

# JSON and JSON Schema Support



Octavian Nadolu, Syncro Soft  
octavian\_nadolu@oxygenxml.com  
@OctavianNadolu

schematron  
Structured editing  
XML  
review  
XQuery  
Publish  
PDF  
DTD  
oxygen  
authoring  
XML Editor  
XSD  
XPR  
frameworks  
Profiling  
WSDL  
styles  
visual  
WebHelp  
DITA  
TEI  
XSL  
PHP  
Ant  
Js

JS  
KML  
XSLT  
SVN  
JSON  
SVG  
IDREFS  
WebDAV  
DocBook  
Single  
Sourc  
Datab  
XHTML  
Cha  
Col  
W

© 2020 Syncro Soft SRL. All rights reserved.

## Agenda

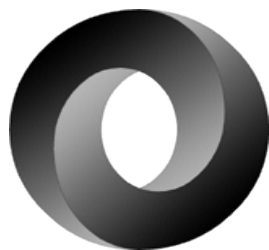
- JSON and JSON Schema in Oxygen
- Validating JSON with Schematron
- Convert between JSON and XML
- XSD to JSON Schema converter
- Instance generator from a JSON Schema
- Generate JSON Schema from JSON instance



# JSON

(JavaScript Object Notation)

- Data representation format
- Used for API and Configs
- Lightweight and easy to read/write

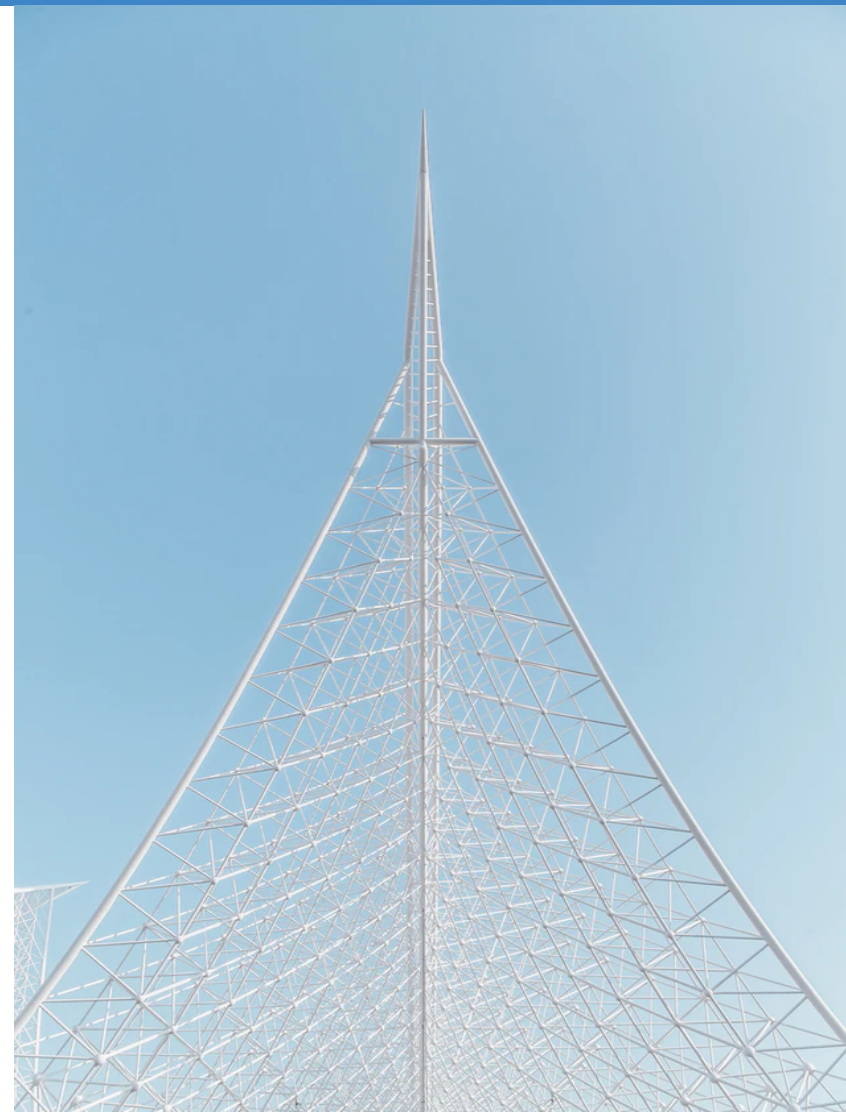


<http://json.org>



## JSON Documents Structure

- JSON Data – name/value pair
- JSON Object {...}
- JSON Array [...]





## Question: How do you use JSON?

- As a data interchange format
- For storing data
- I am using JSON related formats (such as: JSONLD, JSON-RPC)
- In a different way (use the Questions pane to describe that)



## Validating JSON Documents

- Checking **Well-Formedness** in JSON Documents
- **Validating JSON** Documents Against JSON Schema and Schematron
- **Validating JSON Schema** According to the Specification



## Checking Well-Formedness

Check if the JSON document respects the JSON specification

```
{  
  "id": "jane.doe",  
  "firstname": "Jane",  
  "lastname": "Doe",  
  "email": "jane@oxygenxml.com",  
  "age": 37  
}
```

ECMA-404 The JSON Data Interchange Standard

<http://json.org/>

## Example

```
{
  "persons" [
    {
      "id": "jane.doe",
      "firstname": "Jane",
      "lastname": "Doe",
      "email": "jane@oxygenxml.com"
      "age": 37,
    },
    {
      "id": "john.jones,
      "firname": "John",
      "lastname": "Jones,
      "email": "john@oxygenxml.com",
      "age": 42
    }
  ]
}
```

Expected a ':' after a key

Expected ',' character

Unexpected ',' character

Expected quotes

Expected '}'



## JSON Schema

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



<http://json-schema.org>



## Defining JSON Schema

- Similar to XML Schema, RNG, or DTD
- Written in JSON
- Used to define the structure of JSON data

```
{"type": "string"}
```

JSON Schema

```
"I'm a string"
```

JSON Instance

## 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/2019-09 (Draft 8)
- 

## Question: Do you use JSON Schema, what versions?

- No
- Yes. Draft 4, 6 or 7
- Yes. Draft 8 (2019-09)
- Yes. Other (use the Questions pane to provide more details)





## Associate JSON Schema

- Associating a Schema to JSON Documents
  - Directly in JSON document – using `$schema` property

```
{  
  "$schema": "person-schema.json",  
  "persons" [ ... ]  
}
```

→ Absolute or relative URI

- In application options, through a Validation Scenario

## Validating JSON Schema

- Check for well-formedness
- Validate according to the Internet Engineering Task Force (IETF) Specification



## Validating JSON with Schematron

- Specify the JSON structure using JSON Schema
- Express co-constraints and define custom rules using Schematron



## Schematron for JSON

- Use XPath to express the rules

```
<sch:rule context="persons">  
  <sch:assert test="number(age) > preceding-sibling::node()/age ">  
    The person age must be grater than the previous person age </sch:assert>  
</sch:rule>
```



## Editing JSON

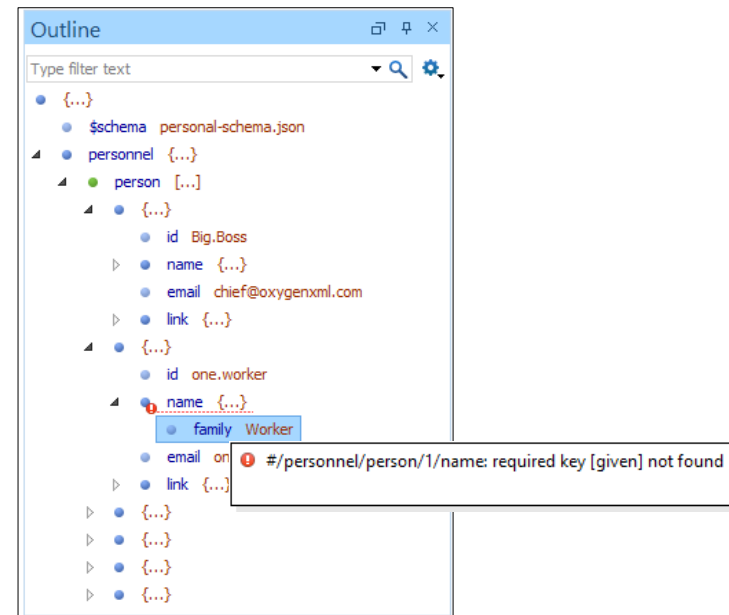
- JSON Text and Grid Editing Mode
- JSON Outline View
- Content Completion Assistant based on JSON Schema





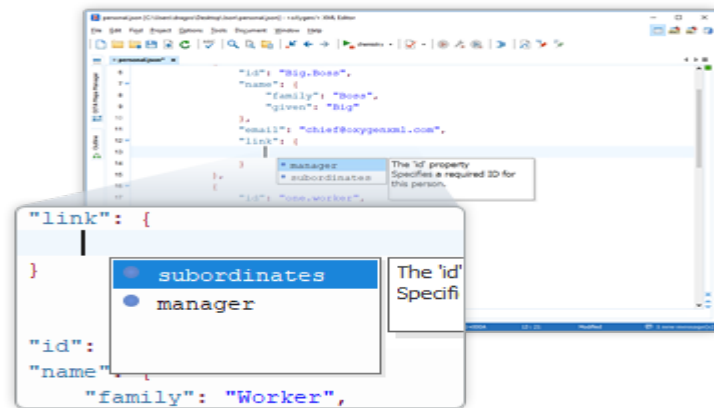
## JSON Outline View

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



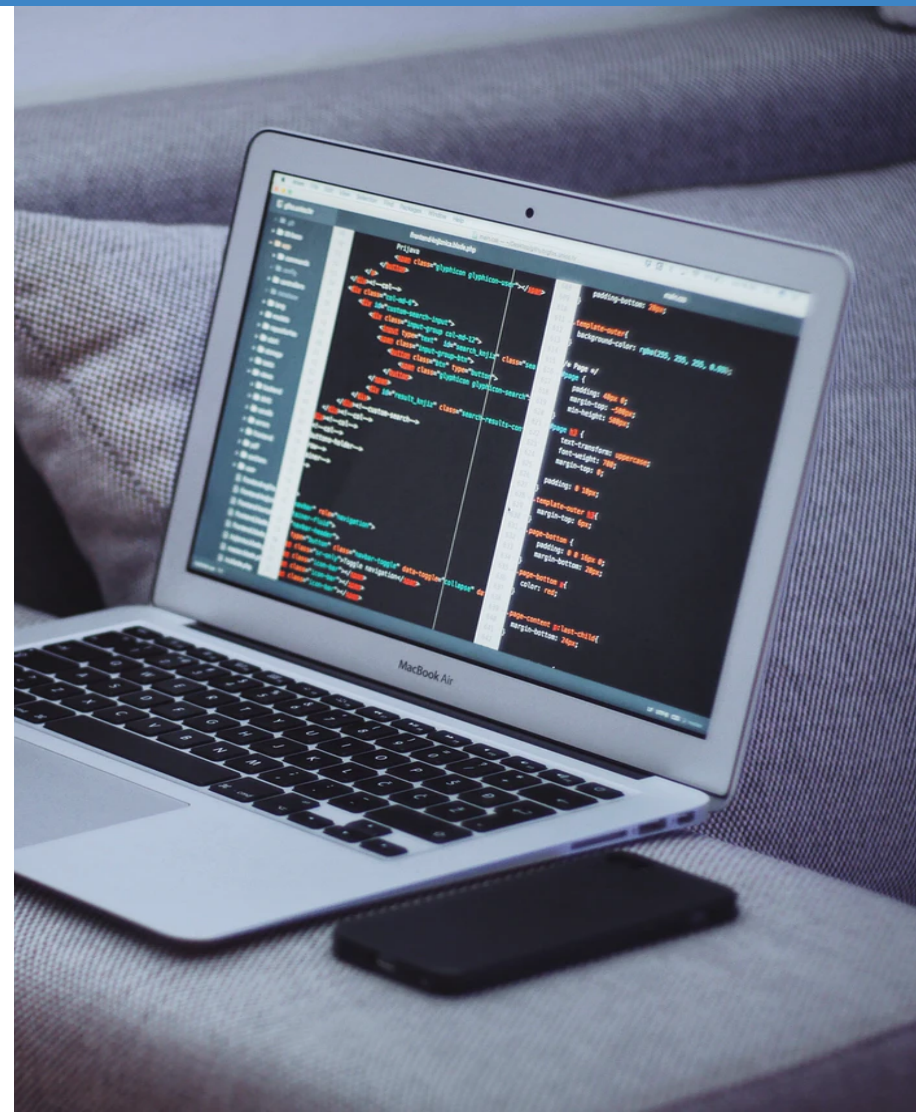
# Content Completion

- Context-Sensitive
- Valid Proposals
- Documentation from Schema
- Code Templates



## Querying and Transforming

- Query using XPath
- Transform JSON using XSLT or XQuery



# XPath

- XPath Toolbar and XPath Builder view
- Content Completion Assistant
- XPath over multiple files

XPath 2.0 `//*[@link/manager/text()='Big.Boss']/name/given`

personal.json

```
7 "id": "Big.Boss",
8 "name": {
9   "family": "Boss",
10  "given": "Big"
11 },
12 "email": "chief@oxygenxml.com",
13 "link": {
```

Description - 5 items	XPath location
One	<code>/JSON[1]/personnel[1]/person[2]/name[1]/given[1]</code>
Two	<code>/JSON[1]/personnel[1]/person[3]/name[1]/given[1]</code>
Three	<code>/JSON[1]/personnel[1]/person[4]/name[1]/given[1]</code>
Four	<code>/JSON[1]/personnel[1]/person[5]/name[1]/given[1]</code>

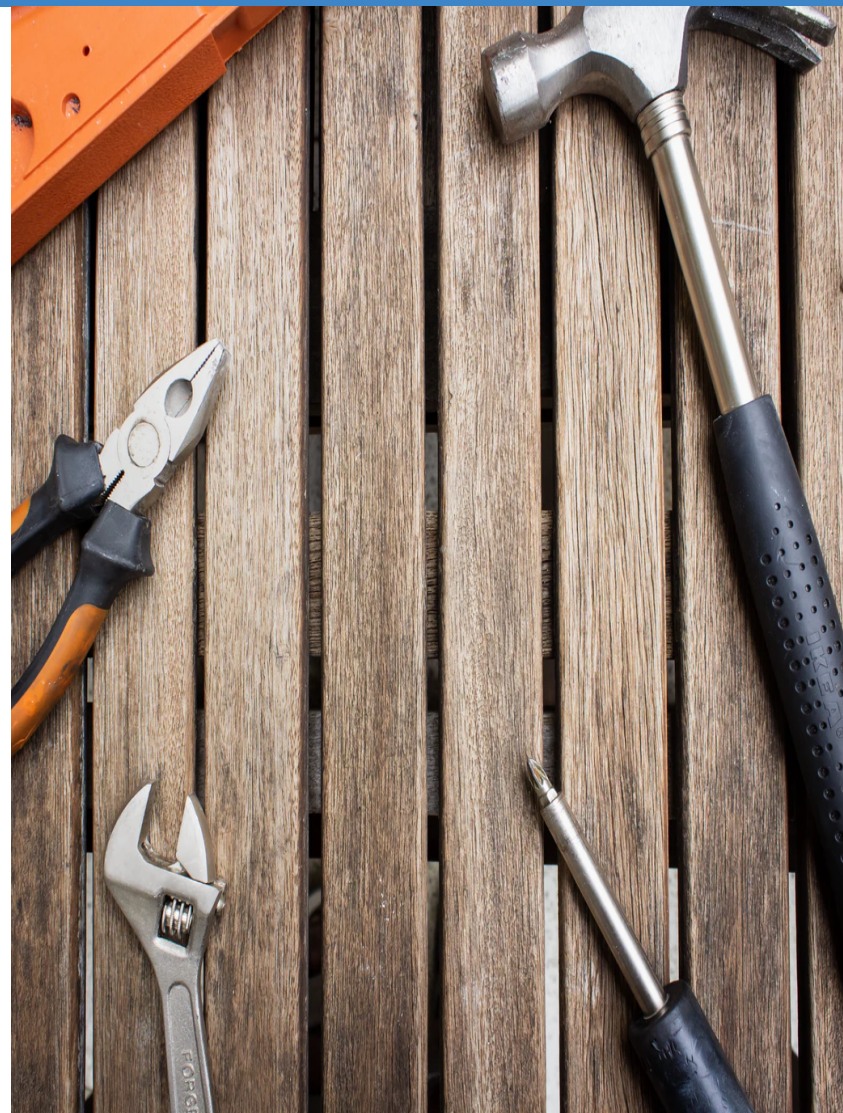
## Transform using XSLT/XQuery

- Transform JSON documents to various formats
- Create XSLT/XQuery transformation scenario for JSON
- XQuery Builder view
- Functions to process JSON documents:
  - `json-doc($href as xs:string?) as item()?`
  - `json-to-xml($json-text as xs:string?) as document-node()?`



## JSON Tools

- Convert between **JSON** and **XML**
- **XSD to JSON Schema** converter
- **Instance generator** from a JSON Schema
- **Generate JSON Schema** from a JSON instance



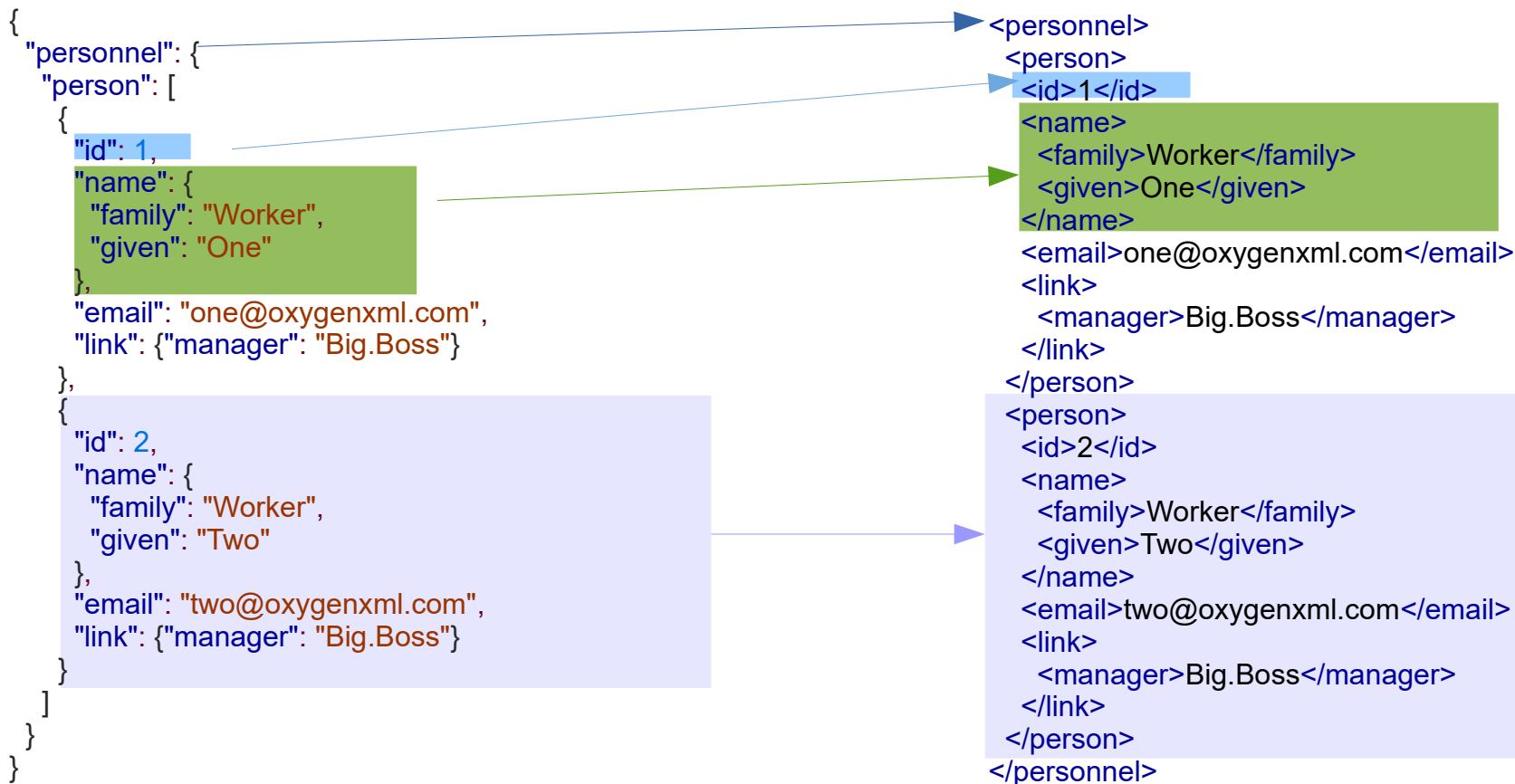


## Question: What JSON tools do you use or intend to use?

- ❑ Convert between JSON and XML
- ❑ XSD to JSON Schema converter
- ❑ Instance generator from a JSON Schema
- ❑ Generate JSON Schema from a JSON instance
- ❑ Other (please use the Questions tab to provide more details)



# JSON to XML



## JSON to XML Conversion Details

- A `<JSON>` element is added as root if the converted JSON has multiple properties on the first level

```
{  
  "person": "one",  
  "id": "personnel-id"  
}
```

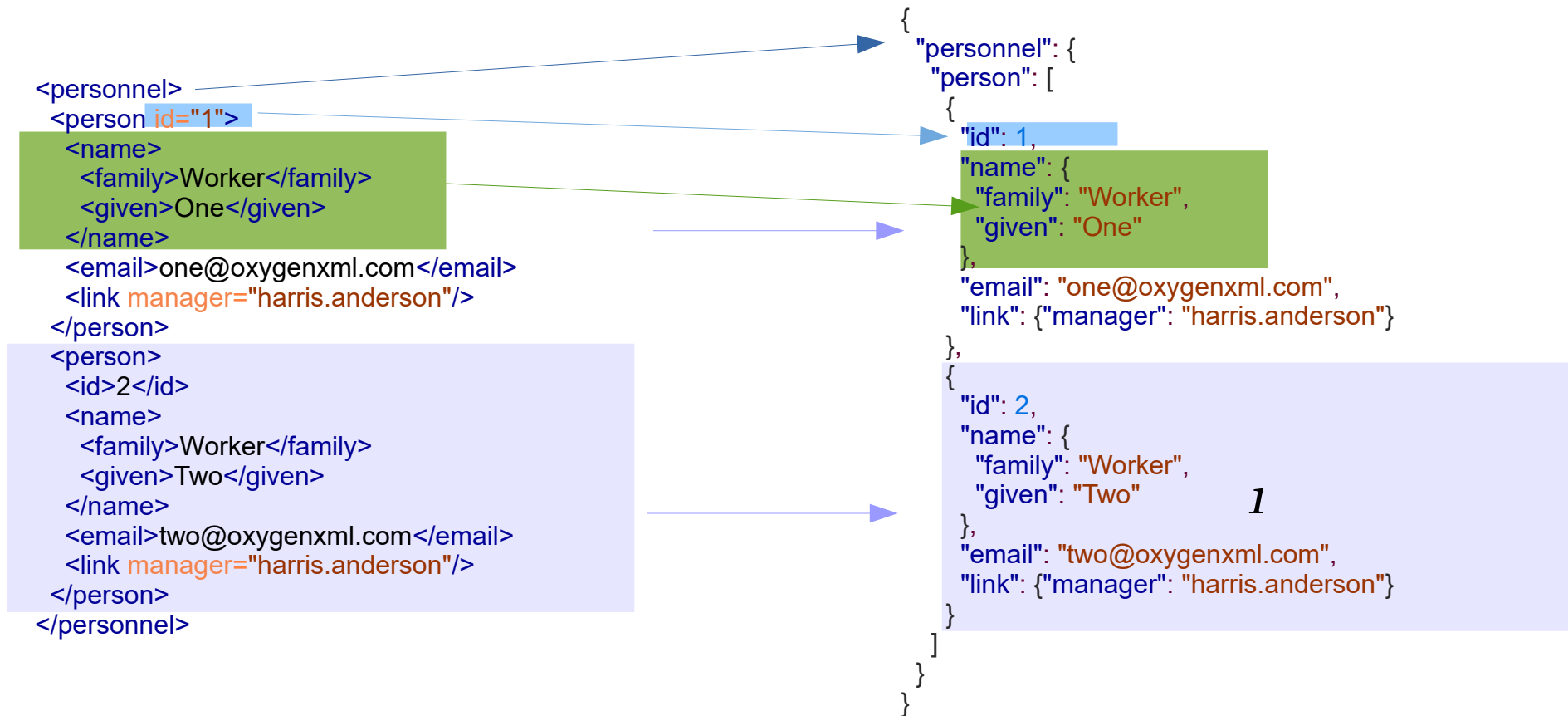
```
<JSON>  
  <person>one</person>  
  <id>personnel-id</id>  
</JSON>
```

- An `<array>` element is added as root if the converted JSON is an array

```
[  
  {"name": "Boss"},  
  {"name": "Worker"}  
]
```

```
<array>  
  <array>  
    <name>Boss</name>  
  </array>  
  <array>  
    <name>Worker</name>  
  </array>  
</array>
```

# XML to JSON



## XML to JSON Conversion Details

- XML attributes are converted to properties

```
<person id="1"/>           {"person": {"id": 1}}
```

- Multiple elements with the same name are converted into an array

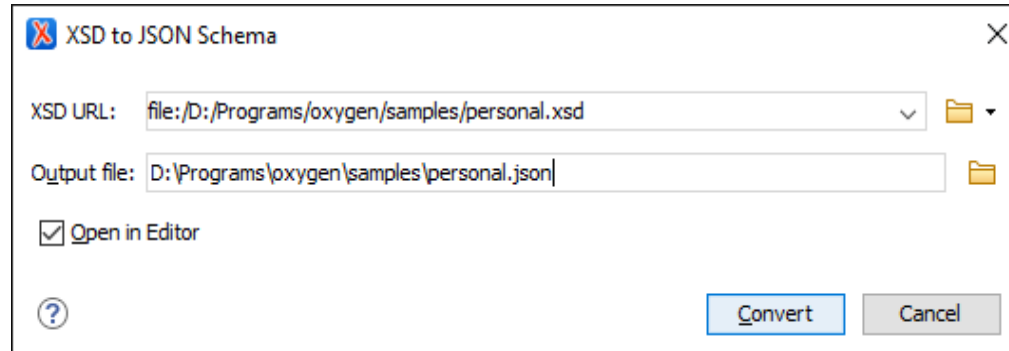
```
<person>One</person>      "person": ["One","Two"]  
<person>Two</person>
```

- Text in mixed content will be converted into a **#text** property

```
<p>This is an <b>example</b>!</p>    p": {"#text": "This is an",  
                                         "b": "example",  
                                         "#text1": "!"}
```

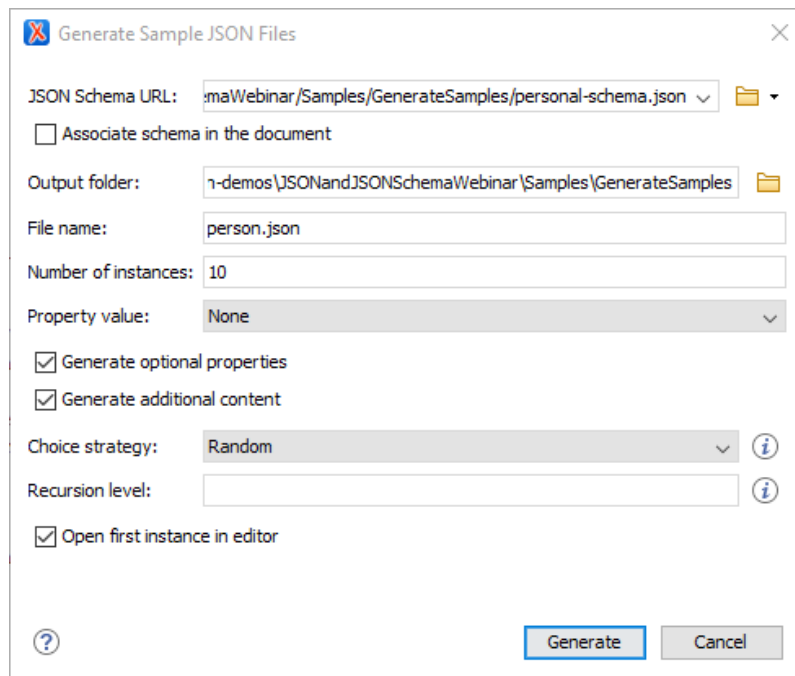
## XSD to JSON Schema converter

- Tool for converting an XML Schema file (XSD) to a JSON Schema



## Instance generator from a JSON Schema

- Tool for generating sample JSON files from a JSON Schema



The screenshot shows a dialog box titled "Generate Sample JSON Files" with the following fields and options:

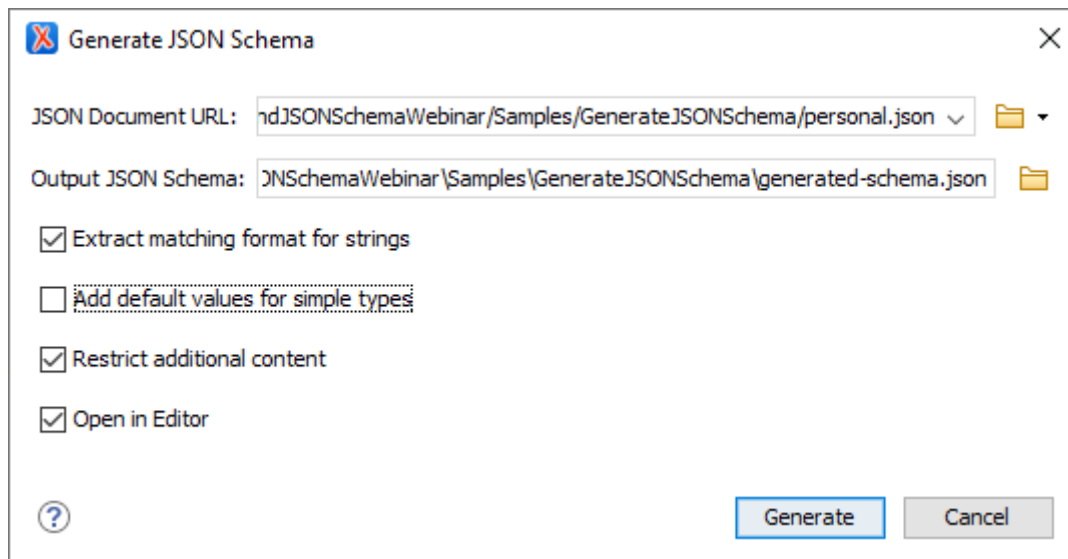
- JSON Schema URL: `maWebinar/Samples/GenerateSamples/personal-schema.json`
- Associate schema in the document
- Output folder: `n-demos\JSONandJSONSchemaWebinar\Samples\GenerateSamples`
- File name: `person.json`
- Number of instances: `10`
- Property value: `None`
- Generate optional properties
- Generate additional content
- Choice strategy: `Random`
- Recursion level: (empty)
- Open first instance in editor

Buttons: `Generate` and `Cancel`. A help icon (?) is located at the bottom left.



# Generate JSON Schema from JSON instance

- Tool for generating JSON Schema



## Conclusion

- JSON conforms to the ECMA/ISO specification
- Validate JSON with JSON Schema and Schematron
- Editing based on JSON Schema
- Query and process JSON using XPath, XSLT/XQuery
- Useful JSON Tools



## Future Plans

- JSON Schema **Diagram**
- **Visual editing** in Author mode for JSON
- **Quick fixes** for JSON problems
- **YAML** support



## Resources

- <https://www.json.org/>
- <https://json-schema.org/>
- [https://www.oxygenxml.com/xml\\_editor/json.html](https://www.oxygenxml.com/xml_editor/json.html)
- <https://www.oxygenxml.com/doc/ug-editor/topics/editing-JSON.html>





Questions?

Octavian Nadolu  
Software Architect at Syncro Soft

[octavian.nadolu@oxygenxml.com](mailto:octavian.nadolu@oxygenxml.com)

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

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