

An Introduction to SHACL

Tool Name	SHACL
Responsible Person	Maharshi Sen
Link to Tool /Relevant Links	https://www.w3.org/TR/shacl/

Tool Description

SHACL(Shapes Constraint Language) is a tool which is used to validate an RDF graph against a given set of conditions. RDF graphs that are used in this manner are called "shapes graphs" in SHACL and the RDF graphs that are validated against a shapes graph are called "data graphs".

Goal/Purpose

SHACL can be used to validate if an RDF graph conforms to a given format: it can be used to validate whether values (objects in RDF triples) of properties(predicates in RDF triples) are valid according to a given set of rules, such as:

- The language the data is supposed to be in:

```
ex:NewZealandLanguagesShape
  a sh:NodeShape ;
  sh:targetNode ex:Mountain, ex:Berg ;
  sh:property [
    sh:path ex:prefLabel ;
    sh:languageIn ( "en" "mi" ) ;
  ] .
```

- The number of values the property can take:

```
ex:SuperShape
  a sh:NodeShape ;
  sh:property [
    sh:path ex:property ;
    sh:minCount 1 ;
  ] .
```

- The datatype of the values:

```
ex:PersonShape
  a sh:NodeShape ;
  sh:targetClass ex:Person ;
  sh:property [ sh:path ex:name ; sh:datatype xsd:string ; sh:minLength 1 ; ] ;
```

- patterns the values must conform to(regular expressions) :

```
:ProductShape a sh:NodeShape ; sh:targetClass :Product ; sh:property [ sh:path schema:productID ; sh:
pattern "^P\\d{3,4}" ; sh:flags "i" ; ] .
```

A practical usage of SHACL can be imagined as follows: After extracting relevant triples from an Organization Knowledge Graph , containing triples about different organizations, their suppliers, their products and applications, and their customers. Through the usage of Python, one can further filter out different kinds of data, such as Organizations, Products, Applications etc. based on the pattern of the data entries(e.g. datat type, starting alphabet etc.).

Tool Input

Tool Output

The tool requirements:

- The tool was tested in VS Code with the extensions 'Stardog RDF Grammars' and 'SHACL Language Server' present.
- An RDF graph to be validated, called the "data graph", which should be completely free from any errors such as typos, the missing semicolons or full-stops, and the usage of special characters(-,?,@etc.) in the URLs for subjects, predicates and objects, after the prefixes. Only '_' is acceptable. For example: This graph: ex: Alice a ex:Person; : livesAt :North-Nuremberg. will not be parsed correctly due to presence of "-" in North-Nuremberg. Use : North_Nuremberg instead.
- The SHACL graph, also called "shapes graph", against which the Data Graph will be validated against, with all the shapes and their properties properly defined with correct syntax.

Open the data graph in VS Code. Go to View Command Palette Linked Data: Validate.

Browse on your system for the shapes graph and select it.

The data graph is validated against the SHACL graph, and according to whether the data graph conforms to the shapes in the SHACL graph.

If it conforms completely, the output will be 'Congratulations! Your file is valid!' accompanied by the number of lines parsed in the data graph.

```
#shapes: my_excellent_shapes.ttl
@prefix ex: <http://example.com/>.
@prefix xsd:<http://www.w3.org/2001/XMLSchema>.
ex:Alice a ex:Person;
ex:name "A"^^xsd:string,"Alicia"^^xsd:string;
ex:age "21"^^xsd:integer;
ex:worksAt "Fraunhofer"@de.

ex:Bob a ex:Person;
ex:name "Bob"^^xsd:string,"B"^^xsd:string,"Porcupine"^^xsd:string;
ex:age "25"^^xsd:integer;
ex:worksAt "Siemens"@de.

ex:Alice ex:Gender "Female"^^xsd:string.
```

🔔 Congratulations, your file is valid!

🔔 Successfully parsed: Statements in the graph: 11

🔔 Successfully parsed: Statements in the graph: 12

If not, it will be give the output - 'Alas, your file is not valid.' It will show the errors in a tabular format in a window and the number of lines successfully parsed in the graph.

```
C:\Users> mahar > OneDrive > Desktop > SHACL_examples > trial2.ttl
1 #shapes: my_excellent_shapes.ttl
2 @prefix ex: <http://example.com/>.
3 @prefix xsd:<http://www.w3.org/2001/XMLSchema>.
4 ex:Alice a ex:Person;
5 ex:name ""^^xsd:string;
6 ex:age "21"^^xsd:integer;
7 ex:worksAt "Fraunhofer"@de.
8
9 ex:Bob a ex:Person;
10 ex:name "Bob"^^xsd:string,"B"^^xsd:string,"Porcupine"^^xsd:string;
11 ex:age "25"^^xsd:integer;
12 ex:worksAt "Siemens"@de.
13
14 ex:Alice ex:Gender "Female"^^xsd:string.
```

Type a keyword...

node	mes...	shape	pre
http://example.com/Alice	Value has less than 1 characters	http://example.com/PersonShape	http...

Showing 1 to 1 of 1 results Previous 1 Next

⚠️ Alas, your file is not valid.

Related Projects

Project

Notes/Comments

https://owncloud.fraunhofer.de/index.php/apps/files/?dir=/FE_Collaboration/Module/Ontologies/SHACL&fileid=713631907	<p>My example demonstrates a simple data graph, and a SHACL graph, the validation process, and how can we convert data from a knowledge graph to a dataframe through Python.</p> <ul style="list-style-type: none"> • The file 'Data_Graph' contains the knowledge graph to be parsed, • The file 'Shapes_Graph' is the SHACL graph, • The file 'Data Processing' shows the process of conversion of data from a knowledge graph to a dataframe.

Example

RDF Graph

```
#shapes: my_excellent_shapes.ttl
#description of a person Alice having names A and Alicia, has age 21 and works at Fraunhofer.
@prefix ex: <http://example.com/>.
@prefix xsd:<http://www.w3.org/2001/XMLSchema#>.
ex:Alice a ex:Person;
ex:name "A"^^xsd:string,"Alicia"^^xsd:string;
ex:age "21"^^xsd:integer;
ex:WorksAt "Fraunhofer"@de.
#description of a person Bob having B and Porcupine, has age 25 and works at Siemens.
ex:Bob a ex:Person;
ex:name "Bob"^^xsd:string,"B"^^xsd:string,"Porcupine"^^xsd:string;
ex:age "25"^^xsd:integer;
ex:WorksAt "Siemens"@de.
#Alice also is a woman.
ex:Alice ex:Gender "Female"^^xsd:string.
```

SHACL Graph

```
@prefix ex: <http://example.com/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd:<http://www.w3.org/2001/XMLSchema#>.

ex:PersonShape
  a sh:NodeShape ;
  sh:targetClass ex:Person ; #PersonShape is a Node belonging to class Person
  sh:property [
    sh:path ex:name ;          sh:datatype xsd:string ;          sh:minLength 1 ;          ] ;
  #the property 'name' of a PersonShape node is of datatype 'string', and has a minimum length of 1.
  sh:property [
    sh:path ex:age ;          sh:datatype xsd:integer ;          sh:minInclusive 0
    ;          sh:maxInclusive 150 ;          ] .#the property 'age' of a PersonShape node is of datatype integer,
and the value of age ranges as 0<=age<=150.
```

[Maharshi Sen](#) please add the command you run in VS Code or a screenshot demonstrating where you activate the extension with a brief description.

Comments