Oxygen JSON Editor

JSON Schema
POWER SHELL
LESS
SHELL
YAML
PHP
JSON
JS DOCKERFILE
CSS JSONL

Mircea Enachescu, Syncro Soft
mircea@oxygenxml.com

Oxygen JSON Editor

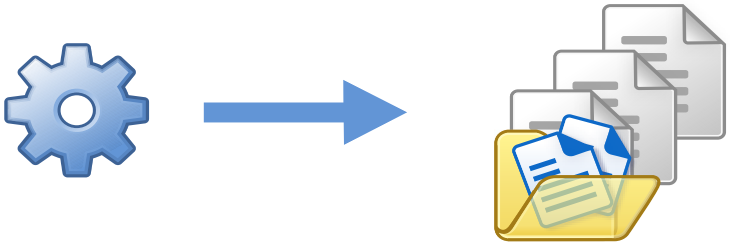
- A specialized tool designed for editing JSON documents

Oxygen JSON Editor- Features

- **JSON** – editing, validation, conversion
- **JSON Schema** – editing, validation, design, refactoring, documentation
- **YAML** - editing, validation, conversion
- **OpenAPI/AsyncAPI** – editing, validation, documentation, testing
- **HTML/CSS** – editing, validation, query, minification/compression
- **Compare Tools** - file and folder comparisons
- **Git/Subversion Client** – for collaboration
- and more ...

Batch Process a List of Files

- Conversion between XML, JSON, and YAML
- Validate and check well-formedness
- Find and replace in files
- XPath in files
- Format and indent
- Check spelling in files

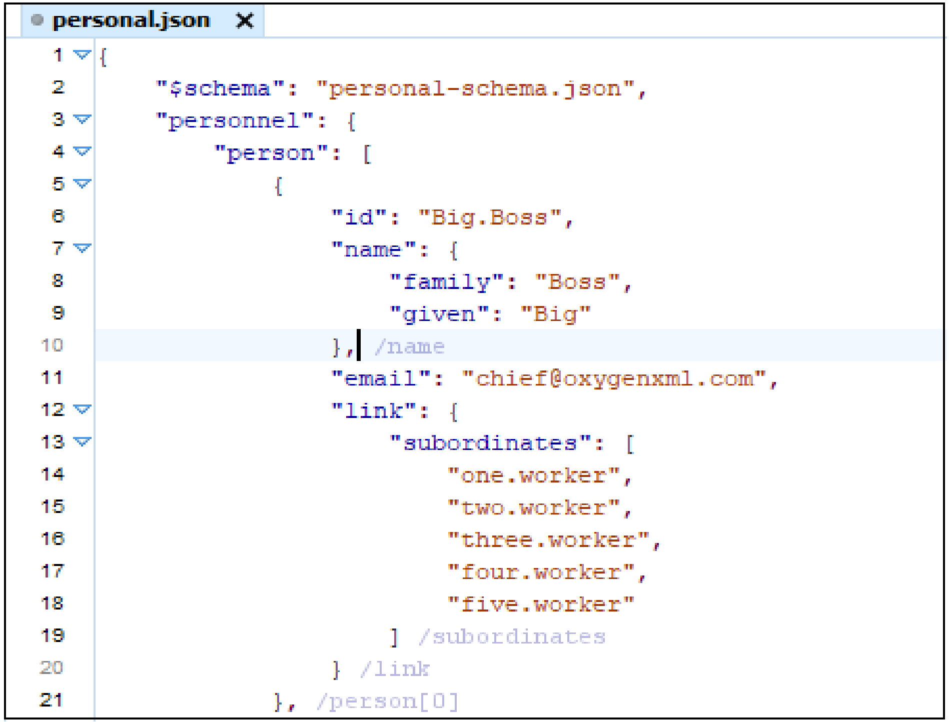


Editing JSON

- JSON Text, Grid, Author Editing Mode
- JSON Outline View
- Validation and Content Completion Assistant based on JSON Schema
- JSON Transformation with XSLT

Text Editing Mode

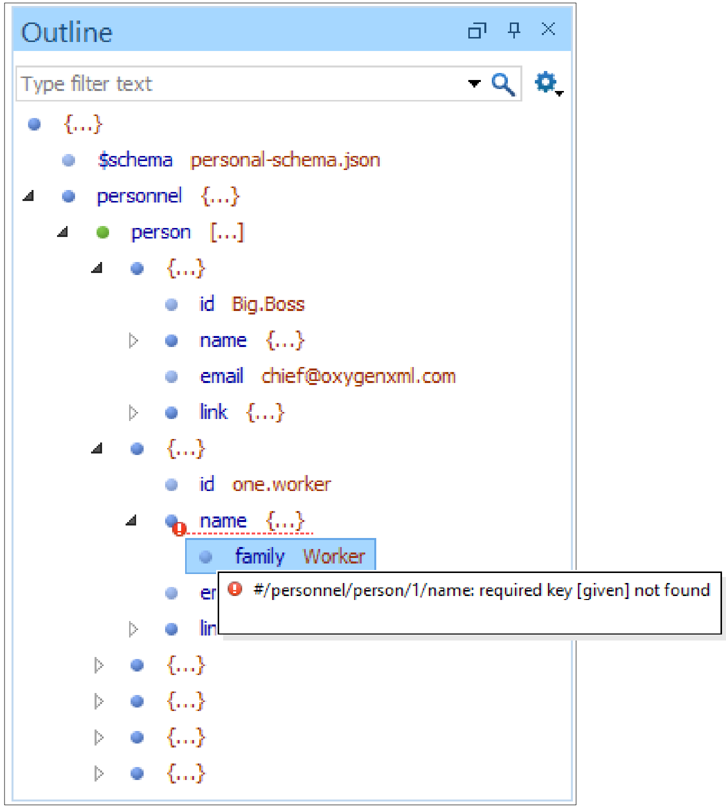
- Syntax Highlights
- Show Property Name After Ending Bracket
- Structure Folding
- Format and Indent
- New Document Templates



```
{
    "$schema": "personal-schema.json",
    "personnel": {
        "person": [
            {
                "id": "Big.Boss",
                "name": {
                    "family": "Boss",
                    "given": "Big"
                }, /name
                "email": "chief@oxygenxml.com",
                "link": {
                    "subordinates": [
                        "one.worker",
                        "two.worker",
                        "three.worker",
                        "four.worker",
                        "five.worker"
                    ] /subordinates
                } /link
            }, /person[0]
```

JSON Outline View

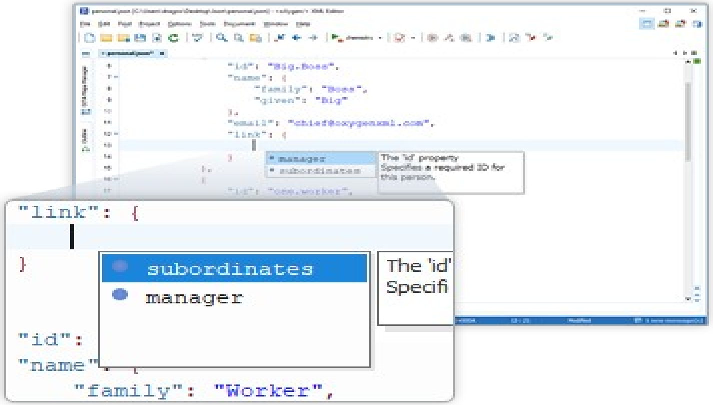
- Document Overview
- Synchronization with Editor
- Filtering
- Error Markers



#/personnel/person/1/name: required key [given] not found

Content Completion

- Context-Sensitive
- Valid Proposals
- Documentation from Schema
- Code Templates
- Learn Structure when no schema is provided



```
"id": "Big.Boss",
"name": {
    "family": "Boss",
    "given": "Big"
},
"email": "chief@oxygenxml.com",
"link": {
    
}
```

- manager
- subordinates

The 'id' property
Specifies a required ID for this person.

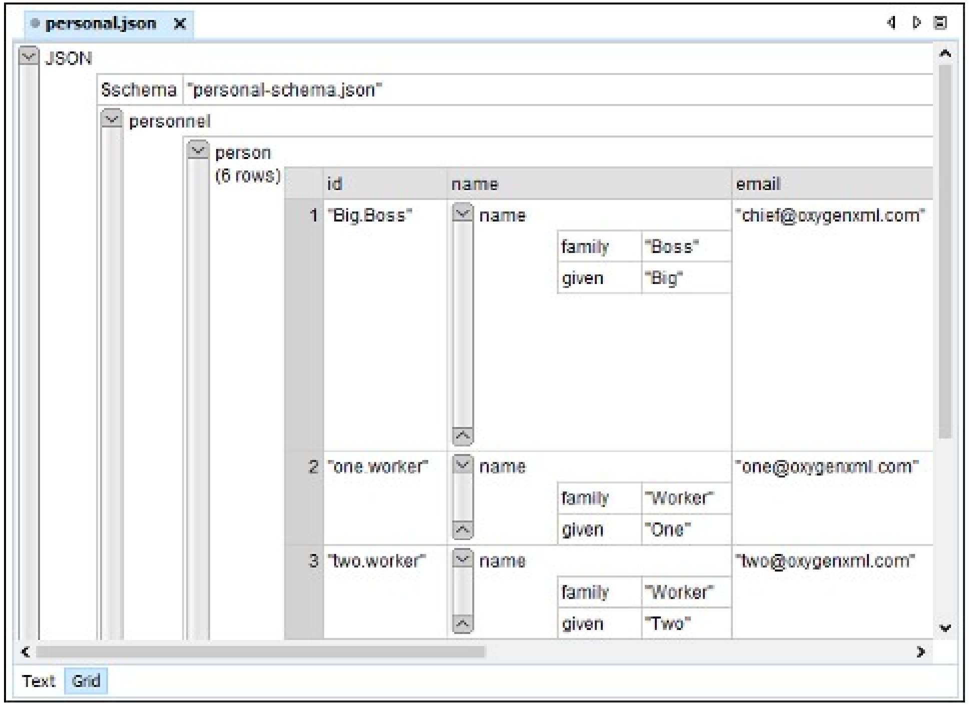
```
"link": {
    |
}

"id":
"name": {
    "family": "Worker",
```

- subordinates
- manager

The 'id'
Specifi

Grid Editing Mode

- JSON content is represented in Grid mode
- JSON data and structure can be easily manipulated
- Table-specific operations or drag-and-drop operations



JSON

$schema "personal-schema.json"

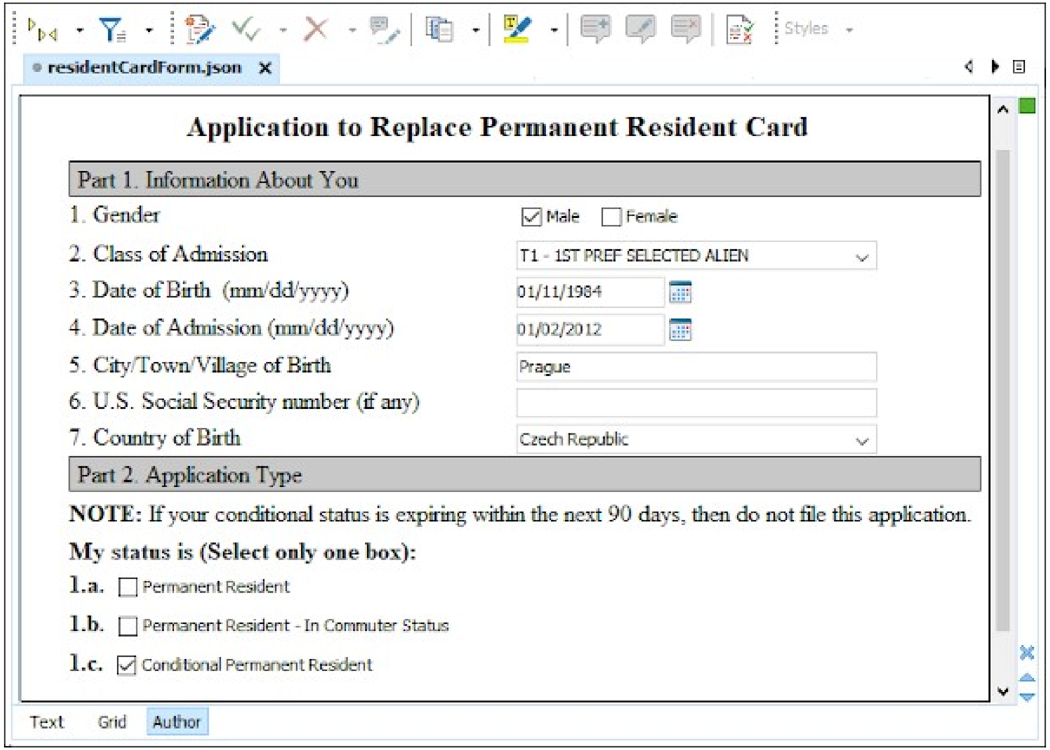
personnel

person (6 rows)

	id	name		email
1	"Big.Boss"	name		"chief@oxygenxml.com"
		family	"Boss"	
		given	"Big"	
2	"one.worker"	name		"one@oxygenxml.com"
		family	"Worker"	
		given	"One"	
3	"two.worker"	name		"two@oxygenxml.com"
		family	"Worker"	
		given	"Two"	

JSON Visual Author Editing Mode

- JSON document is opened in Author mode
- Create your own JSON framework
- Customize using CSS



Application to Replace Permanent Resident Card

Part 1. Information About You

1. Gender ☑ Male □ Female

2. Class of Admission T1 - 1ST PREF SELECTED ALIEN

3. Date of Birth (mm/dd/yyyy) 01/11/1984

4. Date of Admission (mm/dd/yyyy) 01/02/2012

5. City/Town/Village of Birth Prague

6. U.S. Social Security number (if any)

7. Country of Birth Czech Republic

Part 2. Application Type

NOTE: If your conditional status is expiring within the next 90 days, then do not file this application.

My status is (Select only one box):

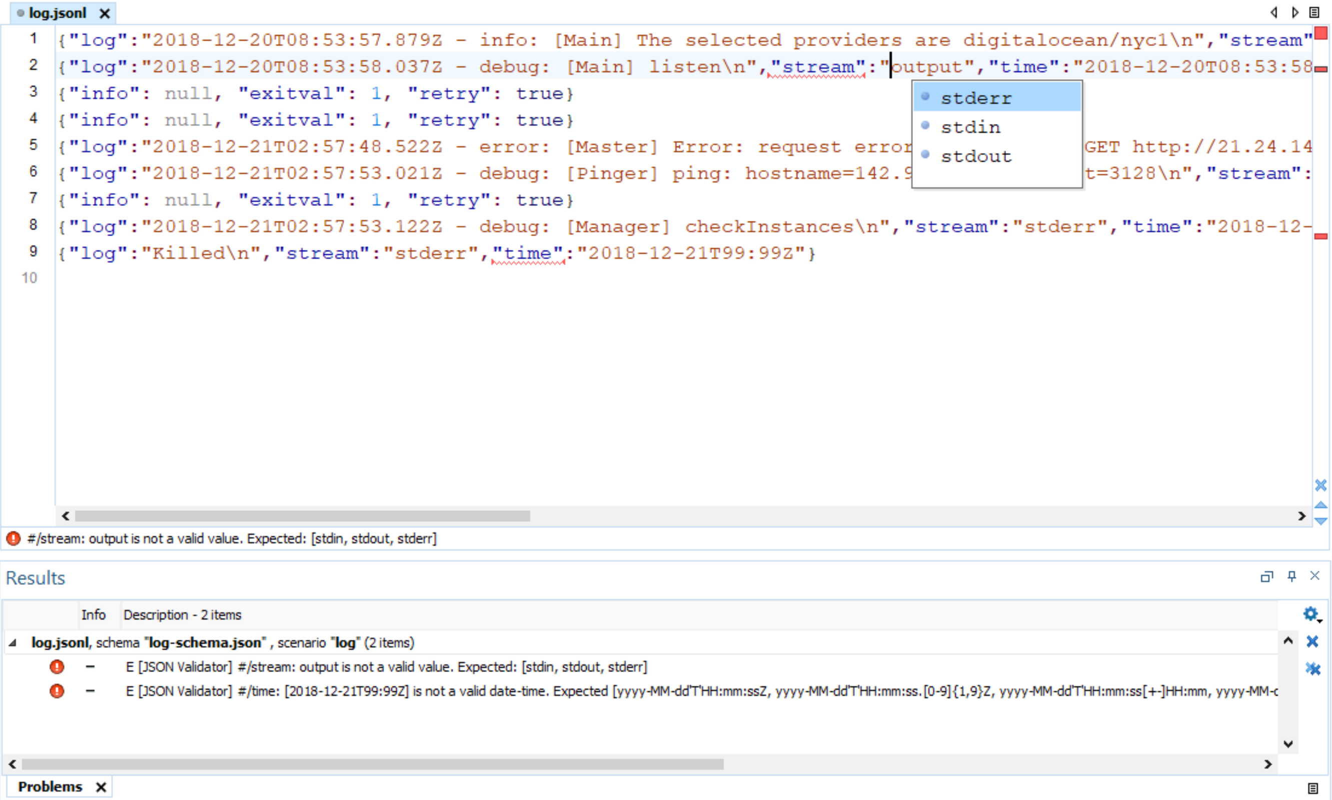
1.a. □ Permanent Resident

1.b. □ Permanent Resident - In Commuter Status

1.c. ☑ Conditional Permanent Resident

JSON Lines

- Editing JSON Lines Documents
- Validation
- Content Completion



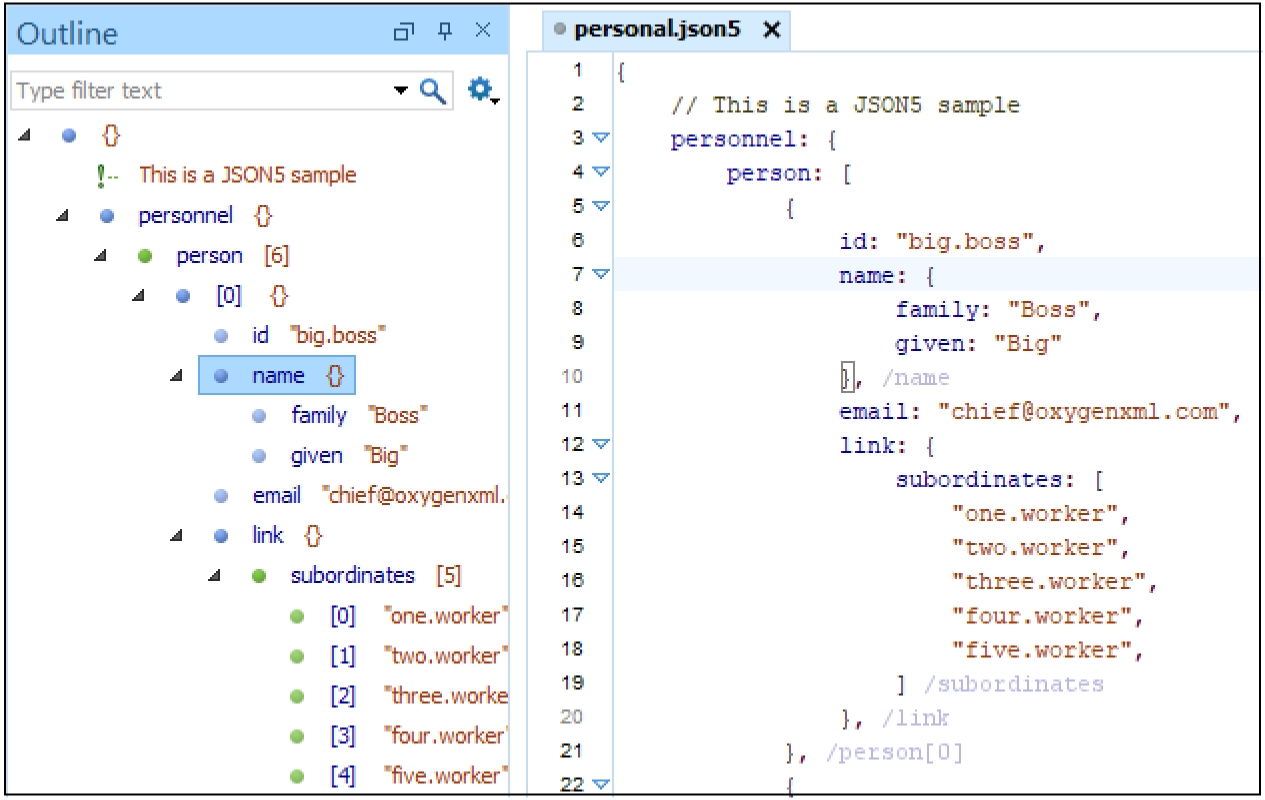
```
{"log":"2018-12-20T08:53:57.879Z - info: [Main] The selected providers are digitalocean/nyc1\n","stream"
{"log":"2018-12-20T08:53:58.037Z - debug: [Main] listen\n","stream":"output","time":"2018-12-20T08:53:58
{"info": null, "exitval": 1, "retry": true}
{"info": null, "exitval": 1, "retry": true}
{"log":"2018-12-21T02:57:48.522Z - error: [Master] Error: request error          GET http://21.24.14
{"log":"2018-12-21T02:57:53.021Z - debug: [Pinger] ping: hostname=142.9          t=3128\n","stream":
{"info": null, "exitval": 1, "retry": true}
{"log":"2018-12-21T02:57:53.122Z - debug: [Manager] checkInstances\n","stream":"stderr","time":"2018-12-
{"log":"Killed\n","stream":"stderr","time":"2018-12-21T99:99Z"}
```

#/stream: output is not a valid value. Expected: [stdin, stdout, stderr]

	Info	Description - 2 items
log.jsonl, schema **"log-schema.json"** , scenario **"log"** (2 items)		
	–	E [JSON Validator] #/stream: output is not a valid value. Expected: [stdin, stdout, stderr]
	–	E [JSON Validator] #/time: [2018-12-21T99:99Z] is not a valid date-time. Expected [yyyy-MM-dd'T'HH:mm:ssZ, yyyy-MM-dd'T'HH:mm:ss.[0-9]{1,9}Z, yyyy-MM-dd'T'HH:mm:ss[+-]HH:mm, yyyy-MM-c

JSON5

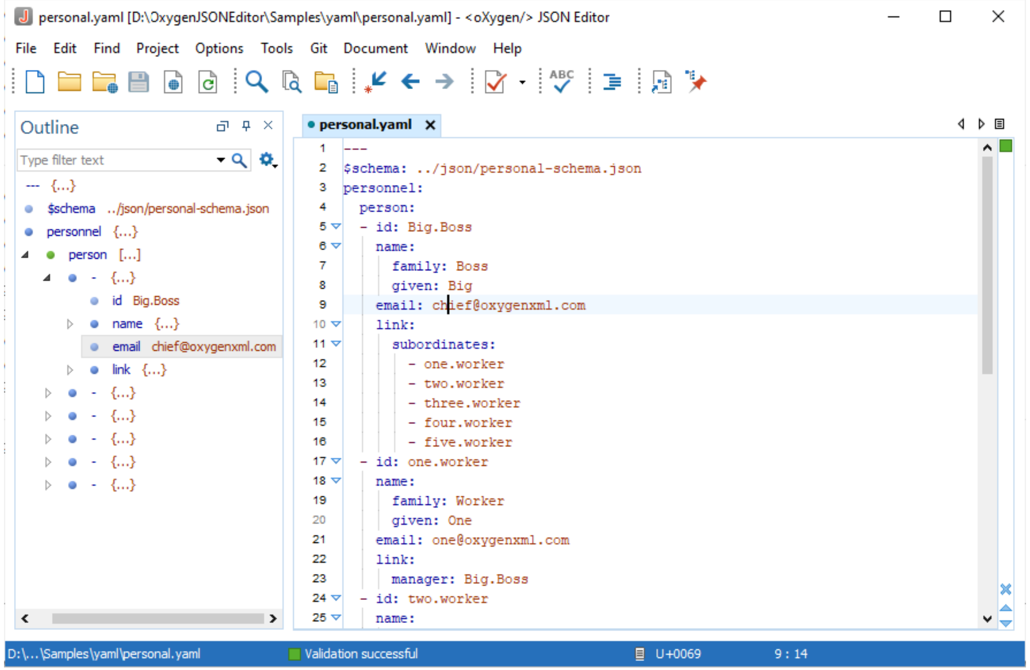
- Editing JSON5 Documents
- Wellform check
- Outliner
- Format and indent



```
{
    // This is a JSON5 sample
    personnel: {
        person: [
            {
                id: "big.boss",
                name: {
                    family: "Boss",
                    given: "Big"
                }, /name
                email: "chief@oxygenxml.com",
                link: {
                    subordinates: [
                        "one.worker",
                        "two.worker",
                        "three.worker",
                        "four.worker",
                        "five.worker",
                    ] /subordinates
                }, /link
            }, /person[0]
            {
```

YAML

- Editor – syntax highlight, outline view, content completion
- Validation - well-formedness check, validate YAML against JSON Schema
- Tools - YAML to JSON and JSON to YAML, batch conversions


personal.yaml [D:\OxygenJSONEditor\Samples\yaml\personal.yaml] - <oXygen/> JSON Editor
File Edit Find Project Options Tools Git Document Window Help
Outline
Type filter text
personal.yaml

--- {...}
$schema ../json/personal-schema.json
personnel {...}
person [...]
- {...}
id Big.Boss
name {...}
email chief@oxygenxml.com
link {...}
- {...}
- {...}
- {...}
- {...}
- {...}

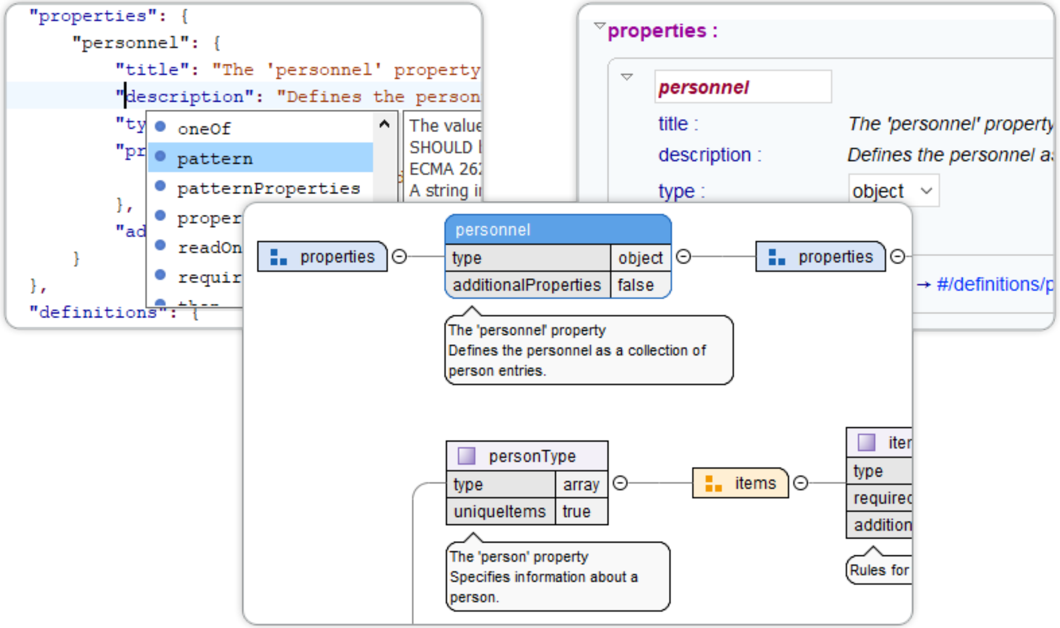
```
---
$schema: ../json/personal-schema.json
personnel:
  person:
  - id: Big.Boss
    name:
      family: Boss
      given: Big
    email: chief@oxygenxml.com
    link:
      subordinates:
        - one.worker
        - two.worker
        - three.worker
        - four.worker
        - five.worker
  - id: one.worker
    name:
      family: Worker
      given: One
    email: one@oxygenxml.com
    link:
      manager: Big.Boss
  - id: two.worker
    name:
```

D:\...\Samples\yaml\personal.yaml   Validation successful   U+0069   9 : 14

JSON Schema Editor

Design, develop, and edit JSON Schemas in:

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

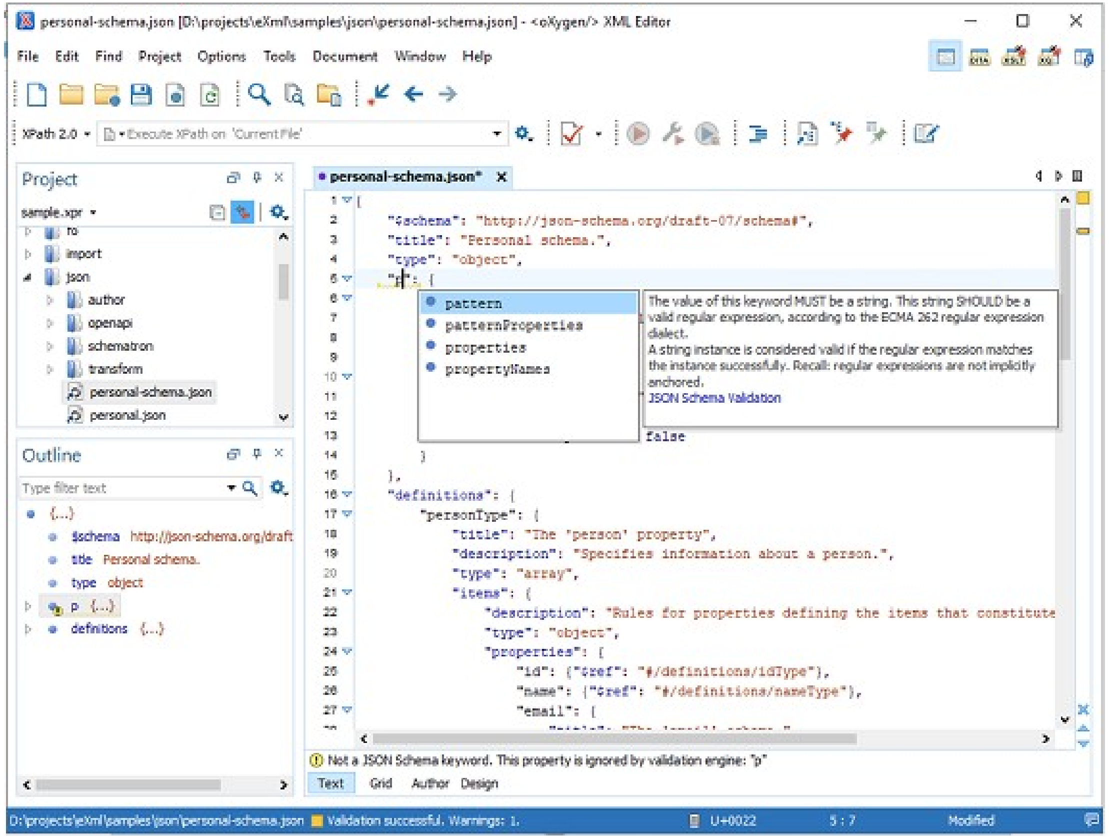


```
"properties": {
    "personnel": {
        "title": "The 'personnel' property
        "description": "Defines the person
        "ty
        "pr
        },
        "ad
    }
},
"definitions": {
```

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)



```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Personal schema.",
    "type": "object",
    "p": {
                                   false
        }
    },
    "definitions": {
        "personType": {
            "title": "The 'person' property",
            "description": "Specifies information about a person.",
            "type": "array",
            "items": {
                "description": "Rules for properties defining the items that constitute
                "type": "object",
                "properties": {
                    "id": {"$ref": "#/definitions/idType"},
                    "name": {"$ref": "#/definitions/nameType"},
                    "email": {
```

The value of this keyword MUST be a string. This string SHOULD be a valid regular expression, according to the ECMA 262 regular expression dialect.
A string instance is considered valid if the regular expression matches the instance successfully. Recall: regular expressions are not implicitly anchored.
JSON Schema Validation

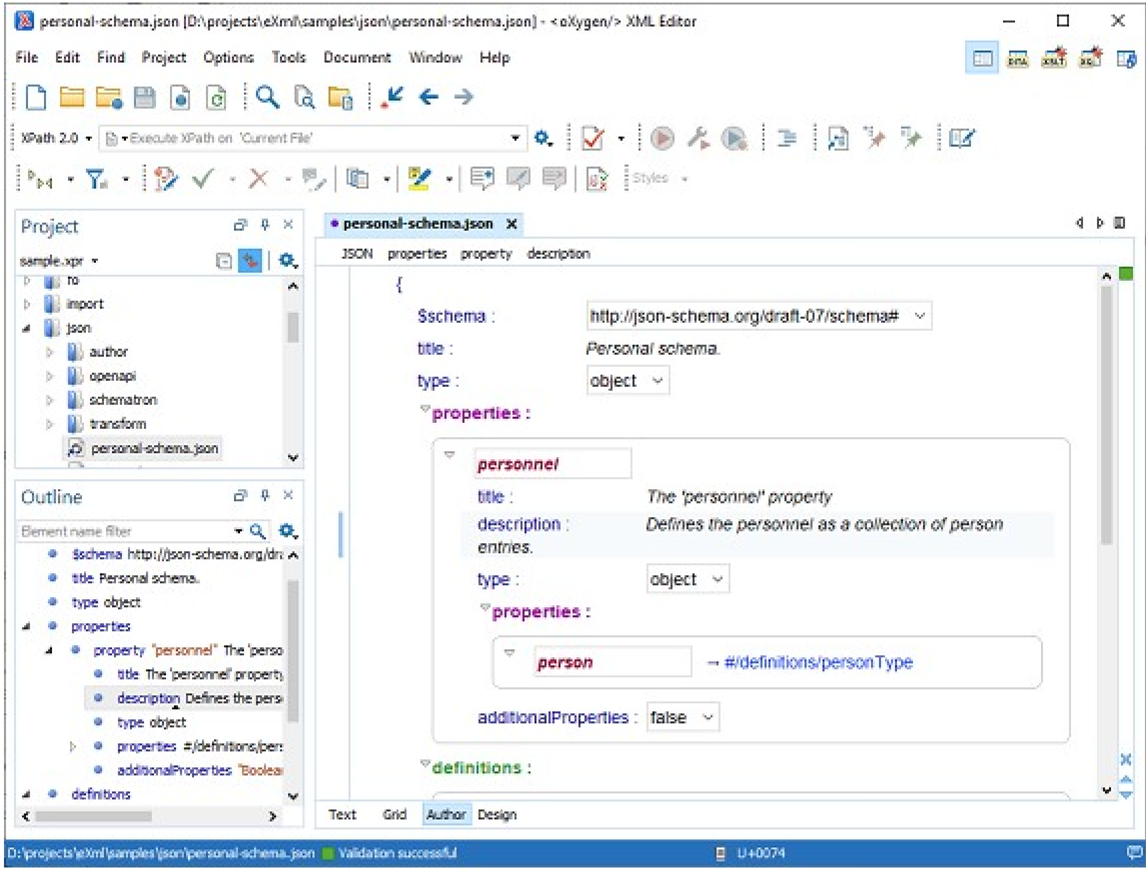
Not a JSON Schema keyword. This property is ignored by validation engine: "p"

D:\projects\eXml\samples\json\personal-schema.json Validation successful. Warnings: 1. U+0022 5 : 7 Modified

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


personal-schema.json [D:\projects\eXml\samples\json\personal-schema.json] - <oXygen/> XML Editor
File Edit Find Project Options Tools Document Window Help
XPath 2.0
Execute XPath on 'Current File'
Styles
Project
sample.xpr
fo
import
json
author
openapi
schematron
transform
personal-schema.json
Outline
Element name filter
personal-schema.json
JSON properties property description
Text Grid Author Design

$schema http://json-schema.org/dr
title Personal schema.
type object
properties
property "personnel" The 'perso
title The 'personnel' property
description Defines the pers
type object
properties #/definitions/pers
additionalProperties "Boolea
definitions

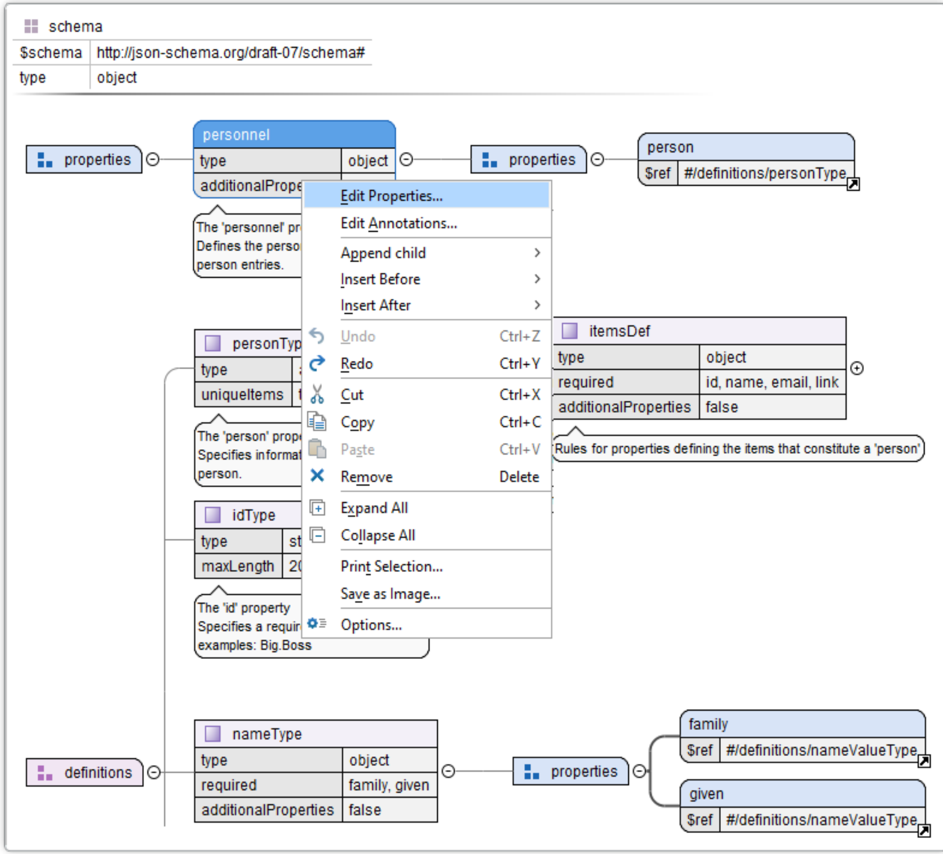
{
$schema : http://json-schema.org/draft-07/schema#
title : Personal schema.
type : object
properties :
personnel
title : The 'personnel' property
description : Defines the personnel as a collection of person entries.
type : object
properties :
person → #/definitions/personType
additionalProperties : false
definitions :

D:\projects\eXml\samples\json\personal-schema.json Validation successful U+0074

Schema Design Mode

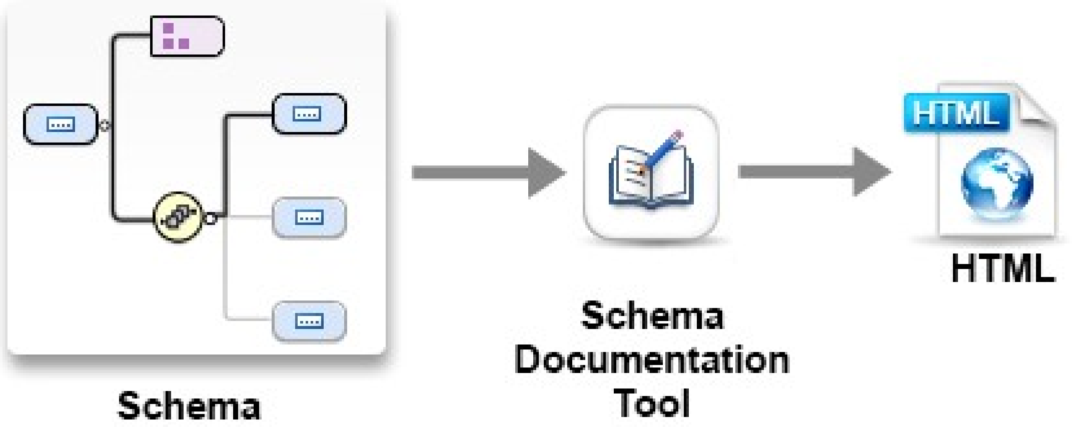
Visualize, edit, and understand JSON Schemas

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



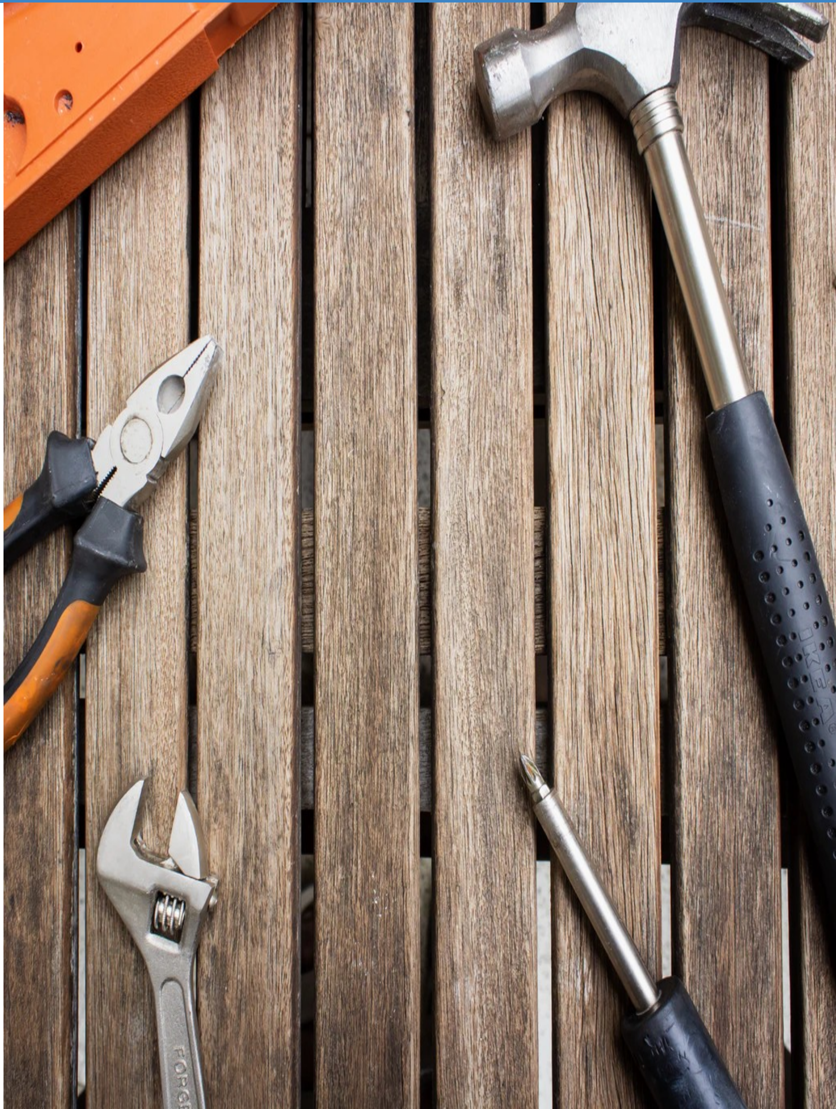
Generate JSON Schema Documentation

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



JSON Tools

- Convert between JSON and XML
- Convert between JSON and YAML
- Generate sample JSON files from JSON Schema
- Generate JSON Schema from a JSON instance
- XSD to JSON Schema converter
- Convert JSON Schema to latest versions (2020-12, 2019-09)



OpenAPI/AsyncAPI Documents

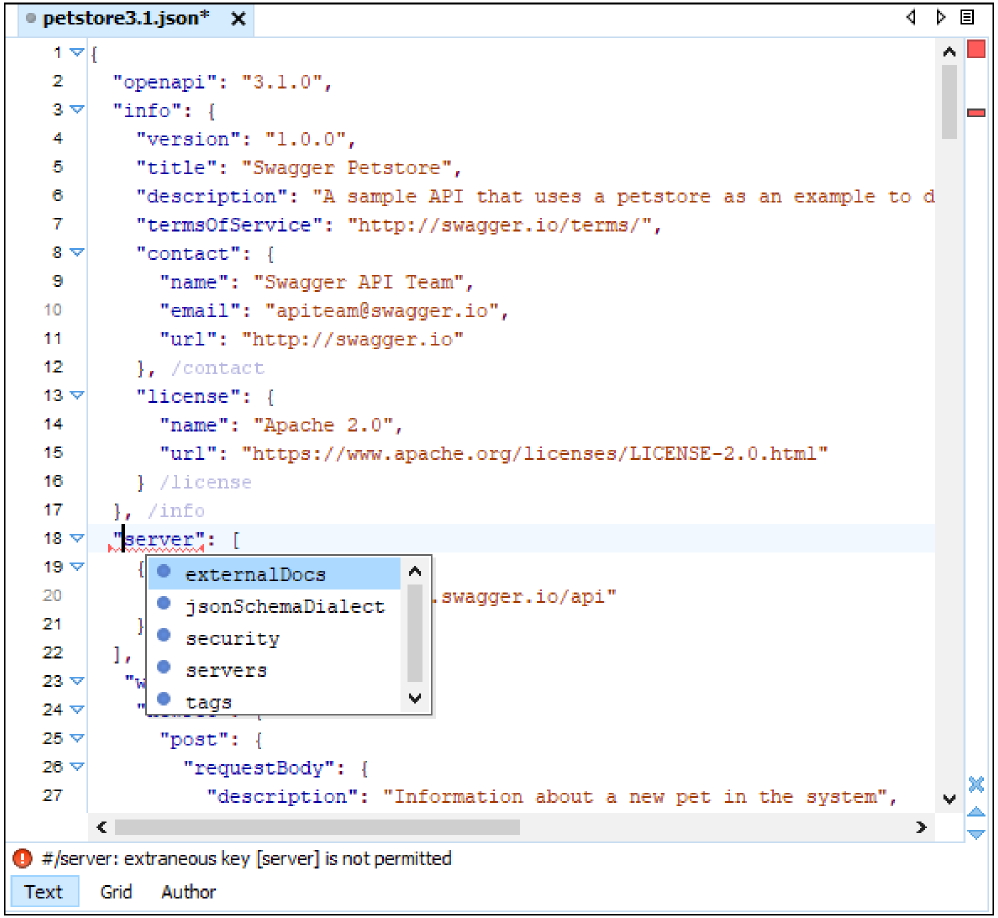
- Can be written in JSON or YAML
- Describe the inputs and outputs of an API
- Include information such as:
  - where the API is hosted
  - what authorization is required to access it
  - details for consumers and producers

OPENAPI

AsyncAPI

OpenAPI/AsyncAPI Editor

- Syntax highlights
- Validation conforms with the schema
- Content completion based on the schema
- Outline view
- Search and replace



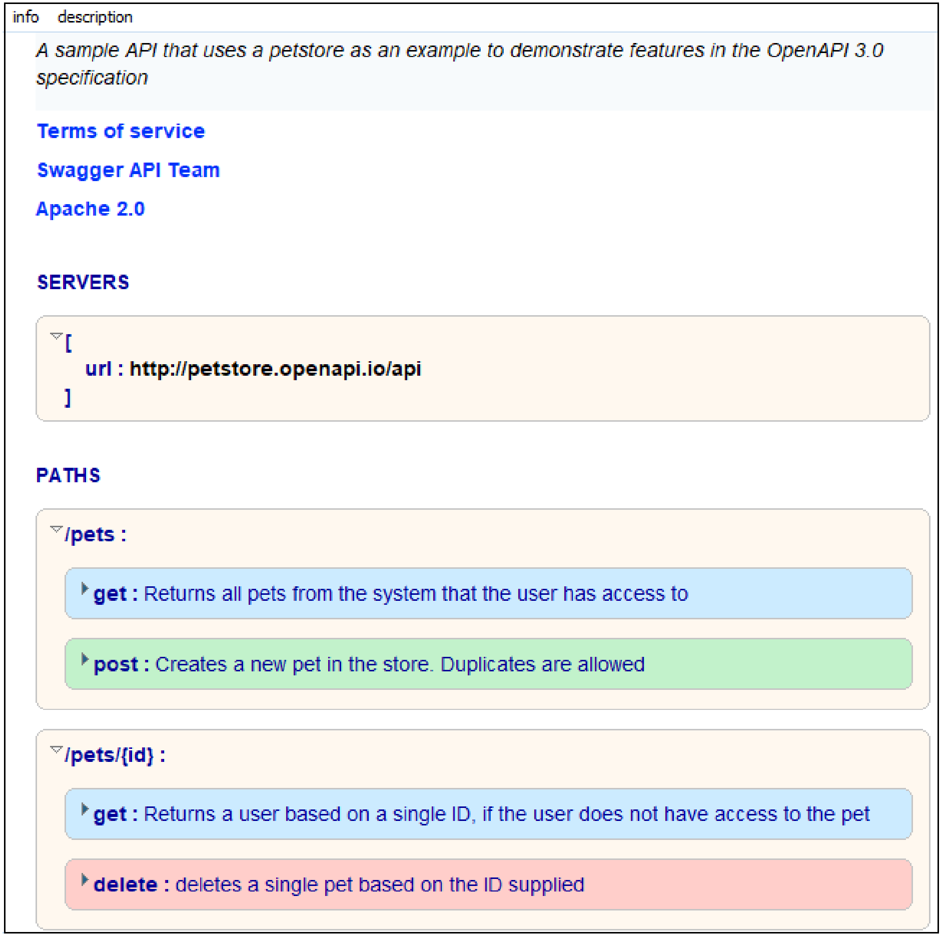
petstore3.1.json*

```
{
  "openapi": "3.1.0",
  "info": {
    "version": "1.0.0",
    "title": "Swagger Petstore",
    "description": "A sample API that uses a petstore as an example to d
    "termsOfService": "http://swagger.io/terms/",
    "contact": {
      "name": "Swagger API Team",
      "email": "apiteam@swagger.io",
      "url": "http://swagger.io"
    }, /contact
    "license": {
      "name": "Apache 2.0",
      "url": "https://www.apache.org/licenses/LICENSE-2.0.html"
    } /license
  }, /info
  "server": [
    {
                    .swagger.io/api"
    }
  ],
   "w
    "
      "post": {
        "requestBody": {
          "description": "Information about a new pet in the system",
```

#/server: extraneous key [server] is not permitted

OpenAPI/AsyncAPI Visual Editor

- Edit documentation
- Form controls to change values
- Collapsible sections
- Customize interface with CSS



info description

A sample API that uses a petstore as an example to demonstrate features in the OpenAPI 3.0 specification

Terms of service

Swagger API Team

Apache 2.0

SERVERS

```
[
  url : http://petstore.openapi.io/api
]
```

PATHS

/pets :

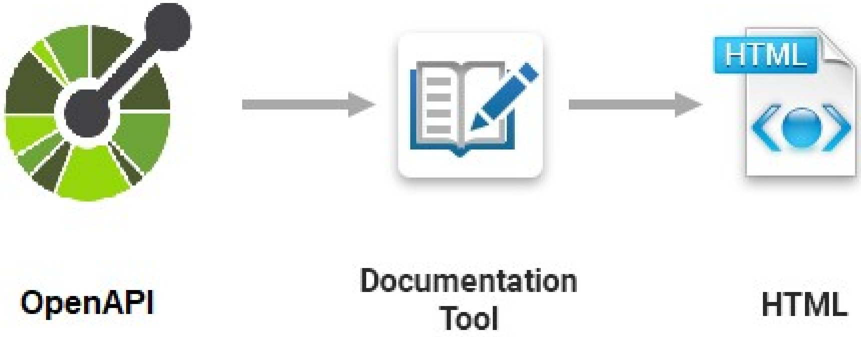
- get : Returns all pets from the system that the user has access to
- post : Creates a new pet in the store. Duplicates are allowed

/pets/{id} :

- get : Returns a user based on a single ID, if the user does not have access to the pet
- delete : deletes a single pet based on the ID supplied

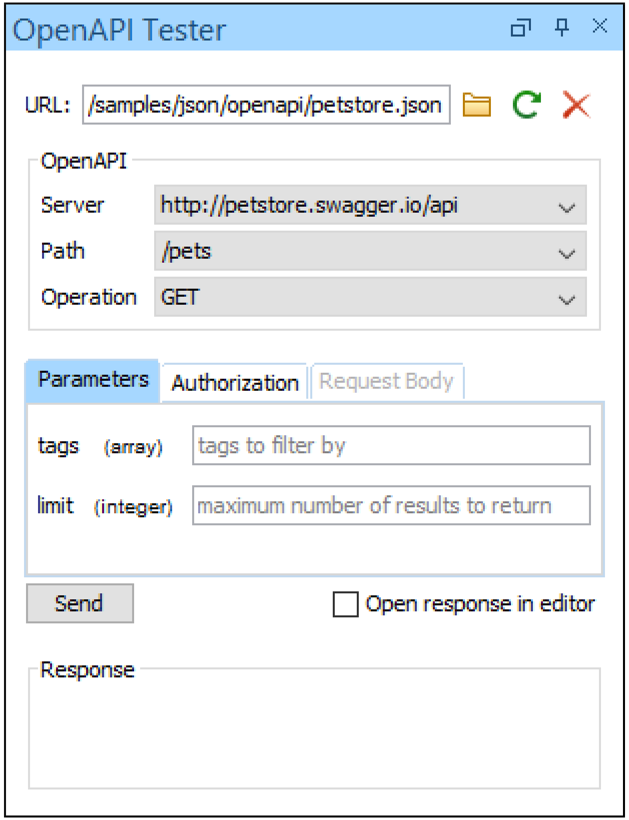
OpenAPI Documentation

- Tool for generating detailed documentation for OpenAPI documents in HTML/DITA format



OpenAPI Tester Tool

- GUI interface testing tool for OpenAPI files
- Execute API requests and validate responses on the fly
- OpenAPI files in JSON or YAML format
- Run a test scenario



OpenAPI Tester

URL: /samples/json/openapi/petstore.json

OpenAPI

Server: http://petstore.swagger.io/api

Path: /pets

Operation: GET

tags (array): tags to filter by

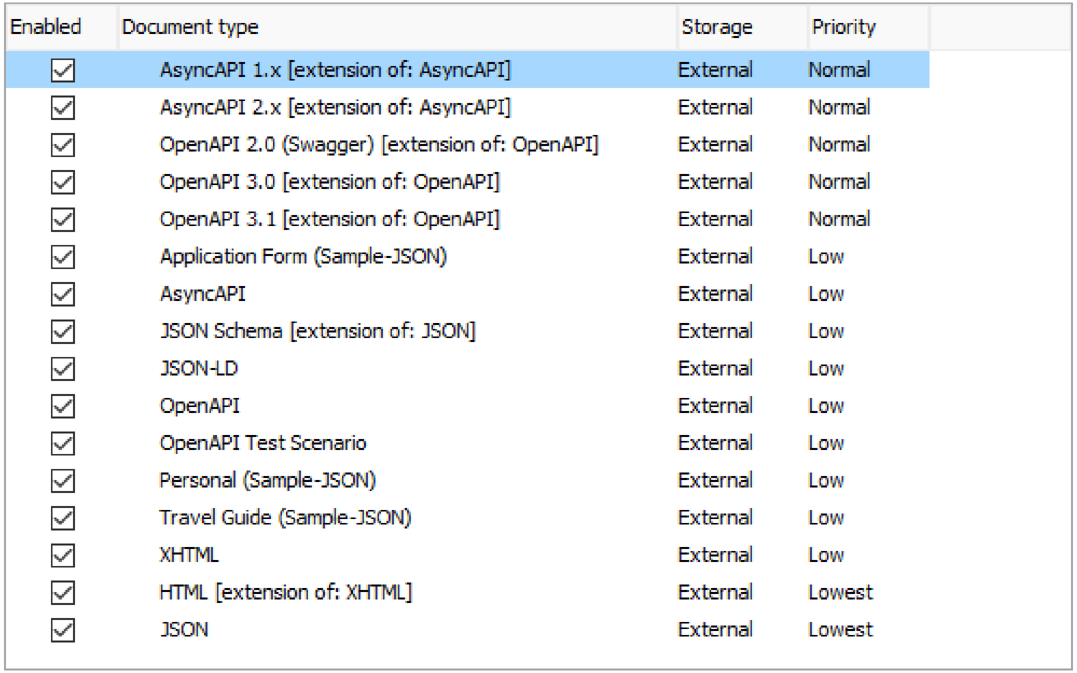
limit (integer): maximum number of results to return

□ Open response in editor

Response

Other Frameworks

- JSON-LD - is a lightweight Linked Data format
- OpenAPI Test Scenario – test suite for an OpenAPI
- Travel Guide – sample
- Application Form – sample
- Personal – sample
- *framework for any JSON*



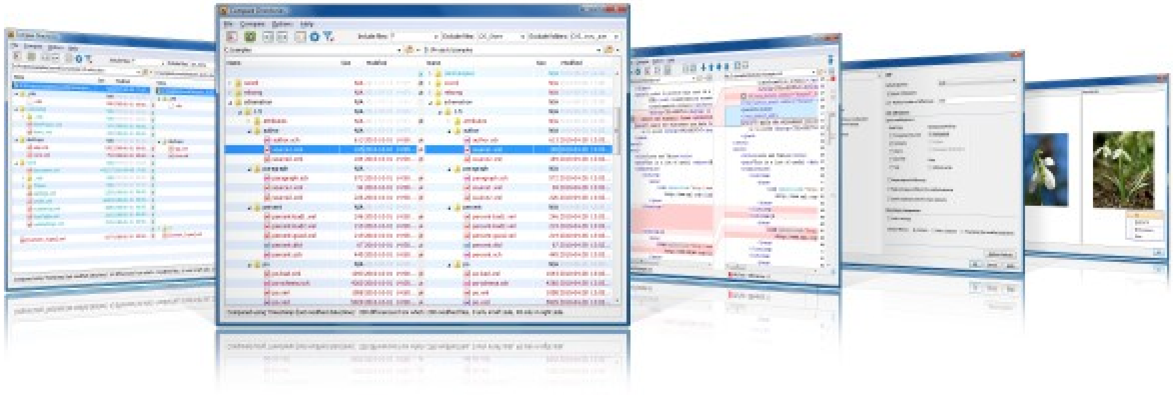
Enabled	Document type	Storage	Priority
☑	AsyncAPI 1.x [extension of: AsyncAPI]	External	Normal
☑	AsyncAPI 2.x [extension of: AsyncAPI]	External	Normal
☑	OpenAPI 2.0 (Swagger) [extension of: OpenAPI]	External	Normal
☑	OpenAPI 3.0 [extension of: OpenAPI]	External	Normal
☑	OpenAPI 3.1 [extension of: OpenAPI]	External	Normal
☑	Application Form (Sample-JSON)	External	Low
☑	AsyncAPI	External	Low
☑	JSON Schema [extension of: JSON]	External	Low
☑	JSON-LD	External	Low
☑	OpenAPI	External	Low
☑	OpenAPI Test Scenario	External	Low
☑	Personal (Sample-JSON)	External	Low
☑	Travel Guide (Sample-JSON)	External	Low
☑	XHTML	External	Low
☑	HTML [extension of: XHTML]	External	Lowest
☑	JSON	External	Lowest

Other File Types Supported

- XML Documents supported (Syntax Highlight, Content completion, Validation)
- JavaScript documents
- Markdown documents
- Python documents
- Text, Java, SQL, Properties, Batch, Shell, Power Shell, Dockerfile, and PHP.

Compare

- Compare and merge files (2-way and 3-way)
- Compare and merge directories (2-way and 3-way)
- Generate comparison report



Versioning Systems

- Git Client – add-on that contributes a built-in Git client directly in Oxygen JSON Editor
- Subversion (SVN) Client

Conclusion

Oxygen JSON Editor

- Tool specialized for JSON
- Great features
- Cross-platform application
- Affordable pricing

Future Plans

- JSON Transformation with XQuery (in 27.0)
- Improve OpenAPI/AsyncAPI support
- Support for ChatGPT fine-tune files
- *feedback is welcome*

Questions?

Mircea Enachescu
Product Architect at Syncro Soft

mircea@oxygenxml.com