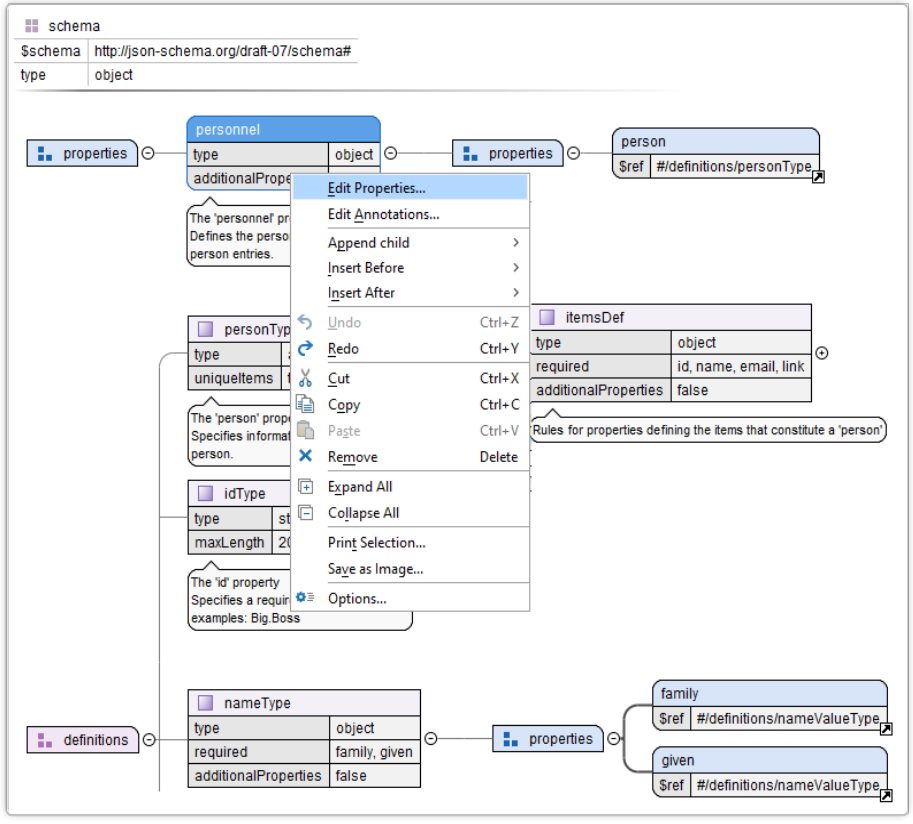
- JSON Schema framework
- Content completion support
- Validation
- Specific CSS for rendering
- Create your own custom JSON framework



Schema Design Mode

Visualize, edit, and understand JSON Schemas

- In-Place Component Editing
- Drag and drop
- Palette view to Create New Components
- Schema Editing Actions
- Print/Save as Image



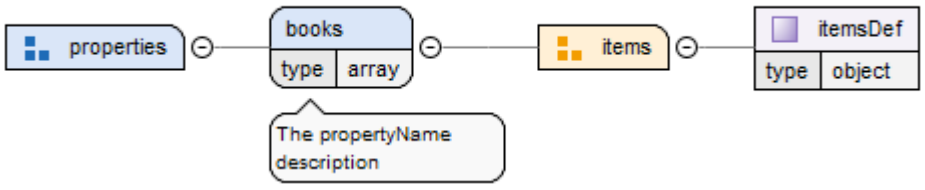
Create JSON Schema from Scratch

- An array of JSON objects that define a library of books
  - title – the book title
  - genre – the literary genre
  - authors – the authors of the book
    - name – the author name
    - short_bio – the biography of the author

```
{
  "books": [{
     "title": "Quantum Mechanics",
     "genre": "Science",
     "authors": [
       {
         "name": "Leonard Susskind",
         "short_bio": ""
       }]
    }
  ]
}
```

Create JSON Schema from Scratch

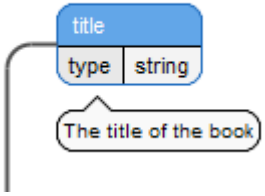
- books – An array of book objects




```
{
  "books": [{
      "title": "Quantum Mechanics",
      "genre": "Science",
      "authors": [
        {
          "name": "Leonard Susskind",
          "short_bio": ""
        }]
    }
  ]
}
```

Create JSON Schema from Scratch

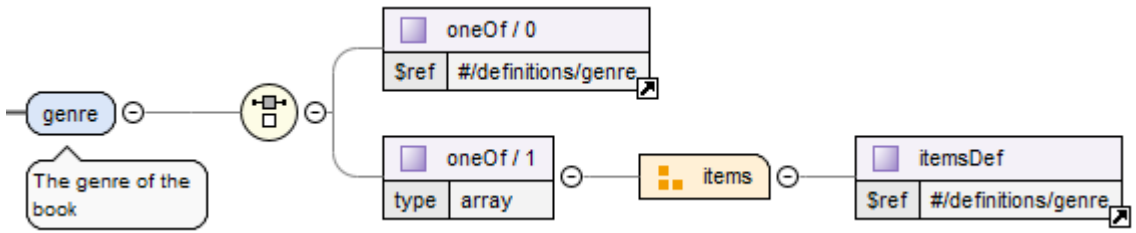
- title – The title of the book
  - Has a string value
  - Is required

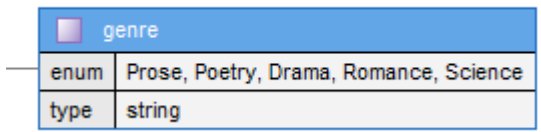



```
{
  "books": [{
      "title": "Quantum Mechanics",
      "genre": "Science",
      "authors": [
        {
          "name": "Leonard Susskind",
          "short_bio": ""
        }]
    }
  ]
}
```

Create JSON Schema from Scratch

- genre – The genre of the book
  - Has a string or an array of strings
  - Is required
  - The values are restricted: Prose, Poetry, Drama, Romance, Science

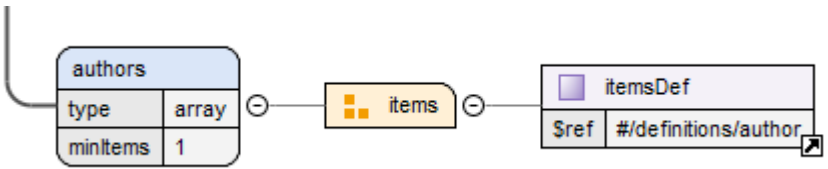





```
{
  "books": [{
     "title": "Quantum Mechanics",
     "genre": "Science",
     "authors": [
       {
         "name": "Leonard Susskind",
         "short_bio": ""
       }]
    }
  ]
}
```

Create JSON Schema from Scratch

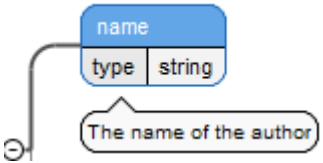
- authors – Authors of the book
  - Has an array of author objects
  - Is required
  - At least one item in the array




```
{
  "books": [{
     "title": "Quantum Mechanics",
     "genre": "Science",
     "authors": [
       {
         "name": "Leonard Susskind",
         "short_bio": ""
       }]
    }
  ]
}
```

Create JSON Schema from Scratch

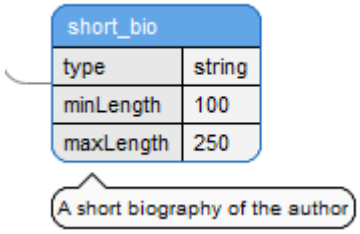
- name – The name of the author
  - Has a string value
  - Is required




```
{
  "books": [{
      "title": "Quantum Mechanics",
      "genre": "Science",
      "authors": [
        {
          "name": "Leonard Susskind",
          "short_bio": ""
        }]
    }
  ]
}
```

Create JSON Schema from Scratch

- short_bio – A short biography of the author
  - Has a string value
  - Is not required
  - The length of the string must be between 100 and 250 characters

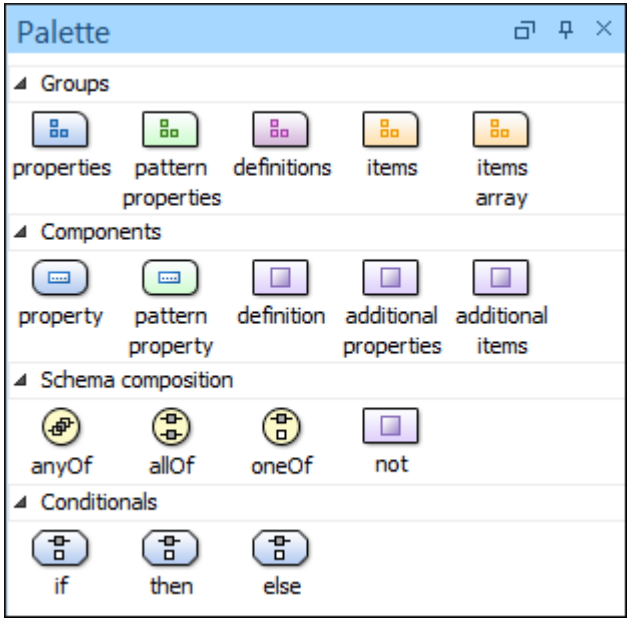



```
{
  "books": [{
      "title": "Quantum Mechanics",
      "genre": "Science",
      "authors": [
        {
          "name": "Leonard Susskind",
          "short_bio": ""
        }]
    }
  ]
}
```

Create JSON Schema from Scratch

Easy to create a schema from scratch

- Create JSON Schema using document template
- Use drag and drop support
- Add new components using the Pallet view
- Edit the components in-place
- Edit the component properties in the in-place view
- Use refactor action

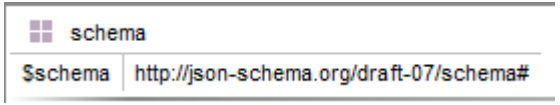


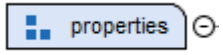
Question: How do you use/intend to use JSON Schema?

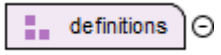
- To validate JSON documents
- To define an API
- In a database
- Other (use the Questions pane to provide more details)

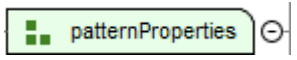
Schema Diagram Components

- 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



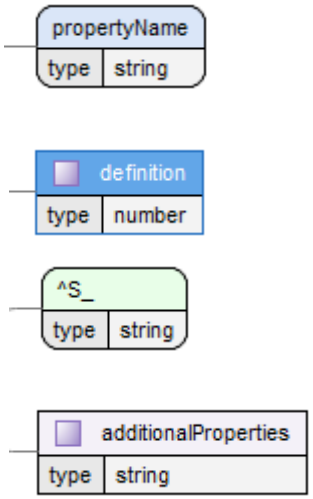






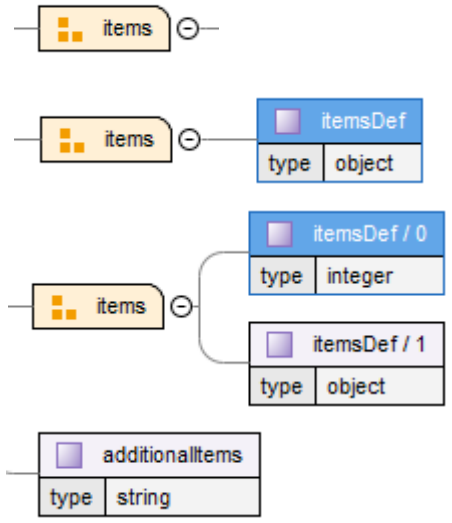
Schema Diagram Components

- 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



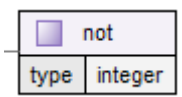
Schema Diagram Components- Arrays

- 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



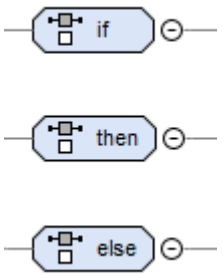
Schema Diagram Components- Composition

- 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

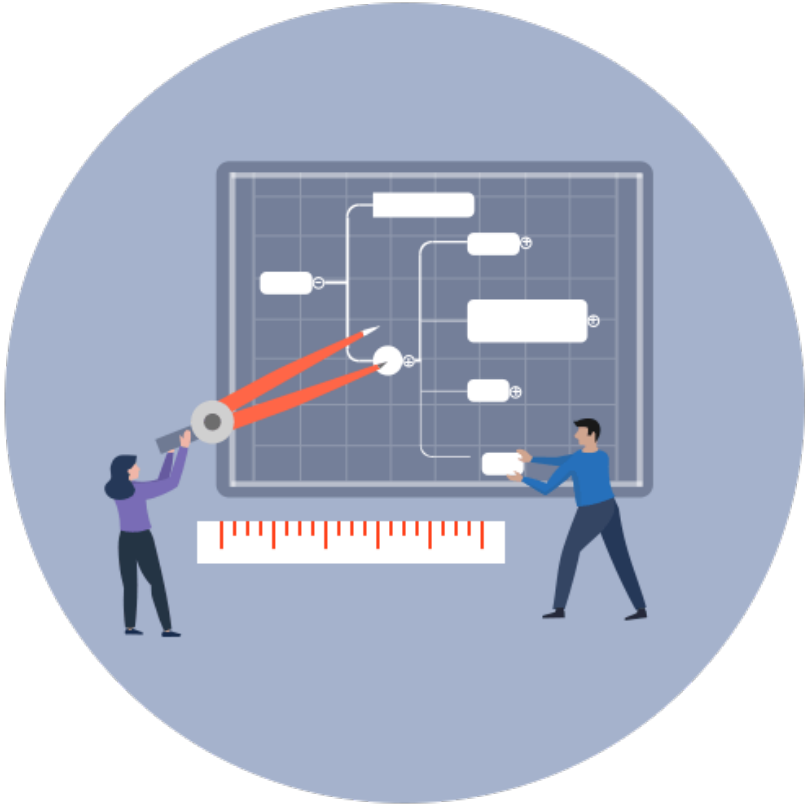


Schema Diagram Components- Conditional

- 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*



Visualize and Edit Complex JSON Schemas

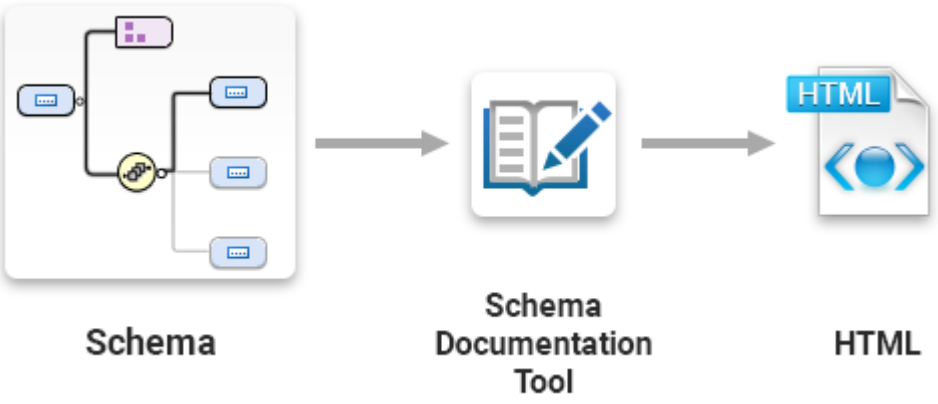


Visualize and Edit Complex JSON Schemas

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

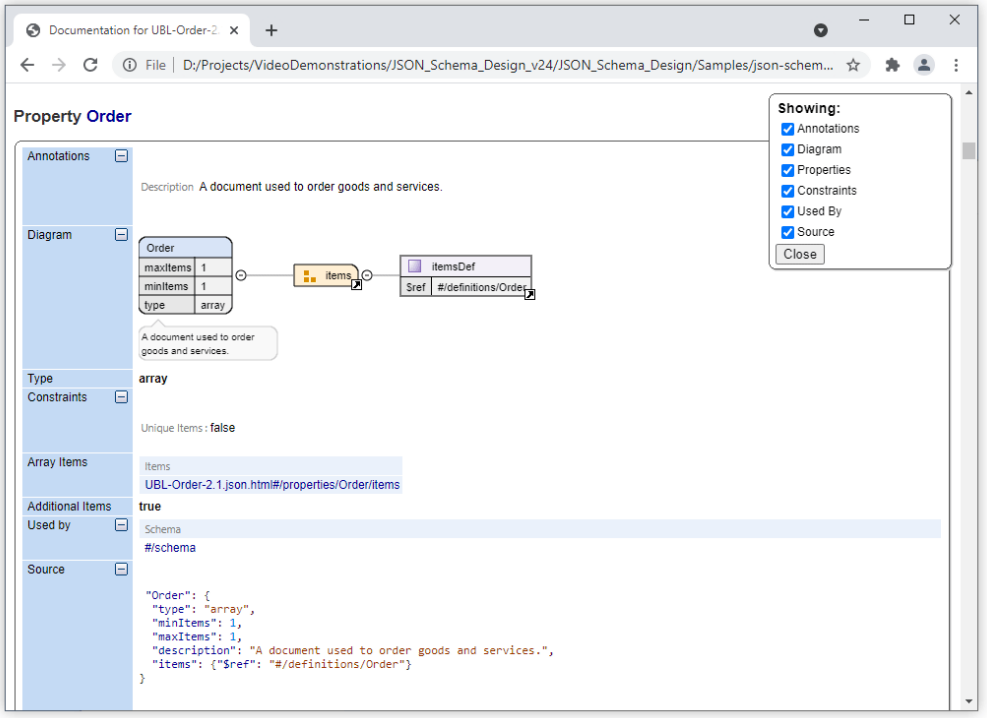
Generate JSON Schema Documentation

- 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



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
- Support JSON Schema 2020-12
- Improve OpenAPI support
- Quick fixes for JSON problems
- Improve YAML support

Question: What features are the most important for you?

- JSON Schema Diagram
- Support JSON Schema 2020-12
- OpenAPI support
- Quick fixes for JSON problems
- YAML support

Resources

- oxygenxml.com/json_schema_editor.html
- oxygenxml.com/doc/ug-editor/topics/editing-JSON-schema.html
- https://json-schema.org/
- https://www.openapis.org
- https://www.asyncapi.com

Video Demos

- oxygenxml.com/demo/json_schema_palette.html
- oxygenxml.com/demo/introducing_the_json_schema_design.html
- 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

- Webinar: OpenAPI Editing, Testing, and Documenting (Wed, April 6, 2022)

Questions?

Octavian Nadolu
Product Manager at Syncro Soft

octavian.nadolu@oxygenxml.com
Twitter: @OctavianNadolu
LinkedIn: octaviannadolu