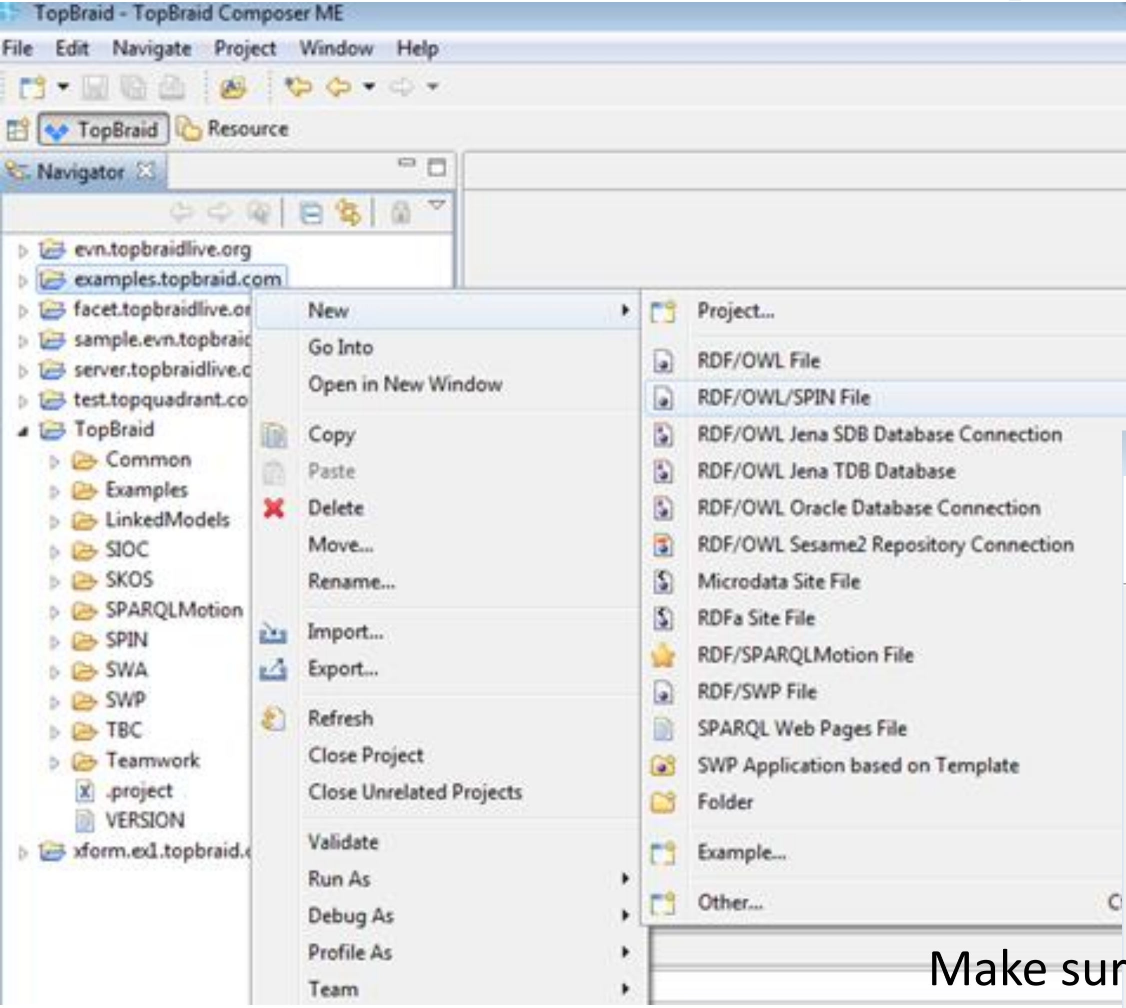
SPIN Templates in 5 Easy Steps

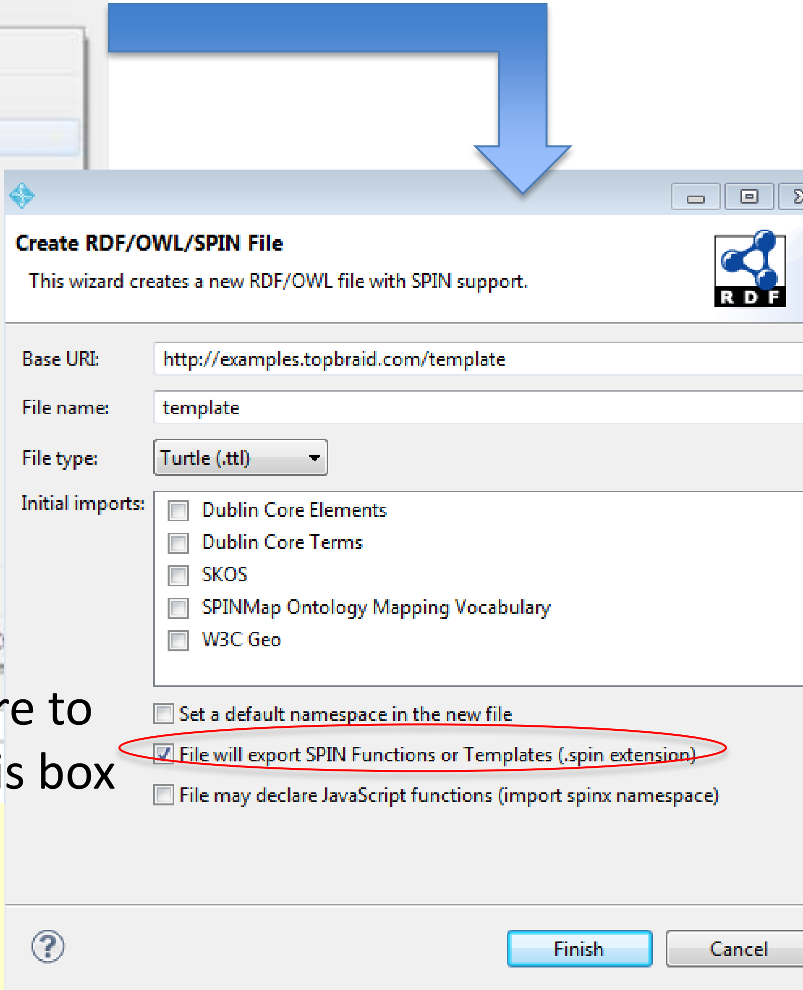
April, 2013

What is SPIN Template?

- ❑ A RESTFul Web Service – provided by TopBraid Template Servlet
  - ❖ Results returned can be in JSON, XML, etc.
- ❑ A parameterized SPARQL query
  - ❖ Arguments (parameters/variables) can be passed to the service
  - ❖ The number of arguments is unlimited and it is also possible to have a query that requires no input parameters

1. Create a SPIN file that will hold your templates





Create RDF/OWL/SPIN File

This wizard creates a new RDF/OWL file with SPIN support.

Base URI: http://examples.topbraid.com/template

File name: template

File type: Turtle (.ttl)

Initial imports:
- □ Dublin Core Elements
- □ Dublin Core Terms
- □ SKOS
- □ SPINMap Ontology Mapping Vocabulary
- □ W3C Geo

□ Set a default namespace in the new file

☑ File will export SPIN Functions or Templates (.spin extension)

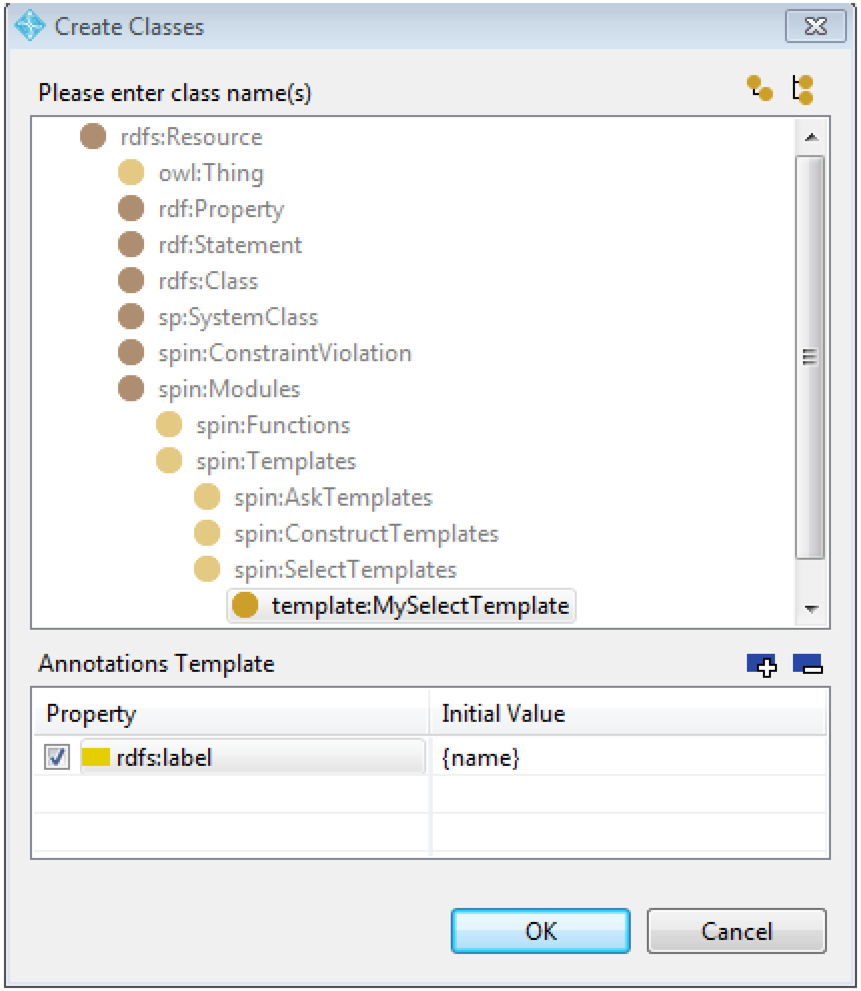
□ File may declare JavaScript functions (import spinx namespace)

Make sure to check this box

You could also add templates directly into your existing model as long as it imports spin.ttl graph. But in order to use templates as services, your file must have .spin extension.

2. Create a Template

- ❑ Expand the class tree to see spin:Templates class.
- ❑ Create a new subclass under the parent that corresponds to the type of the Template you are building.
- ❑ Here, we are creating a new Select Template and calling it ‘**MySelectTemplate**’.



Create Classes

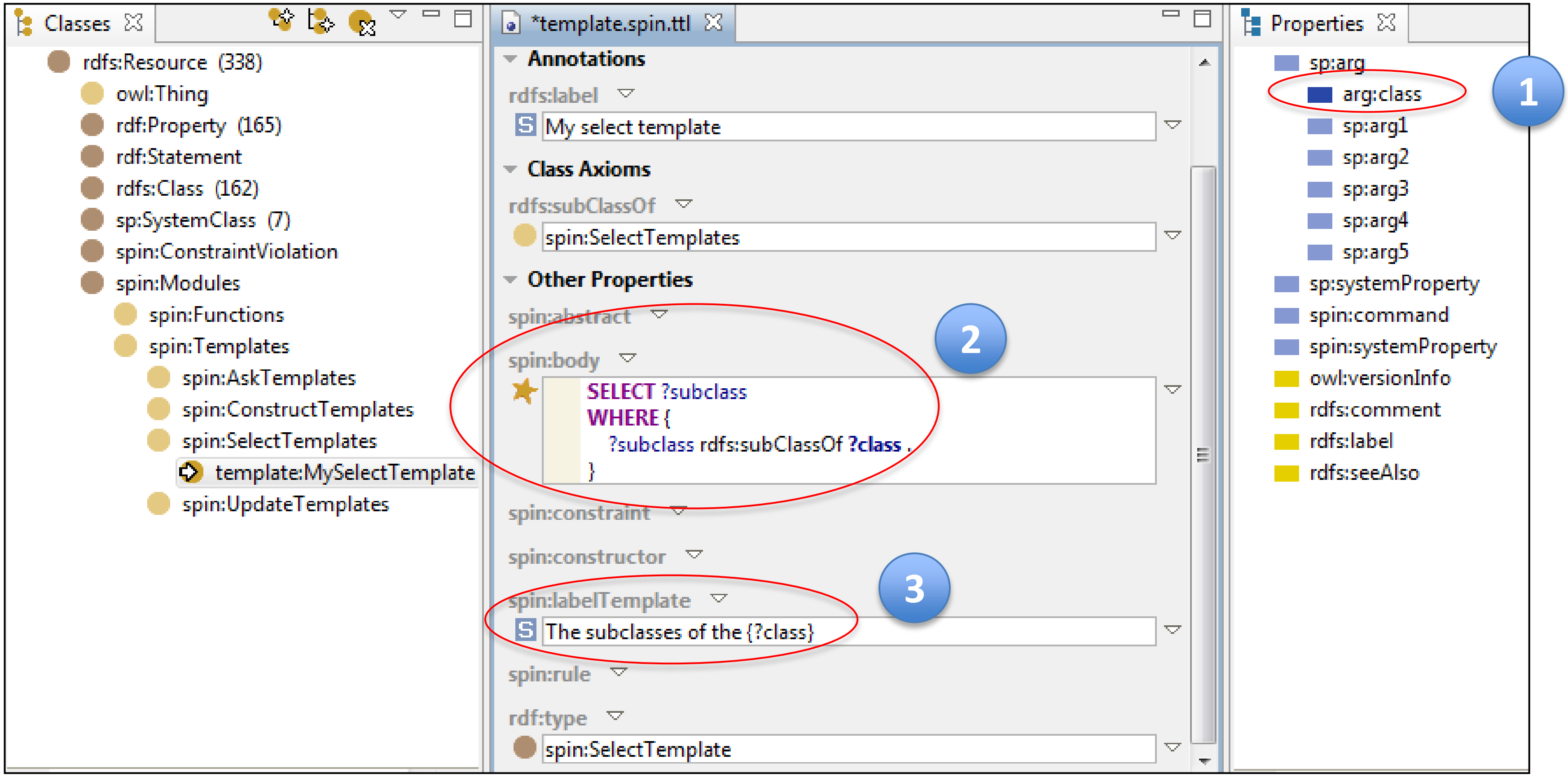
Please enter class name(s)

- rdfs:Resource
  - owl:Thing
  - rdf:Property
  - rdf:Statement
  - rdfs:Class
  - sp:SystemClass
  - spin:ConstraintViolation
  - spin:Modules
    - spin:Functions
    - spin:Templates
      - spin:AskTemplates
      - spin:ConstructTemplates
      - spin:SelectTemplates
        - template:MySelectTemplate

Annotations Template

Property	Initial Value
☑ rdfs:label	{name}

3. Define a Query



rdfs:label: My select template

rdfs:subClassOf: spin:SelectTemplates

spin:body:

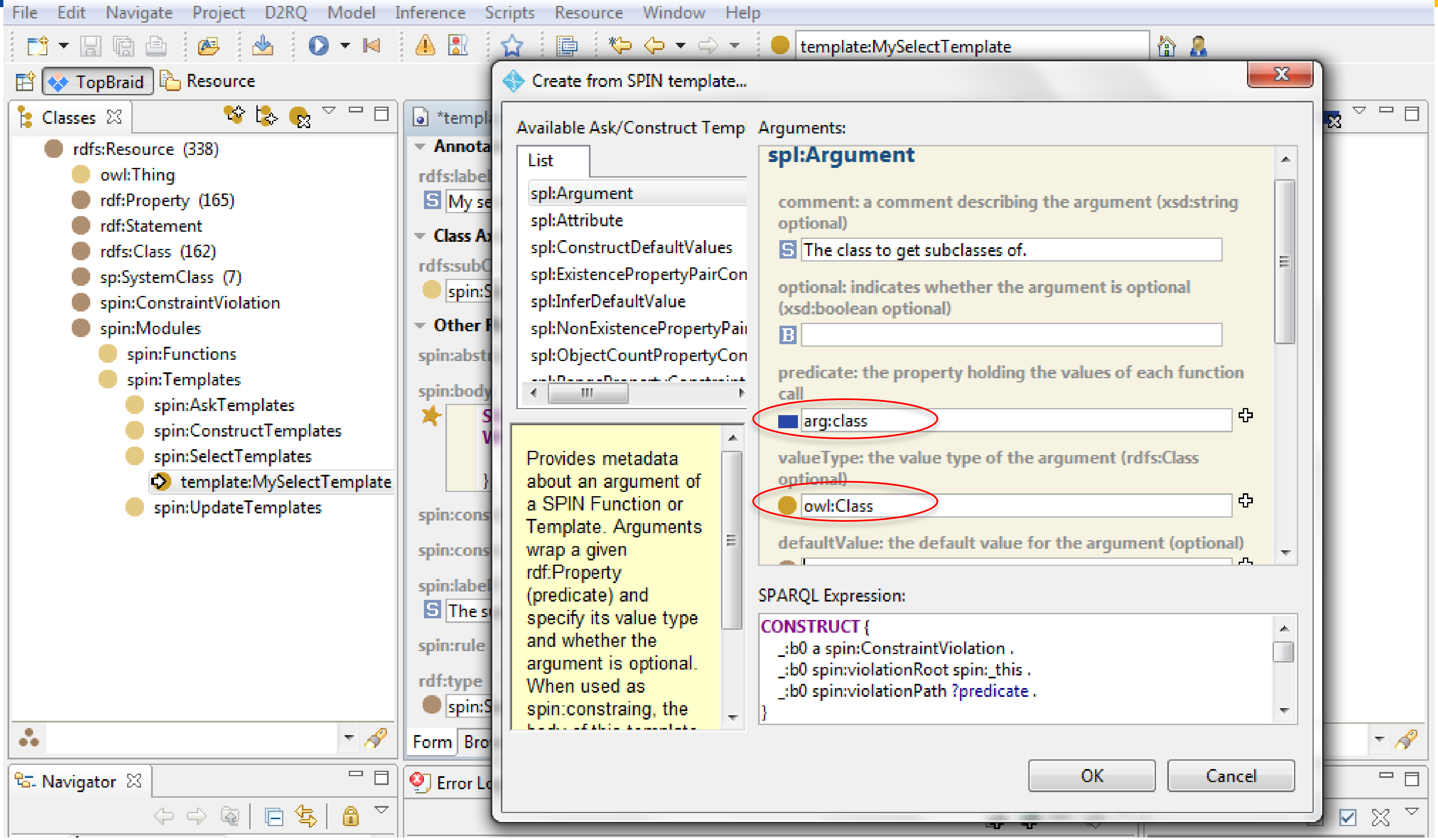
```
SELECT ?subclass
WHERE {
   ?subclass rdfs:subClassOf ?class .
}
```

spin:labelTemplate: The subclasses of the {?class}

rdf:type: spin:SelectTemplate

1. Create new property(s) to use as argument(s) for your service(s). Here, we created a new property **arg:class.**
2. Navigate to the template class you've created and enter your query in the spin:body field. Use your argument(s) by replacing the prefix with '?' e.g., **?class.**
3. Use spin:labelTemplate field to provide a text description.

4. Specify Argument(s)



Provides metadata about an argument of a SPIN Function or Template. Arguments wrap a given rdf:Property (predicate) and specify its value type and whether the argument is optional. When used as spin:constraing, the

spl:Argument

comment: a comment describing the argument (xsd:string optional)

S The class to get subclasses of.

optional: indicates whether the argument is optional (xsd:boolean optional)

B

predicate: the property holding the values of each function call

arg:class

valueType: the value type of the argument (rdfs:Class optional)

owl:Class

defaultValue: the default value for the argument (optional)

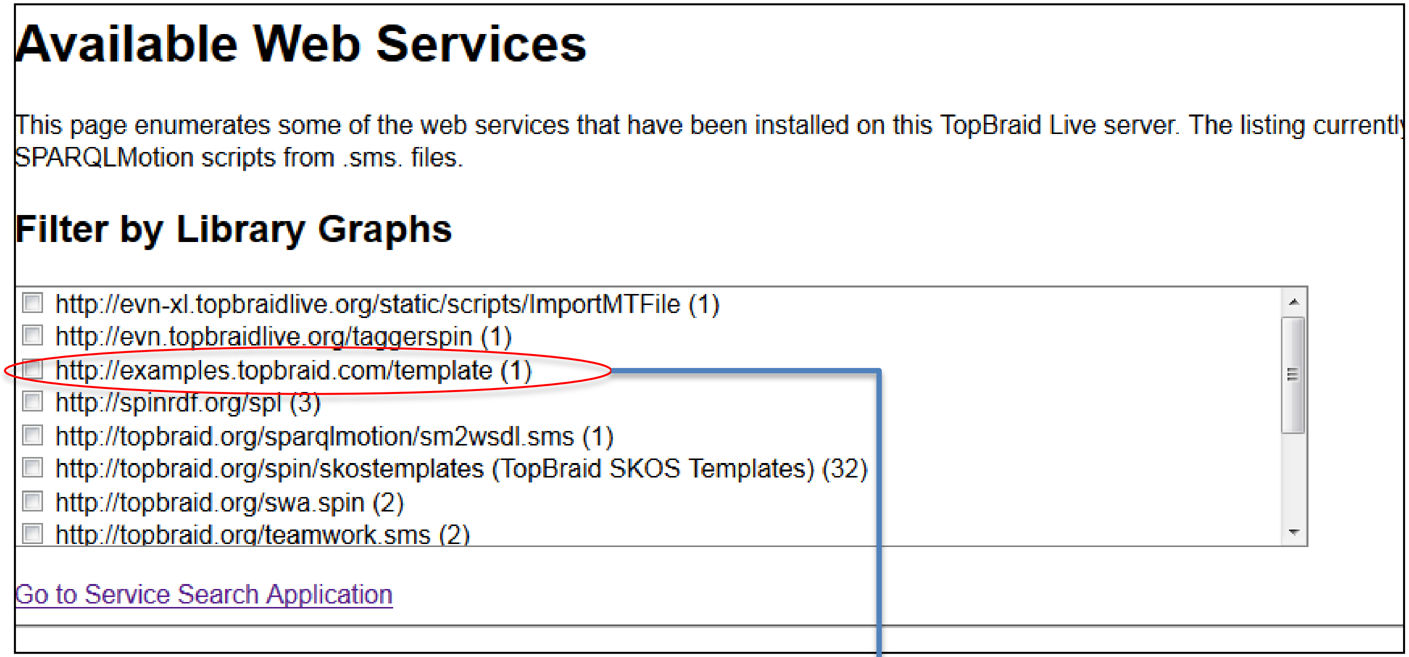
SPARQL Expression:

```
CONSTRUCT {
    _:b0 a spin:ConstraintViolation .
    _:b0 spin:violationRoot spin:_this .
    _:b0 spin:violationPath ?predicate .
}
```

- ❑ In the menu for spin:constraint field, select **Create from SPIN template …**
- ❑ Specify argument name in the predicate field. Optionally, provide the value type (in this case, any OWL class) for the argument. Provide a descriptive comment.
- ❑ Repeat, as needed, for additional arguments.

View Description of your new service

- ❑ Save your work, then go to http://localhost:8083/tbl/admin/ and select **Available Web Services.**



Available Web Services

This page enumerates some of the web services that have been installed on this TopBraid Live server. The listing currently

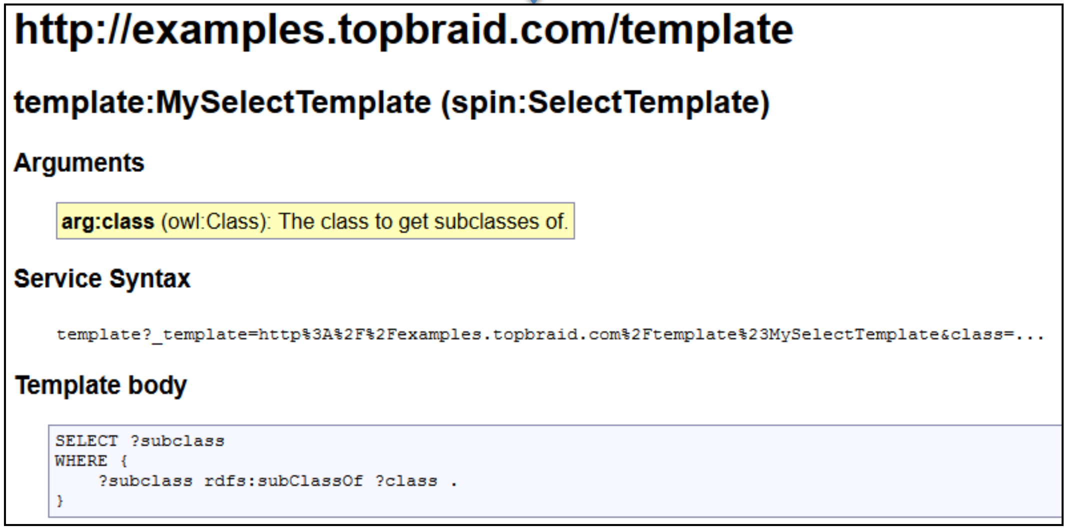
SPARQLMotion scripts from .sms. files.

Filter by Library Graphs

- □ http://evn-xl.topbraidlive.org/static/scripts/ImportMTFile (1)
- □ http://evn.topbraidlive.org/taggerspin (1)
- □ http://examples.topbraid.com/template (1)
- □ http://spinrdf.org/spl (3)
- □ http://topbraid.org/sparqlmotion/sm2wsdl.sms (1)
- □ http://topbraid.org/spin/skostemplates (TopBraid SKOS Templates) (32)
- □ http://topbraid.org/swa.spin (2)
- □ http://topbraid.org/teamwork.sms (2)

Go to Service Search Application

- ❑ You will se a list of graphs that contain web services (SPIN templates, SPARQMotion or SWP services). You will see your new SPIN file among them.
- ❑ Click to select your graph to see documentation of the service(s) it contains.



http://examples.topbraid.com/template

template:MySelectTemplate (spin:SelectTemplate)

Arguments

arg:class (owl:Class): The class to get subclasses of.

Service Syntax

```
template?_template=http%3A%2F%2Fexamples.topbraid.com%2Ftemplate%23MySelectTemplate&class=...
```

Template body

```
SELECT ?subclass
WHERE {
    ?subclass rdfs:subClassOf ?class .
}
```

5. Run your new Template (Service)

❑ To execute the template, use the service syntax provided in the service documentation:

- ❖Prefacing it with the server address. In our case, localhost
- ❖Providing required argument(s). In our example, we will use owl:Thing as the argument

Server address

http://localhost:8083/tbl/template?_template=http://examples.topbraid.com/template%23MySelectTemplate&
class=http://www.w3.org/owl%23Thing&_base=

Argument

We could also use qname owl:Thing as an argument, instead of the URI:

http://localhost:8083/tbl/template?_template=http://examples.topbraid.com/template%23MySelectTemplate&class=owl:Thing

Additional considerations

- Look at TopBraid Composer Help Page found in TopBraid Composer > TopBraid Live Integration > **SPIN Template Service** for additional arguments that can be passed to the service.
  - For example, how to specify the output format.
- **IMPORTANT:**
  - When running on local host in TBC-ME, template will be executed against the currently selected graph
  - When running on the server, it is important that the web services call specifies what graph to run the service against. This is done using **_base** argument.