

JSON-VL

version 1.0

draft 2

May 15, 2008

Editors:

Claudio D'Angelo

Versions

<i>Version</i>	<i>Date</i>	<i>Description</i>
1.0 draft	May 12, 2008	Initial release
1.0 draft 2	May 15, 2008	Adds AnyValidator

Table of contents

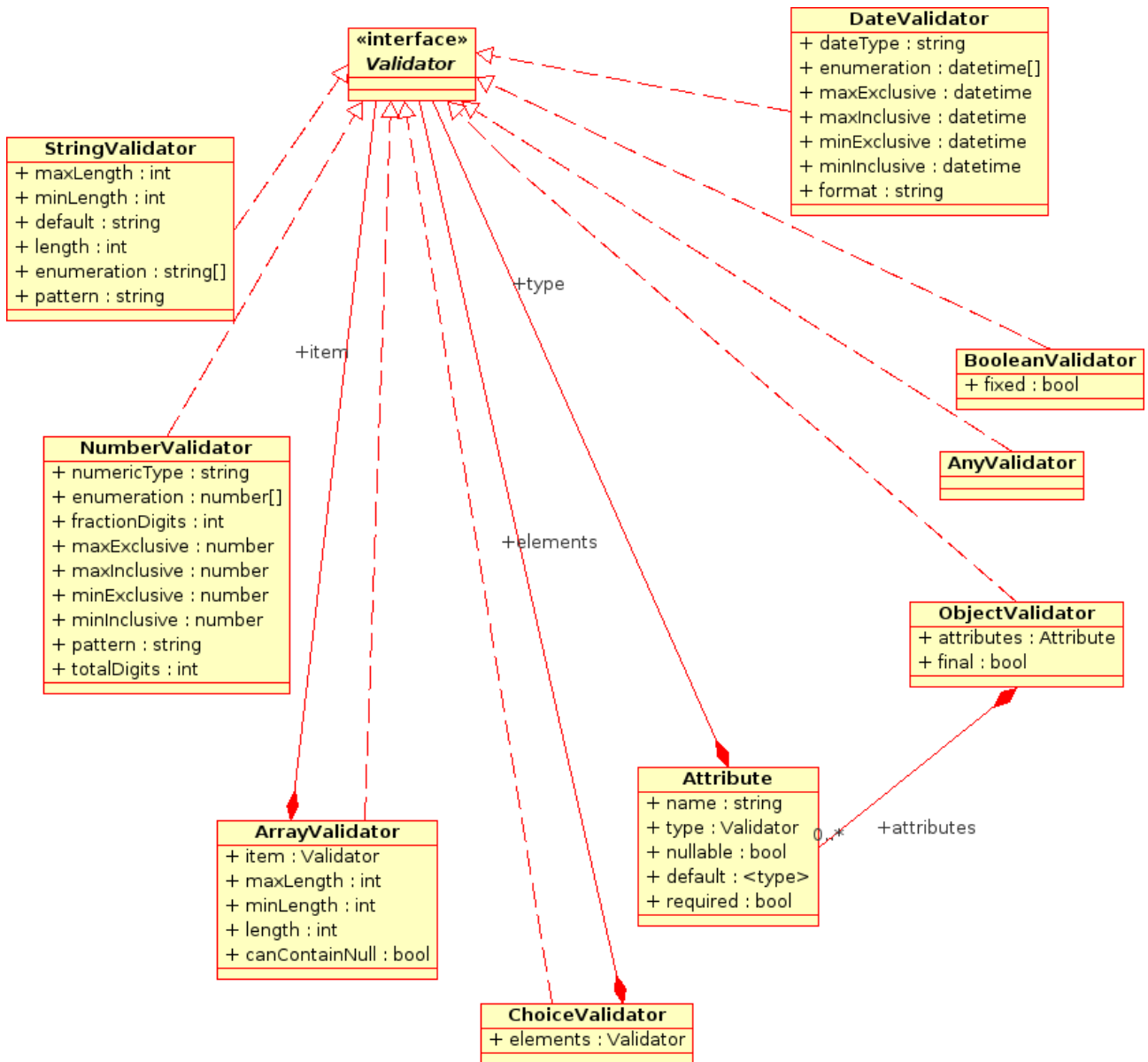
<u>Versions.....</u>	<u>2</u>
<u>JSON-VL.....</u>	<u>4</u>
<u>Validator.....</u>	<u>5</u>
<u>StringValidator.....</u>	<u>6</u>
<u>NumberValidator.....</u>	<u>7</u>
<u>NumericType.....</u>	<u>8</u>
<u>BooleanValidator.....</u>	<u>8</u>
<u>NullValidator.....</u>	<u>8</u>
<u>AnyValidator.....</u>	<u>9</u>
<u>DateValidator.....</u>	<u>9</u>
<u>dateType.....</u>	<u>10</u>
<u>format.....</u>	<u>10</u>
<u>ArrayValidator.....</u>	<u>11</u>
<u>ObjectValidator.....</u>	<u>11</u>
<u>Attribute.....</u>	<u>12</u>
<u>ChoiceValidator.....</u>	<u>13</u>
<u>Reference.....</u>	<u>13</u>

JSON-VL

JSON-VL (JSON-ValidationLanguage) is a porting for json of xml schema.

The first goal of JSON-VL is the description of a json structur and its validation. The language's structure is the same of json, it follows the json-schema specification but with the goal to use like a validation system. JSON-VL can be extended for any use using the annotation property where the developer can be adds any other informations used by other tools to generate gui, classes or other. A json-vl instance is formed by one or plus validator instance. Any validator instance represents a description of how must be validated a json element. There is a special validator called *choice* that is used to permit to an element to represents various types.

Below there is a class diagram that represents an idea of a json-vl instance:



Validator

Represents an abstract validator; in the language a validator is a json object that contains this properties:

Attribute name	JSON Type	Mandatory	Description
type	String	Y	Indicate the validator's type. Permitted values are: <ul style="list-style-type: none"> ● string ● number ● boolean ● object ● array

			<ul style="list-style-type: none"> ● choice ● date ● reference (used for validation reference)
id	String	N	A URI or URN used to identify in a unique manner a validator (like namespace in xml schema)
extends	String	N	Reference to a validation id. This attribute indicate that this validator extend the validator referenced by the attribute
extendsLocation	String	N	Indicate where can be loaded the validator referenced by the attribute extends. It must be a URI. Example: <pre>{ "type" : "object", "extends" : "http://myserver.com/validators/form", "extendsLocation" : "file:///home/my/form.json-vl", }</pre>
documentation	Any	N	This attribute is used to insert any documentation of the validator. The attribute's value can be any type (object, array, number, etc...)
Annotation	Array	N	This attribute contain a list of annotation used to extends the validation's information. The array's item must be an object with an id attribute that represents an URI or URN. Any value that isn't an object with a valid id is an invalid annotation.

StringValidator

The StringValidator is used to constraints a string value. StringValidator's attributes are:

Attribute name	JSON Type	Mandatory	Description
maxLength	number(integer)	N	Indicate the maximum length of the string
minLength	number(integer)	N	Indicate the minimum length of the string
length	number(integer)	N	Indicate the exact length tht the string must have.
default	String	N	The value that the string must have if is null in the json

			stream.
enumeration	Array<String>	N	Set of acceptable values. Can contain null.
pattern	String	N	Regular expression that indicate the pattern of acceptable values.

NumberValidator

The NumberValidator is used to constraints a string value. NumberValidator's attributes are:

Attribute name	JSON Type	Mandatory	Description
numericType	String	N	Indicate the format of the numeric value. Valid values are: <ul style="list-style-type: none"> ● long (default) ● byte ● decimal ● int ● integer ● negativeInteger ● nonNegativeInteger ● nonPositiveInteger ● positiveInteger ● short ● unsignedLong ● unsignedInt ● unsignedShort ● unsignedByte
maxExclusive	number	N	Indicate the maximum value of the number not inclusive
maxInclusive	number	N	Indicate the maximum or equals value of the number
minExclusive	number	N	Indicate the minimum value of the number not inclusive
minInclusive	number	N	Indicate the minimum or equals value of the number
totalDigits	number(integer)	N	Indicate the exact number of digits permitted in the number.
fractionDigits	number(integer)	N	Indicate the maximum number of digits permitted in the number.
default	number	N	The value that the number must have if is null in the json stream.
enumeration	Array<n	N	Set of acceptable values. Can contain null.

	umber>		
pattern	String	N	Regular expression that indicate the pattern of acceptable values.

NumericType

The attribute numericType indicate the format of the number. Below a table that specifies formats:

Name	Description
byte	A signed 8-bit integer
decimal	A decimal value
int	A signed 32-bit integer
integer	An integer value
long	A signed 64-bit integer
negativeInteger	An integer containing only negative values (.., -2, -1.)
nonNegativeInteger	An integer containing only non-negative values (0, 1, 2, ..)
nonPositiveInteger	An integer containing only non-positive values (.., -2, -1, 0)
positiveInteger	An integer containing only positive values (1, 2, ..)
short	A signed 16-bit integer
unsignedLong	An unsigned 64-bit integer
unsignedInt	An unsigned 32-bit integer
unsignedShort	An unsigned 16-bit integer
unsignedByte	An unsigned 8-bit integer

BooleanValidator

A simple validator to check a boolean value, attributes are:

Attribute name	JSON Type	Mandatory	Description
fixed	boolean	N	Indicate that value is permitted. Null is permitted.

NullValidator

A simple validator to check a boolean value (without attributes).

AnyValidator

This validator define that any type can be inserted. No check will be executed. If you want define constraints you can use the ChoiceValidator

DateValidator

Validator to check a date value. In JSON doesn't exist a date format and a string value must be used. DateValidator's attributes are:

Attribute name	JSON Type	Mandatory	Description
dateType	String	N	Indicate the format of the date value. Permitted values are: <ul style="list-style-type: none">● date● dateTime (default)● duration● gDay● gMonth● gMonthDay● gYear● gYearMonth● time● custom
enumeration	Array<String>	N	Set of permitted values. Can contain null. The values in the array must follow the format in the dateType attribute.
maxExclusive	String	N	Indicate the maximum value of the date not inclusive. The value in the array must follow the format in the dateType attribute.
maxInclusive	String	N	Indicate the maximum or equals value of the date. The value in the array must follow the format in the dateType attribute.
minExclusive	String	N	Indicate the minimum value of the date not inclusive. The value in the array must follow the format in the dateType attribute.
minInclusive	String	N	Indicate the minimum or equals value of the date. The value in the array must follow the format in the dateType attribute.
format	String	Y (only if dateType value is	Indicate the format of the date. This is used only if the dateType attribute is "custom"

		“custom”)	
--	--	-----------	--

dateType

Below a table with the specification of values:

Name	Description
date	Defines a date value in format YYYY-MM-DD
dateTime	Define a date and time value in format YYYY-MM-DD'T'HH:MIN:SS[Z]. The time zone is optional.
duration	<p>The duration data type is used to specify a time interval. The time interval is specified in the following form "PnYnMnDTnHnMnS" where:</p> <ul style="list-style-type: none"> • P indicates the period (required) • nY indicates the number of years • nM indicates the number of months • nD indicates the number of days • T indicates the start of a time section (required if you are going to specify hours, minutes, or seconds) • nH indicates the number of hours • nM indicates the number of minutes • nS indicates the number of seconds <p>For example: P5Y3M => specify a period of 5 years and 3 months.</p>
gDay	Defines a part of a date - the day (DD)
gMonth	Defines a part of a date - the month (MM)
gMonthDay	Defines a part of a date - the month and day (MM-DD)
gYear	Defines a part of a date - the year (YYYY)
gYearMonth	Defines a part of a date - the year and month (YYYY-MM)
time	Defines a time value (HH:MIN:SS)

format

The format attribute define the date's format used in the custom dataType:

Letter	Date or Time Component	Presentation	Examples
G	Era designator	<u>Text</u>	AD

y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

ArrayValidator

The ArrayValidator defines constraints for an array value. Permitted attributes are:

Attribute name	JSON Type	Mandatory	Description
item	Object	Y	Define what type of item can be contain. The object is a validator instance or a reference instance.
maxLength	number(integer)	N	Define a maximum length of the array
minLength	number(integer)	N	Define a minimum length of array
length	number(integer)	N	Define the exact length of the array
canContainsNull	boolean	N	Define if the array can contain null values.

ObjectValidator

The ObjectValidator defines constraints for an object value. Permitted attributes are:

Attribute name	JSON Type	Mandatory	Description
attributes	Object	N	Defines a map of attributes that specifies the type/constraints. The key define the attribute name of the object, the value define an attribute that is a validator extension (see below)
final	boolean	N	Defines that this object is final and can't be extended

Attribute

The attributes map is a json object which attribute names are the object attribute name which value is an attribute instance.

The attribute instance is an extension of a valid validator and contains (apart from validator instance attribute) this attributes :

Attribute name	JSON Type	Mandatory	Description
@required	boolean	N	Define that this attribute must be present in the json object. Default false
@nullable	boolean	N	Define that this attribute can be null. Default true
@default	Any(must be the same type of validator)	N	Define the default value if is null or missed in json object. Overwrite validator's default

The '@' is for @ttribute and it identifies attributes of attribute instance.

Examples:

```
{ "type" : "object",
  "attributes" : {
    "name" : {
      "type" : "string",
      "minLength" : 3,
      "@nullable" : false,
      "@required" : true
    }
  }
}
```

```

    },
    "surname" : {
        "type" : "string",
        "@default" : "Unkown",
        "@required" : true
    }
}

```

This validator define an objet with 2 attributes "name" and "surname"

ChoiceValidator

ChoiceValidator is a special validator that permit of use a choice between validators (with the same type too).

The use of ChoiceValidator is expensive for a validator, use it cheaply.

Attribute name	JSON Type	Mandatory	Description
elements	Array<Object>	Y	Define the list of permitted element. Any object is a validator's instance.

Reference

Reference is a special object/validator used to reference an external or internal validator.

Attribute name	JSON Type	Mandatory	Description
ref	string	Y	Define the id of the validator that must be referenced. If no id are registered in the parser, the parsing must stop with an error.
location	string	N	Define the location where the validator can be found. Location identifies a URI of a json stream that will be parsed and registered in the parser. If location is invalid the parser must stop the parsing with an error.

Examples:

```

{ //in myvalidator.json
  "id" : "http://myserver.com/person",

```

```
“type” : “object”,
“attributes” : {
    “name” : {“type” : “string”},
    “surname” : {“type” : “string”},
    “street” : {
        “id” : “http://myserver.com/street”,
        “attributes” : {
            ...
        }
    }
}
```

//reference example

```
{
“type” : “object”,
“attributes” : {
    “home” : {
        “type” : “reference”,
        “ref” : “http://myserver.com/street”,
        “location” : “file:///home/test/myvalidator.json”
    },
    “client” : {
        “type” : “reference”,
        “ref” : “http://myserver.com/person”
    }
}
}
```