xTree REST Anwender-Dokumentation

zu xTree REST Version 0.86 (2020-09-28)

Aktualisierungen von Version 0.80 nach Version 0.81 (07.10.2014):

- Neu: 2.1.i.8. ***processingStatus*** (Bearbeitungsstatus)
- Neu: 2.1.i.9. ***lastUpdated*** (letztes Update)

Aktualisierungen von Version 0.81 nach Version 0.83 (29.07.2019):

- Neu: Zusätzlicher Wert für 14.a. mappingRelation: unqualifizierte Mapping-Relation ***mappingRelation***

Aktualisierungen von Version 0.83 nach Version 0.84 (30.04.2020):

- Neue Methode getDatasetsForUser unter 1.11.

Aktualisierungen von Version 0.84 nach Version 0.85 (13.05.2020):

- Aktivierung von HTTPS

Aktualisierungen von Version 0.85 nach Version 0.86 (28.09.2020):

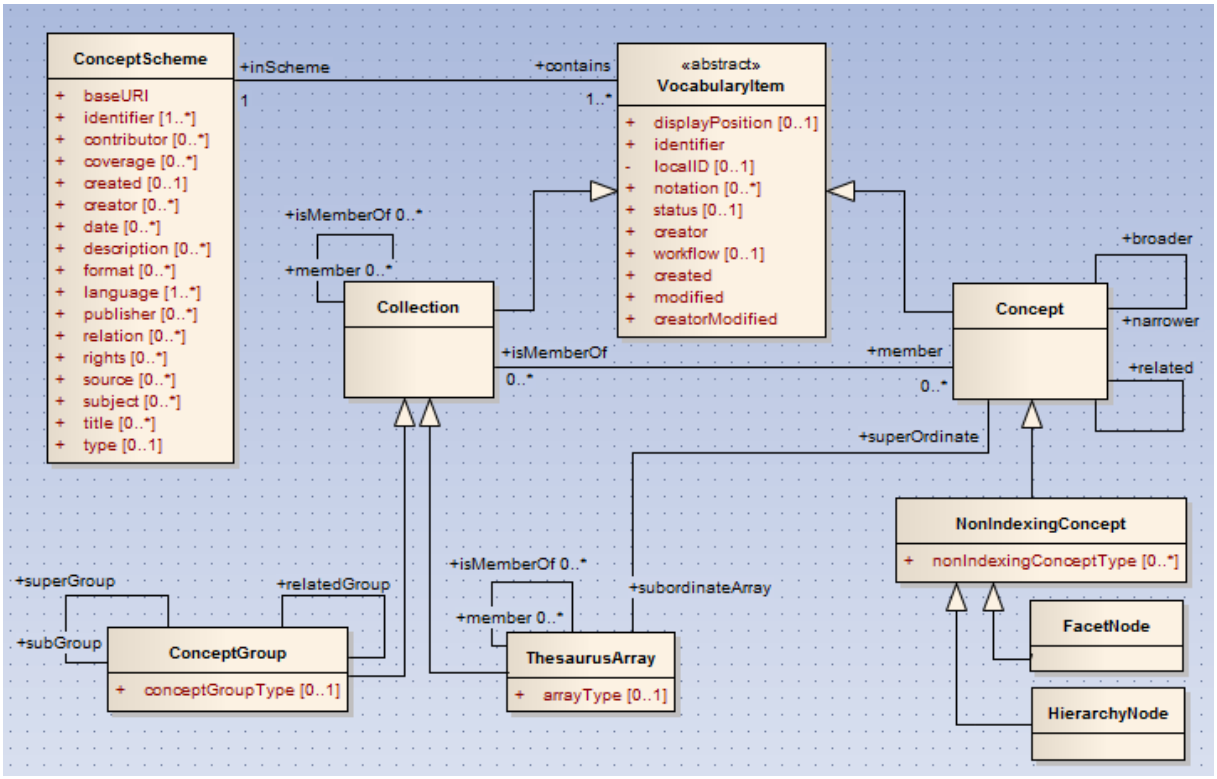
- Geänderter json-Header
- Neu: 2.1.i.11: ***languageVariety*** (Sprachvarietät)

Dieses Dokument beschreibt die Verwendung der RESTful Methoden und das zurückgelieferte JSON Format für die xTree REST API

Die Methoden und das JSON -Format liegen noch nicht in der finalen Version vor. Änderungshinweise werden zukünftig in der Doku mitgeführt.

xTree REST orientiert sich weitestgehend an dem vocnet-Schema in der Version 0.8:
http://schema.vocnet.org

UML der Vokabularelement-Klassen:



Quelle: Jutta Lindenthal

URLs zu den Anwendungen:

- xTree REST, Version 0.85 (Produktiv):
  https://xtree-rest.digicult-verbund.de

1. **REST Methoden**

   Es gibt 11 lesende REST-Methoden. Die Methoden können über den Standard-Port 80 des HTTP Protokolls als GET Anfrage gesendet werden.

   1. **getTopClassTC**

      gibt alle Begriffe + nonIndexingConcepts + FacetNodes + HierarchyNodes zurück die keine übergeordneten Elemente von dem gleichen vocItem-Typ (vocItem = Vokabular Element) haben.

      Parameter:

      i. **vocabulary** (URI, Pflichtfeld)

      ii. **start** (für Blätterfunktionalität, z.B. start= 10: zeige count Einträge ab 10)

      iii. **count** (für Blätterfunktionalität, z.B. count=20: zeige 20 Einträge ab start)

      iv. **jsonfull** (Rückgabemodus)

      je nach angefordertem Rückgabemodus enthält das zurückgelieferte JSON die vollständigen vocItems oder nur Teile davon.

      0 = standard

      1 = full (wie standard + Notes und Semantic Mappings)

      v. **lang**

      all oder Sprachcode

   2. **getTopCategories**

      gibt alle Kategorien zurück, die keine übergeordneten Kategorien haben.

      Parameter wie 1.1

   3. **getTopConcepts**

      gibt alle Begriffe zurück, die keine übergeordneten Begriffe haben.

      Parameter wie 1.1

   4. **getTopNonIndexingConcepts**

      gibt alle nonIndexingConcepts zurück, die keine übergeordneten nonIndexingConcepts haben.

      Parameter wie 1.1

   5. **getTopFacetNodes**

      gibt alle FacetNodes zurück, die keine übergeordneten FacetNodes haben.

      Parameter wie 1.1

   6. **getTopHierarchyNodes**

      gibt alle HierarchyNodes zurück, die keine übergeordneten HierarchyNodes haben.

      Parameter wie 1.1

   7. **getTopNodeLabels**

      gibt alle NodeLabels zurück, die keine übergeordneten NodeLabels haben.

      Parameter wie 1.1

   8. **getSearchVocItemsByTerm**

      Methode für die Suche nach Bezeichnungen im verwendeten Vokabular.

      Parameter:

      i. **vocabulary** (URI, Pflichtfeld)

      ii. **searchtermslist**

      Durch Leerzeichen getrennte Suchbegriffe. Links- oder Rechtstrunkierung möglich (z.B. „*Burg“, „Burg*)

iii. **mode (**Pflichtfeld)
UND / ODER Suche
mode = AND (UND Suche)
mode = OR (ODER Suche)
iv. **start** (s. 1.1.ii)
v. **count** (s. 1.1.iii)
vi. searchfields
Auswahl über die Rollen der Bezeichnungen
searchfields = all | alt | hidden | pref | quasi-synonym | useBroader | useComb
vii. typeofvocitem
Auswahl der zurückgelieferten Vokabularelemente
viii. restrictedto
Durch Leerzeichen getrennte Liste an Identifikatoren auf die die Suche beschränkt sein soll
ix. **jsonfull** (s. 1.1.iv)
x. **lang** (s. 1.1.v)
xi. homonymlexicalvalue
Suchfeld für Homonyme
Vorsicht: Der Parameter "homonymlexicalvalue" wir zukünftig noch in "qualifier" umbenannt

9. getSearchVocItemsById

i. **vocabulary** (URI, Pflichtfeld)
ii. **searchidslist**
Durch Leerzeichen getrennte Identifikatoren
iii. **start** (s. 1.1.ii)
iv. **count** (s. 1.1.iii)
v. **typeofvocitem** (s.2.8.vii)
vi. **jsonfull** (s. 1.1.iv)
vii. **lang** (s. 1.1.v)

10. getFetchHierarchy

Methode für die Rückgabe von einem Hierarchiezweig im angegebenen Vokabular.
Parameter:

i. **vocabulary** (URI, Pflichtfeld)
ii. **nodeid**
ein vokabularspezifischer Identifikator, Pflichtfeld
iii. direction
up oder down, Pflichtfeld
iv. **level**
1 oder N, Pflichtfeld
v. **start** (s. 1.1.ii)
vi. **count** (s. 1.1.iii)
vii. **typeofvocitem** (s.2.8.vii)

viii. **term**
optionaler Suchterm

ix. **jsonfull** (s. 1.1.iv)

x. **lang** (s. 1.1.v)

11. **getDatasetsForUser**
Über die Methode getDatasetsForUser werden die einem REST-Account zugewiesenen Datasets zurückgeliefert. Siehe dazu das Format unter Nr. 3.

2. JSON Format Vokabularelemente

ein in http://jsonviewer.net decodierter JSON String aus xTree REST:


```
$json (
    vocItemCount =  1
    VocabularyItem => Array (1)
    (
        ['0'] (
            Concept (
                id =  "601.78"
                status =  "approved"
                displayPosition =  "10"
                inScheme =  "http://digicult.vocnet.org/stil"
                Term => Array (3)
                (
                    ['0'] (
                        labelRole =  "prefLabel"
                        lang =  "de"
                        Term =  "Pointillismus"
                    )
                    ['1'] (
                        labelRole =  "prefLabel"
                        lang =  "en"
                        Term =  "Pointillism"
                    )
                    ['3'] (
                        labelRole =  "altLabel"
                        lang =  "de"
                        Term =  "Chromoluminarismus"
                    )
                )
                broader => Array (1)
                (
                    ['0'] (
                        Concept (
                            id =  "601.2"
                            status =  "approved"
                            inScheme =  "http://digicult.vocnet.org/stil"
                            Term => Array (1)
                            (
                                ['0'] (
                                    labelRole =  "prefLabel"
                                    lang =  "de"
                                    Term =  "Kunst des 19. Jhdt./Frühe Moderne"
                                )
                            )
                        )
                    )
                )
            )
        )
    )
)
```

Der Aufbau des JSON Formats:

1. **VocabularyItem** => Array
    i. **Concept | ConceptGroup | ThesaurusArray | NonIndexingConcept | FacetNode | HierarchyNode**

    unterschiedliche Benennungen für vocItems:
    - Concept = Begriff
    - ConceptGroup = Kategorie = Begriffsgruppe
    - ThesaurusArray = Node Label = Gruppierung nach Merkmal
    - NonIndexingConcept = Begriff als Nichtdeskriptor
    - FacetNode = Facette
    - HierarchyNode = Hierachieknoten

    1. **id**
       lokaler xTree Identifikator
    2. **status**
    3. **notation**
    4. **displayPosition**
    5. **earliestDate**
    6. **latestDate**
    7. **pk**
       Primary Key
    8. **processingStatus**
       Bearbeitungsstatus
    9. **lastUpdated**
       letztes Update
    10. **inScheme**
        URI Vokabular
    11. **Term** => Array
        a. **labelRole**
        b. **lang**
        c. **grammaticalNumber**
        d. **pk**
           Primary Key
        e. **Term**
        f. **qualifier**
           Homonymzusatz, neu in Version 0.79
        g. **languageVariety**
           Sprachvarietät
           i. Wertevorrat:
              1. Standardsprache
                 http://digicult.vocnet.org/terminology/001

2. Amtssprache
   http://digicult.vocnet.org/terminology/008
3. Einfache Sprache
   http://digicult.vocnet.org/terminology/015
4. Fachsprache
   http://digicult.vocnet.org/terminology/004
5. Formel
   http://digicult.vocnet.org/terminology/016
6. Jargon
   http://digicult.vocnet.org/terminology/017
7. Regionalsprache
   http://digicult.vocnet.org/terminology/003
8. Sondersprache
   http://digicult.vocnet.org/terminology/005
9. Umgangssprache
   http://digicult.vocnet.org/terminology/009
10. Vernakularsprache
    http://digicult.vocnet.org/terminology/018
11. Wissenschaftssprache
    http://digicult.vocnet.org/terminology/011

12. hierarchische Relationen

- **a.** für Concept:
  - **i.** **broader** => Array
    übergeordnete vocItems: Concept, NonIndexingConcept, FacetNode, HierarchyNode
  - **ii.** **narrower** => Array
    untergeordnete vocItems: Concept, NonIndexingConcept, FacetNode, HierarchyNode
  - **iii.** **isMemberOf** => Array
    übergeordnet: ThesaurusArray, ConceptGroup
  - **iv.** **subordinateArray** => Array
    untergeordnet: ThesaurusArray
- **b.** für ConceptGroup
  - **i.** **superGroup** => Array
    übergeordnet: ConceptGroup
  - **ii.** **subGroup** => Array
    untergeordnet: ConceptGroup
  - **iii.** **member** => Array
    zugeordnet: Concept, NonIndexingConcept, FacetNode, HierarchyNode
- **c.** für ThesaurusArray
  - **i.** **superOrdinate** => Array
    übergeordnete vocItems: Concept, NonIndexingConcept, FacetNode, HierarchyNode
  - **ii.** **member => Array**
    untergeordnete vocItems: Concept,

NonIndexingConcept, FacetNode, HierarchyNode, ThesaurusArray

iii. **isMemberOf**
übergeordnet: ThesaurusArray

d. für NonIndexingConcept
wie Concept

e. für FacetNode
wie Concept

f. für HierarchyNode
wie Concept

13. **Note** => Array
    a. **lang**
    b. **source**
    c. Anmerkungstypen
        i. **changeNote**
        ii. **definition**
        iii. **editorialNote**
        iv. **etmology**
        v. **example**
        vi. **explanation**
        vii. **extDefinition**
        viii. **glossaryEntry**
        ix. **historyNote**
        x. **miscNote**
        xi. **note**
        xii. **scopeNote**
14. **MapItem** => Array
    a. **mappingRelation**
    mappingRelation | closeMatch | exactMatch | broadMatch | narrowMatch | relatedMatch
    s.a. http://www.w3.org/TR/skos-reference/#mapping
    b. **mapItemID**
    c. **mapItemIDType**
    uri | url | hdl | local | urn | doi
    d. **mapItemLabel**
    e. **mapItemSource**
15. **related** => Array
    assoziative Beziehung zwischen zwei Begriffen

3. **JSON Format für die Methode getDatasetsForUser**
Aufbau JSON-Format:
    1. **count**
    Anzahl der für das Account freigegebenen Vokabulare oder Teilvokabulare

2. **Datasets**

   JSON-Objekt mit untergeordnetem Array

   i. **uriSpace**

      Liste mit URIs

Beispiel:


```
{
    "count": 2,
    "Datasets": {
        "uriSpace": [
            "http://lvr.vocnet.org/wnk",
            "http://lvr.vocnet.org/datierung"
        ]
    }
}
```