

LexEVS 6.0 CTS2 Query 5 - Value Set Query Operation API

Contents of this Page

- [Introduction](#)
- [Interface](#)
- [Query Functions](#)
 - [listValueSets](#)
 - [listAllValueSets](#)
 - [getValueSetDetails](#)
 - [listValueSetContents](#)
 - [checkValueSetSubsumption](#)
 - [checkConceptValueSetMembership](#)
 - [listValueSetsWithConceptCode](#)

CTS2 Links for LexEVS 6.0

- [CTS2 API Main Page](#)
- [Programmer's Guide Main Page](#)
- [LexEVS 6.0 Main Page](#)
- [LexEVS Current Release](#)

Introduction

LexEVS CTS2 Value Set Query API provides capability to query Value Sets available in the system and also to query the binding between Value Set and Concept Domain.

Interface

org.lexevs.cts2.query.ValueSetQueryOperation is the main interface for all the queries against Value Set. This interface can be accessed using main LexEVSCTS2 interface, like:

```
org.lexevs.cts2.query.ValueSetQueryOperation vsQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().
getQueryOperation().getValueSetQueryOperation();
```

Query Functions

Here are the major query functions available using ValueSetQueryOperation interface:

listValueSets

This function returns value sets that are available in the system. Optionally, the list can be filtered by providing the criteria as input parameters.

listValueSets(String codeSystemId, String conceptDomainId, String usageContextId, String codeSystemURI, SortOption sortOption)

Description:	Lists the value sets that are available to the CTS 2 service.
Input:	• java.lang.String codeSystemId - (Optional) code system id the value set should reference. • java.lang.String conceptDomainId - (Optional) conceptDomain the value set should be bound to. • java.lang.String usageContextId - (Optional) usage context in which the value set can be used. • java.lang.String codeSystemURI - (Optional) URI of code system the codeSystemId or conceptDomainId or usageContextId belongs to. • org.LexGrid.LexBIG.DataModel.InterfaceElements.SortOption sortOption - (Optional) to sort the return list of value set identifiers in Ascending or Descending.
Output:	java.util.List<java.lang.String> - List of value set identifiers
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	Step 1: Instantiate ValueSetQueryOperation if it is not done yet:
	<pre>org.lexevs.cts2.query.ValueSetQueryOperation vsQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getValueSetQueryOperation();</pre>
	Step 2: Set SortOption to sort the list in ascending order:
	<pre>SortOption sortOption = new SortOption(); sortOption.setAscending(true);</pre>
	Step 3: Call listValueSets method by passing any filter criteria:
	<pre>java.util.List<java.lang.String> valueSetList = vsQuery.listValueSets("Automobiles", "Autos", null, null, sortOption);</pre>

listAllValueSets

This function returns all the value sets that are available in the system.

listAllValueSets(SortOption sortOption)

Description:	Lists all the value sets that are available to the CTS 2 service.
Input:	<i>org.LexGrid.LexBIG.DataModel.InterfaceElements.SortOption sortOption</i> - (Optional) to sort the return list of value set identifiers in Ascending or Descending.
Output:	<i>java.util.List<java.lang.String></i> - List of value set identifiers
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Sample Call:	Step 1: Instantiate ValueSetQueryOperation if it is not done yet:
	<pre>org.lexevs.cts2.query.ValueSetQueryOperation vsQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getValueSetQueryOperation();</pre>
	Step 2: Set SortOption to sort the list in ascending order:
	<pre>SortOption sortOption = new SortOption(); sortOption.setAscending(true);</pre>
	Step 3: Call listAllValueSets method to get all the value sets loaded in the system:
	<pre>java.util.List<java.lang.String> valueSetList = vsQuery.listAllValueSets(sortOption);</pre>

getValueSetDetails

This function returns detailed information about the value set.

getValueSetDetails(String valueSetId, String valueSetVersion)

Description:	Returns detailed information about the value set.
--------------	---

Input:	• <i>java.lang.String valueSetId</i> - (Mandatory) Id of an value set. • <i>java.lang.String valueSetVersion</i> - (Optional) Version of an value set.
Output:	<i>org.LexGrid.valueSets.ValueSetDefinition</i> - Value Set Definition object with the details of value set
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Sample Call:	• <i>Step 1:</i> Instantiate ValueSetQueryOperation if it is not done yet: <pre>org.lexevs.cts2.query.ValueSetQueryOperation vsQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getValueSetQueryOperation();</pre> • <i>Step 2:</i> Call getValueSetDetails method by providing value set id: <pre>org.LexGrid.valueSets.ValueSetDefinition vsd = vsQuery.getValueSetDetails("SRITEST:AUTO:AllDomesticButGM", null);</pre>

listValueSetContents

This function returns the contents of value set (expanded value set).

`listValueSetContents(String valueSetId, String valueSetVersion, AbsoluteCodingSchemeVersionReferenceList csVersionList, String versionTag, SortOption sortOption)`

Description:	Returns the contents of value set (expanded value set).
Input:	• <i>java.lang.String valueSetId</i> - (Mandatory) Id of an value set. • <i>java.lang.String valueSetVersion</i> - (Optional) Version of an value set. • <i>jorg.LexGrid.LexBIG.DataModel.Collections.AbsoluteCodingSchemeVersionReferenceList csVersionList</i> - (Optional) Code System Version reference list to be used to resolve value set. • <i>java.lang.String versionTag</i> - (Optional) The tag (e.g. "devel", "production", ...) to be used to determine which code system to be used. • <i>org.LexGrid.LexBIG.DataModel.InterfaceElements.SortOption sortOption</i> - (Optional) to sort the contents of value set in Ascending or Descending.
Output:	<i>org.lexgrid.valuesets.dto.ResolvedValueSetDefinition</i> - A resolved Value Set definition containing the code system version reference list that was used to resolve the value set and an iterator for resolved concepts.
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	Step 1: Instantiate ValueSetQueryOperation if it is not done yet:
	<pre>org.lexevs.cts2.query.ValueSetQueryOperation vsQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getValueSetQueryOperation();</pre>
	Step 2: Populate the Code System Version reference list to be used for resolving the value set definition:
	<pre>AbsoluteCodingSchemeVersionReference acsvr = new AbsoluteCodingSchemeVersionReference(); acsvr.setCodingSchemeURN("urn:oid:11.11.0.1"); acsvr.setCodingSchemeVersion("1.0"); AbsoluteCodingSchemeVersionReferenceList csList = new AbsoluteCodingSchemeVersionReferenceList(); csList.addAbsoluteCodingSchemeVersionReference(acsvr);</pre>
	Step 3: Call listValueSetContents method by providing value set id and code system version list:
	<pre>ResolvedValueSetDefinition vsdResolved = vsQueryop.listValueSetContents("SRITEST:AUTO:DomasticLeafOnly", null, csList, null, null); </pre>

checkValueSetSubsumption

This function determines whether one of the two supplied value sets subsumes the other.

checkValueSetSubsumption(String childValueSetId, String childValueSetVersion, String parentValueSetId, String parentValueSetVersion, AbsoluteCodingSchemeVersionReferenceList csVersionList, String versionTag)	
Description:	Determine whether one of the two supplied value sets subsumes the other.
Input:	java.lang.String childValueSetId - (Mandatory) Id of a child value set. java.lang.String childValueSetVersion - (Optional) Version of a child value set. java.lang.String parentValueSetId - (Mandatory) Id of a parent value set. java.lang.String parentValueSetVersion - (Optional) Version of a parent value set. jorg.LexGrid.LexBIG.DataModel.Collections.AbsoluteCodingSchemeVersionReferenceList csVersionList - (Optional) Code System Version reference list to be used to resolve both value sets. java.lang.String versionTag - (Optional) The tag (e.g. "devel", "production", ...) to be used to determine which code system to be used.
Output:	boolean - True; if childValueSet subsumes parentValueSet. False; otherwise.
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	Step 1: Instantiate ValueSetQueryOperation if it is not done yet: <pre>org.lexevs.cts2.query.ValueSetQueryOperation vsQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getValueSetQueryOperation();</pre> Step 2: Populate the Code System Version reference list to be used for resolving the value set definitions: <pre>AbsoluteCodingSchemeVersionReference acsvr = new AbsoluteCodingSchemeVersionReference(); acsvr.setCodingSchemeURN("urn:oid:11.11.0.1"); acsvr.setCodingSchemeVersion("1.0"); AbsoluteCodingSchemeVersionReferenceList csList = new AbsoluteCodingSchemeVersionReferenceList(); csList.addAbsoluteCodingSchemeVersionReference(acsvr);</pre> Step 3: Call checkValueSetSubsumption method by providing child and parent value set id and code system version list: <pre>boolean subsume = vsQueryop.checkValueSetSubsumption("SRITEST:AUTO:DomasticLeafOnly", null, "SRITEST:AUTO:Everything", null, csList, null); </pre>
---------------------	--

checkConceptValueSetMembership

This function determines whether the supplied coded concept exists in the supplied value set.

checkConceptValueSetMembership(String conceptCode, URI entityCodeNamespace, AbsoluteCodingSchemeVersionReference codeSystemAndVersion, String valueSetId, String valueSetVersion, String versionTag)

Description:	Determine whether the supplied coded concept exists in the supplied value set.
Input:	java.lang.String conceptCode - (Mandatory) Coded concept to check membership. java.net.URI entityCodeNamespace - (Optional) Namespace of coded concept. jorg.LexGrid.LexBIG.DataModel.Collections.AbsoluteCodingSchemeVersionReference codeSystemAndVersion - (Optional) Code System Version that contains the coded concept and to be used to resolve value set. java.lang.String valueSetId - (Mandatory) Id of a value set. java.lang.String valueSetVersion - (Optional) Version of a value set. java.lang.String versionTag - (Optional) The tag (e.g. "devel", "production", ...) to be used to determine which code system to be used.
Output:	boolean - True; if coded concept exists in value set. False; otherwise.
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	• <i>Step 1:</i> Instantiate ValueSetQueryOperation if it is not done yet: <pre>org.lexevs.cts2.query.ValueSetQueryOperation vsQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getValueSetQueryOperation();</pre> • <i>Step 2:</i> Populate the Code System Version reference that contains the coded concept and to be used for resolving the value set definition: <pre>AbsoluteCodingSchemeVersionReference acsvr = new AbsoluteCodingSchemeVersionReference(); acsvr.setCodingSchemeURN("urn:oid:11.11.0.1"); acsvr.setCodingSchemeVersion("1.0");</pre> • <i>Step 3:</i> Call checkConceptValueSetMembership method by providing coded concept, code system version and value set id: <pre>boolean member = vsQueryop.checkConceptValueSetMembership("GMC", new URI("Automobiles"), acsvr, "SRITEST:AUTO:GM", null, null); </pre>
---------------------	---

listValueSetsWithConceptCode

This function returns all the value set identifiers that contains supplied concept code.

```
listValueSetsWithConceptCode(String conceptCode, URI entityCodeNamespace,
AbsoluteCodingSchemeVersionReferenceList csVersionList, String versionTag)
```

Description:	Returns all the value set identifiers that contains supplied concept code.
Input:	• <i>java.lang.String conceptCode</i> - (Mandatory) Coded concept to check membership. • <i>java.net.URI entityCodeNamespace</i> - (Optional) Namespace of coded concept. • <i>jorg.LexGrid.LexBIG.DataModel.Collections.AbsoluteCodingSchemeVersionReferenceList csVersionList</i> - (Optional) Code System Version List that contains the coded concept and to be used to resolve value set. • <i>java.lang.String versionTag</i> - (Optional) The tag (e.g. "devel", "production", ...) to be used to determine which code system to be used.
Output:	<i>java.util.List<java.lang.String></i> - List of value set identifiers that contains the supplied concept code.
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Sample Call:	• <i>Step 1:</i> Instantiate ValueSetQueryOperation if it is not done yet: <pre>org.lexevs.cts2.query.ValueSetQueryOperation vsQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getValueSetQueryOperation();</pre> • <i>Step 2:</i> Populate the Code System Version reference list to be used for resolving the value sets: <pre>AbsoluteCodingSchemeVersionReference acsvr = new AbsoluteCodingSchemeVersionReference(); acsvr.setCodingSchemeURN("urn:oid:11.11.0.1"); acsvr.setCodingSchemeVersion("1.0"); AbsoluteCodingSchemeVersionReferenceList csList = new AbsoluteCodingSchemeVersionReferenceList(); csList.addAbsoluteCodingSchemeVersionReference(acsvr);</pre> • <i>Step 3:</i> Call listValueSetsWithConceptCode method by providing coded concept and code system version list: <pre>List<String> valueSetIdList = vsQueryop.listValueSetsWithConceptCode("GMC", new URI ("Automobiles"), csList, null);</pre>

