JSON: introduzione

Mario G.C.A. Cimino

a) JSON Example

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

Editor online che indenta JSON:
https://jsoneditoronline.org

b) XML Example:

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

c) XML Example (compact):

```
<employees>
  <employee firstName="John" lastName="Doe" />
  <employee firstName="Anna" lastName="Smith" />
  <employee firstName="Peter" lastName="Jones" />
</employees>
```

JSON, JSON-Schema: introduzione

JSON (JavaScript Object Notation) è un formato (alternativo ad XML) per strutturare oggetti, nella forma di coppie attributo-valore. Rispetto a XML e' piu' compatto, ha regole di parsing semplici.

JSON usato in scambio dati, e nell'archiviazione. Es. usato nel database "MongoDB", di tipo NoSQL, per archiviare documenti (dati semi strutturati).

Esiste anche **JSON schema** (https://json-schema.org) per la validazione dei documenti JSON

a) **Tipi** JSON

numero (es. "3.14"), stringa (es. "Antonio"), booleano ("true" o "false")

b) **Oggetto** (**Object**) JSON (elemento formato da proprietà eterogenee)

"*giocatore*": *{ "email": "mario.rossi@gmail.com", "punti": "340" }*

"nomeoggetto": { "nomeproprietà": "valoreproprietà", "nomeproprietà": "valoreproprietà", … }

c) **Vettore (Array)** JSON (formato da elementi con la medesima struttura)

"automobili":["Ford", "BMW", "Fiat"]

"nomevettore": ["primo elemento", "secondo elemento", "terzo elemento"]

JSON, JSON-Schema: introduzione

identificativo dello schema person

versione del vocabolario json-schema con cui si esprime lo schema

Schema di Person:

```
{
  "$id": "https://example.com/person.schema.json",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Person",
  "type": "object",
  "properties": {
    "firstName": {
      "type": "string",
      "description": "The person's first name."
    },
    "lastName": {
      "type": "string",
      "description": "The person's last name."
    },
    "age": {
      "description": "Age in years which must be equal to or greater than zero.",
      "type": "integer",
      "minimum": 0
    }
  }
}
```

Istanza di Person:

```
{
  "firstName": "John",
  "lastName": "Doe",
  "age": 21
}
```

JSON, JSON-Schema: introduzione

Schema di geographical-location:

```
{
  "$id": "https://example.com/geographical-location.schema.json",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Longitude and Latitude Values",
  "description": "A geographical coordinate.",
  "required": [ "latitude", "longitude" ],
  "type": "object",
  "properties": {
    "latitude": {
      "type": "number",
      "minimum": -90,
      "maximum": 90
    },
    "longitude": {
      "type": "number",
      "minimum": -180,
      "maximum": 180
    }
  }
}
```

Istanza di geographical-location:

```
{
  "latitude": 48.858093,
  "longitude": 2.294694
}
```

JSON, JSON-Schema: introduzione

Schema di fruit-and-vegetables:

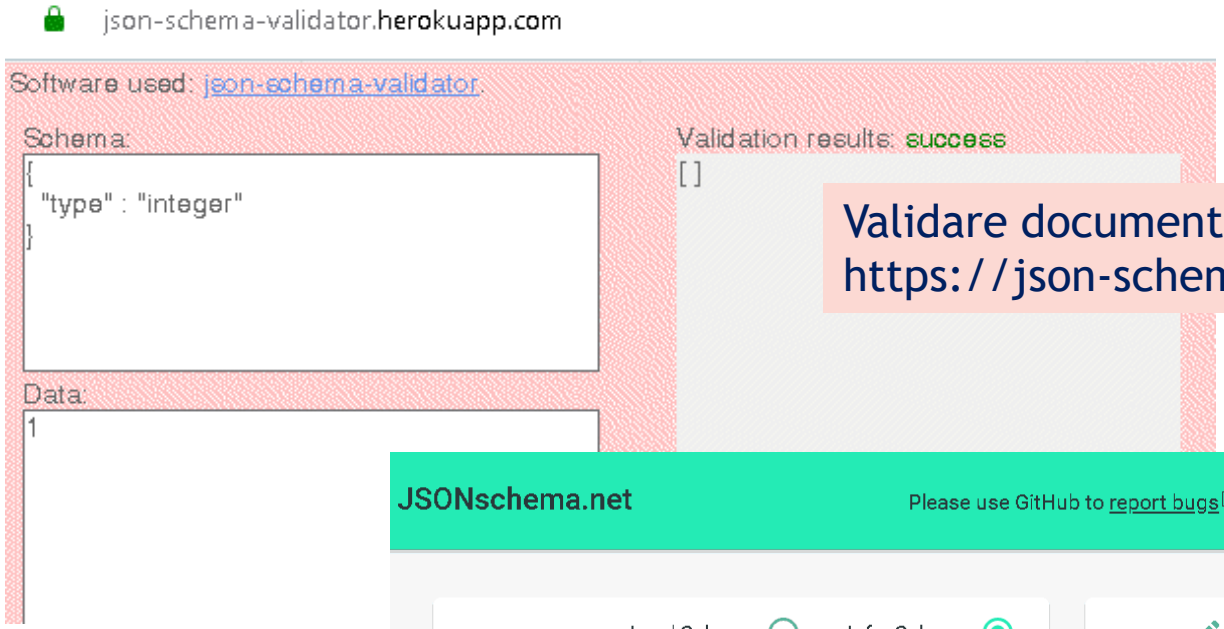

```
{
 "$id": "https://example.com/vegetables.schema.json",
 "$schema": "http://json-schema.org/draft-07/schema#",
 "description": "Fruits and vegetables of interest",
 "type": "object",
 "properties": {
  "fruits": {
   "type": "array",
   "items": {
    "type": "string"
   }
  },
  "vegetables": {
   "type": "array",
   "items": { "$ref": "#/definitions/veggie" }
  }
 },
 "definitions": {
  "veggie": {
   "type": "object",
   "required": [ "veggieName", "veggieLike" ],
   "properties": {
    "veggieName": {
     "type": "string",
     "description": "The name of the vegetable."
    },
    "veggieLike": {
     "type": "boolean",
     "description": "Do I like this vegetable?"
    }
   }
  }
 }
}
```

Istanza di fruit-and-vegetables:

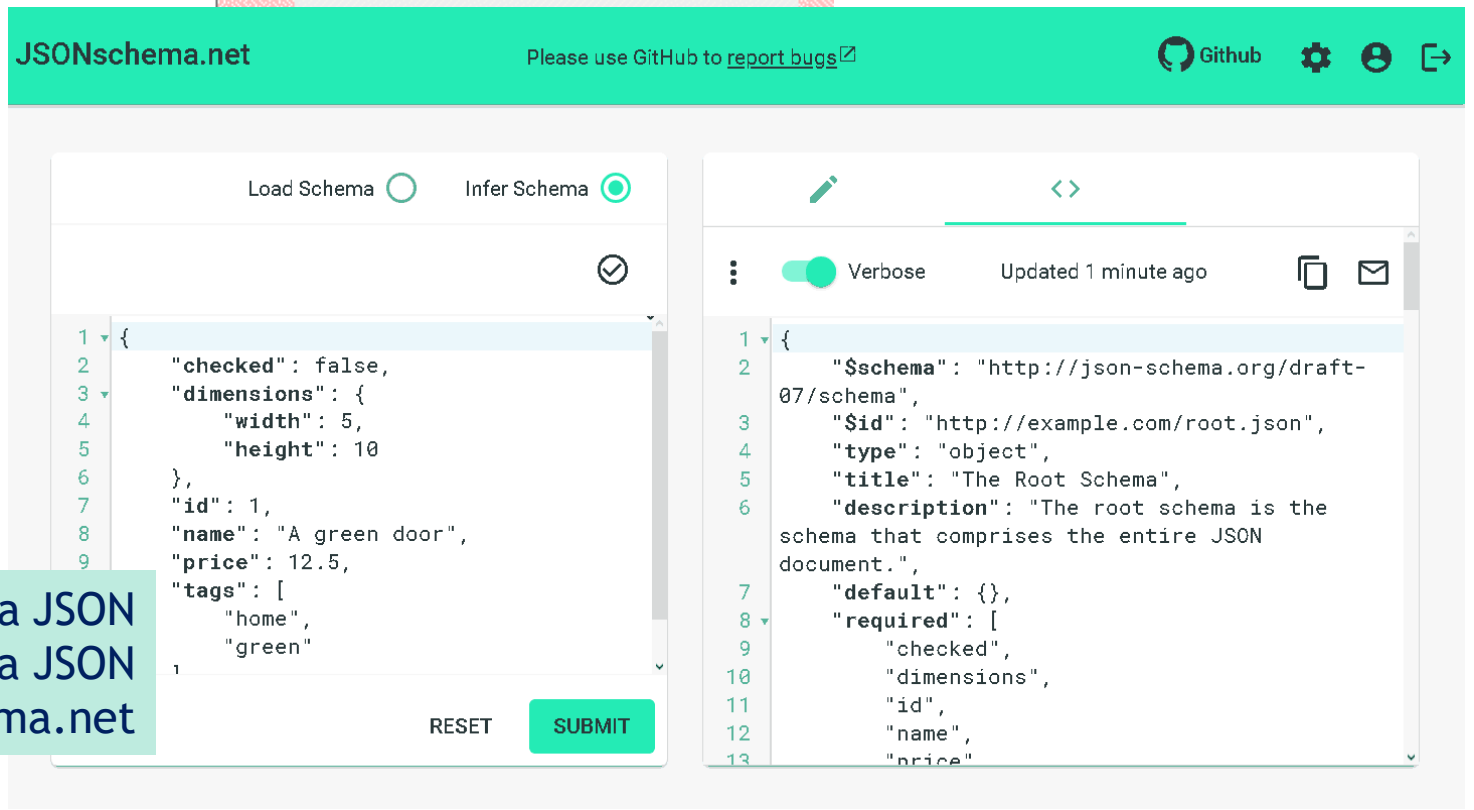
```
{
  "fruits": [ "apple", "orange", "pear" ],
  "vegetables": [
    {
      "veggieName": "potato",
      "veggieLike": true
    },
    {
      "veggieName": "broccoli",
      "veggieLike": false
    }
  ]
}
```

riferimento ad un sotto schema

JSON e JSON-Schema: introduzione



Validare documenti JSON
https://json-schema-validator.herokuapp.com



Inferire lo schema JSON
dall'istanza JSON
https://jsonschema.net