

LexEVS 6.0 CTS2 Query 4 - Usage Context Query Operation API

Contents of this Page
<ul style="list-style-type: none">• Introduction• Interface• Query Functions<ul style="list-style-type: none">◦ getUsageContextCodingScheme◦ getUsageContextEntity◦ getUsageContextEntitisWithName◦ getUsageContextCodedNodeSet◦ listAllUsageContextEntities◦ listAllUsageContextIds

CTS2 Links for LexEVS 6.0
<ul style="list-style-type: none">• CTS2 API Main Page• Programmer's Guide Main Page• LexEVS 6.0 Main Page• LexEVS Current Release

Introduction

LexEVS CTS2 Usage Context Query API provides capability to query Usage Context available in the system and also to query the binding between Value Set and Concept Domain in conjunction with Usage Context. In LexEVS, the Usage Context is essentially an entity of type 'usageContext' in a code system version. Using LexEVS CTS 2 Authoring or Loader interfaces, you could create a Code System specially for Usage Context entities, but this is not necessary, but will be convenient to have Usage Context entries in specific code system(s).

Interface

org.lexevs.cts2.query.UsageContextQueryOperation is the main interface for all the queries against Usage Context. This interface can be accessed using main LexEVSCTS2 interface, like:

```
org.lexevs.cts2.query.UsageContextQueryOperation cdQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().
getQueryOperation().getUsageContextQueryOperation();
```

Query Functions

Here are the major query functions available using UsageContextQueryOperation interface:

getUsageContextCodingScheme

This function returns detailed usage context code system.

getUsageContextCodingScheme(String codeSystemNameOrURI, String codeSystemVersion)

Description:	Gets the usage context code system.
Input:	<ul style="list-style-type: none">• <i>java.lang.String codeSystemNameOrURI</i> - (Mandatory) Name or URI of the code system that contains usage context entities.• <i>java.lang.String codeSystemVersion</i> - (Optional) Version of of the code system that contains usage context entities.
Output:	<i>org.LexGrid.codingSchemes.CodingScheme</i> - Detailed usage context Code System
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	<ul style="list-style-type: none"> • <i>Step 1:</i> Instantiate UsageContextQueryOperation if it is not done yet: <pre>org.lexevs.cts2.query.UsageContextQueryOperation ucQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getUsageContextQueryOperation();</pre> <ul style="list-style-type: none"> • <i>Step 2:</i> Call getUsageContextCodingScheme method to get detailed Usage Context code system: <pre>org.LexGrid.codingSchemes.CodingScheme usageContextCS = ucQuery.getUsageContextCodingScheme("usageContextCodingScheme", "1.0");</pre>
---------------------	---

getUsageContextEntity

This function returns detailed Usage Context entity.

```
getUsageContextEntity(String usageContextId, String namespace, String codeSystemNameOrURI, String codeSystemVersion)
```

Description:	Returns Usage Context entity object.
Input:	<ul style="list-style-type: none"> • <i>java.lang.String usageContextId</i> - (Mandatory) ID of a Usage Context. • <i>java.lang.String namespace</i> - (Optional) Namespace of a Usage Context. • <i>java.lang.String codeSystemNameOrURI</i> - (Mandatory) Name or URI of code system that contains the Usage Context. • <i>java.lang.String codeSystemVersion</i> - (Optional) Version of code system that contains the Usage Context.
Output:	<i>org.LexGrid.concepts.Entity</i> - Detailed Usage Context Entity
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Sample Call:	<ul style="list-style-type: none"> • <i>Step 1:</i> Instantiate UsageContextQueryOperation if it is not done yet: <pre>org.lexevs.cts2.query.UsageContextQueryOperation ucQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getUsageContextQueryOperation();</pre> <ul style="list-style-type: none"> • <i>Step 2:</i> Call getUsageContextEntity method to get detailed Usage Context Entity: <pre>org.LexGrid.concepts.Entity usageContextEntity = ucQuery.getUsageContextEntity("ActAdjudicationInformationCode", null, "usageContextCodingScheme", "1.0"); </pre>

getUsageContextEntitisWithName

This function returns all the Usage Context entities that matches the name supplied.

```
getUsageContextEntitisWithName(String usageContextName, String codeSystemNameOrURI, String codeSystemVersion, SearchDesignationOption option, String matchAlgorithm, String language)
```

Description:	Returns list of Usage Context entities matching the name provided.
Input:	<ul style="list-style-type: none"> • <i>java.lang.String usageContextName</i> - (Mandatory) Name of a Usage Context to match. • <i>java.lang.String codeSystemNameOrURI</i> - (Mandatory) Name or URI of code system that contains the Usage Context. • <i>java.lang.String codeSystemVersion</i> - (Optional) Version of code system that contains the Usage Context. • <i>org.LexGrid.LexBIG.LexBIGService.CodedNodeSet.SearchDesignationOption option</i> - (Optional) Indicates the designations to search (one of the enumerated type SearchDesignationOption). • <i>java.lang.String matchAlgorithm</i> - (Mandatory) Local name of the match algorithm - possible algorithms are returned in LexEVSCTS2.getSupportedSearchAlgorithmNames(). • <i>java.lang.String language</i> - (Optional) Language of search string. If missing, uses the default language specified in the context.

Output:	<i>java.util.List<org.LexGrid.concepts.Entity></i> - List of detailed Usage Context entities that match supplied name.
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Sample Call:	<ul style="list-style-type: none"> <i>Step 1:</i> Instantiate UsageContextQueryOperation if it is not done yet: <pre>org.lexevs.cts2.query.UsageContextQueryOperation ucQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getUsageContextQueryOperation();</pre> <i>Step 2:</i> Call <code>getUsageContextEntitisWithName</code> method by passing the name to search: <pre>java.util.List<org.LexGrid.concepts.Entity> usageContextEntities = ucQuery.getUsageContextEntitisWithName("code", "usageContextCodingScheme", null, null, "subString", null); </pre>

getUsageContextCodedNodeSet

This is a helper function that returns Coded Node Set(CNS) for Usage Context entities. Further restrictions like, `matchingDesignation`, `Status`, `Properties`, etc., can be applied on this CNS and then resolved.

```
getUsageContextCodedNodeSet(String codeSystemNameOrURI, String codeSystemVersion)
```

Description:	Returns coded node set for UsageContext entities.
Input:	<ul style="list-style-type: none"> <i>java.lang.String codeSystemNameOrURI</i> - (Mandatory) Name or URI of code system that contains the Usage Context. <i>java.lang.String codeSystemVersion</i> - (Optional) Version of code system that contains the Usage Context.
Output:	<i>org.LexGrid.LexBIG.LexBIGService.CodedNodeSet</i> - Unresolved Usage Context coded node set.
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Sample Call:	<ul style="list-style-type: none"> <i>Step 1:</i> Instantiate UsageContextQueryOperation if it is not done yet: <pre>org.lexevs.cts2.query.UsageContextQueryOperation ucQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getUsageContextQueryOperation();</pre> <i>Step 2:</i> Call <code>getUsageContextCodedNodeSet</code> to get Usage Context Coded node Set: <pre>CodedNodeSet usageContextCNS = ucQuery.getUsageContextCodedNodeSet("usageContextCodingScheme", null);</pre> <i>Step 3:</i> Apply restrictions to the CNS and resolve: <pre>usageContextCNS.restrictToMatchingProperties(null, new PropertyType[] { PropertyType.PRESENTATION }, "Auto", "startsWith", null); org.LexGrid.LexBIG.DataModel.Core.ResolvedConceptReference[] rcr = usageContextCNS.resolveToList(null, null, null, 1).getResolvedConceptReference();</pre>

listAllUsageContextEntities

This function returns all the Usage Context entities found in supplied code system version.

```
listAllUsageContextEntities(String codeSystemNameOrURI, String codeSystemVersion)
```

Description:	Returns all the Usage Context entities found in supplied code system version.
Input:	<ul style="list-style-type: none"> • <i>java.lang.String codeSystemNameOrURI</i> - (Mandatory) Name or URI of code system that contains the Usage Context. • <i>java.lang.String codeSystemVersion</i> - (Optional) Version of code system that contains the Usage Context.
Output:	<i>java.util.List<org.LexGrid.concepts.Entity></i> - List of Usage Context entities found in supplied code system version.
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Sample Call:	<ul style="list-style-type: none"> • <i>Step 1:</i> Instantiate UsageContextQueryOperation if it is not done yet: <pre>org.lexevs.cts2.query.UsageContextQueryOperation ucQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getUsageContextQueryOperation();</pre> • <i>Step 2:</i> Call listAllUsageContextEntities to get all Usage Context entities: <pre>java.util.List<org.LexGrid.concepts.Entity> usageContexts = ucQuery.listAllUsageContextEntities("usageContextCodingScheme", "1.0");</pre>

listAllUsageContextIds

This function is similar to previous one, but instead of returning detailed Usage Context entities, it just returns identifiers of the Usage Context.

```
listAllUsageContextIds(String codeSystemNameOrURI, String codeSystemVersion)
```

Description:	Returns all the Usage Context identifiers found in supplied code system version.
Input:	<ul style="list-style-type: none"> • <i>java.lang.String codeSystemNameOrURI</i> - (Mandatory) Name or URI of code system that contains the Usage Context. • <i>java.lang.String codeSystemVersion</i> - (Optional) Version of code system that contains the Usage Context.
Output:	<i>java.util.List<String></i> - List of Usage Context identifiers found in supplied code system version.
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Sample Call:	<ul style="list-style-type: none"> • <i>Step 1:</i> Instantiate UsageContextQueryOperation if it is not done yet: <pre>org.lexevs.cts2.query.UsageContextQueryOperation ucQueryOp = new org.lexevs.cts2.LexEvsCTS2Impl().getQueryOperation().getUsageContextQueryOperation();</pre> • <i>Step 2:</i> Call listAllUsageContextIds to get all Usage Context identifiers: <pre>java.util.List<java.lang.String> usageContextIds = ucQuery.listAllUsageContextIds("usageContextCodingScheme", "1.0");</pre>