

Weikersheim, Residenzschloss

Team Redaktion

2024-03-22

Table of contents

Chapter 1

Katalog zur Ausstellung: Der Große Saal (Rittersaal)

Ein Katalog mit Kunstwerken aus der CbDD-Sammlung. Textteil: 6e73f774-4b7f-4e37-937b-e11cc35c5bc8

Der Große Saal (Rittersaal) [Raum]

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International License.

2CHAPTER 1. KATALOG ZUR AUSSTELLUNG: DER GROSSE SAAL (RITTERSAAL)

Chapter 2

Die Saaldecke der Renaissance von Balthasar Katzenberger

```
from datetime import datetime
import sys
import time
from SPARQLWrapper import SPARQLWrapper, JSON
import requests
from PIL import Image
import html

endpoint_url = "https://computational-publishing-service.wikibase.cloud/query/sparql"

query_txt = """PREFIX cps: <https://computational-publishing-service.wikibase.cloud/entity/>
PREFIX cpss: <https://computational-publishing-service.wikibase.cloud/entity/statement/>
PREFIX cpsv: <https://computational-publishing-service.wikibase.cloud/value/>
PREFIX cpspt: <https://computational-publishing-service.wikibase.cloud/prop/direct/>
PREFIX cpsp: <https://computational-publishing-service.wikibase.cloud/prop/>
PREFIX cpsps: <https://computational-publishing-service.wikibase.cloud/prop/statement/>
PREFIX cpspq: <https://computational-publishing-service.wikibase.cloud/prop/qualifier/>

SELECT ?textItem ?kuratorLabel ?textUrl
WHERE
{
  <placeholder>
  ?textItem cpsp:P46 ?kuratorStatement.
```

```

?kuratorStatement cpsps:P46 ?kuratorItem.
?kuratorItem rdfs:label ?kuratorLabel.
?textItem csp:P57 ?urlstatement.
?urlstatement cpsps:P57 ?textUrl.
}""

query_img = ""PREFIX cps: <https://computational-publishing-service.wikibase.cloud/entity/>
PREFIX cpss: <https://computational-publishing-service.wikibase.cloud/entity/statement/>
PREFIX cpsv: <https://computational-publishing-service.wikibase.cloud/value/>
PREFIX cpspt: <https://computational-publishing-service.wikibase.cloud/prop/direct/>
PREFIX cpsp: <https://computational-publishing-service.wikibase.cloud/prop/>
PREFIX cpsps: <https://computational-publishing-service.wikibase.cloud/prop/statement/>
PREFIX cpspq: <https://computational-publishing-service.wikibase.cloud/prop/qualifier/>

SELECT DISTINCT ?itemLabel ?itemDescr ?imgItem ?imgUrl ?publishDate
WHERE
{
  ?imgItem cpsp:P107 ?urlStatement.
  ?urlStatement cpsps:P107 ?imgUrl.
  ?imgItem cpsp:P60 ?dateStatement.
  ?dateStatement cpsps:P60 ?publishDate.
  ?imgItem cpsp:P6 ?partOfStatement.
  ?partOfStatement cpsps:P6 ?partOfItem.
  <placeholder>

  FILTER (datatype(?publishDate) = xsd:edtf)

  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "en,de".
    ?imgItem rdfs:label ?itemLabel.
    ?imgItem schema:description ?itemDescr.
  }
}""

query_graph = ""PREFIX cps: <https://computational-publishing-service.wikibase.cloud/entity/>
PREFIX cpss: <https://computational-publishing-service.wikibase.cloud/entity/statement/>
PREFIX cpsv: <https://computational-publishing-service.wikibase.cloud/value/>
PREFIX cpspt: <https://computational-publishing-service.wikibase.cloud/prop/direct/>
PREFIX cpsp: <https://computational-publishing-service.wikibase.cloud/prop/>
PREFIX cpsps: <https://computational-publishing-service.wikibase.cloud/prop/statement/>
PREFIX cpspq: <https://computational-publishing-service.wikibase.cloud/prop/qualifier/>

SELECT ?x ?y
WHERE
{
  ?a cpsp:P2 ?c.

```



```

?c cpsps:P2 ?d.
?a rdfs:label ?x.
?d rdfs:label ?y.

}LIMIT 1"""

query_graph2 = """
SELECT ?a ?b ?c
WHERE
{
    ?a rdfs:label ?c
}LIMIT 100"""

def run_query(endpoint_url, query):
    user_agent = "WDQS-example Python/%s.%s" % (sys.version_info[0], sys.version_info[1])
    # TODO adjust user agent; see https://w.wiki/CX6
    sparql = SPARQLWrapper(endpoint_url, agent=user_agent)
    sparql.setQuery(query)
    sparql.setMethod("POST") #this NEEDS to be added to get results (not included in the wikibase
    sparql.setReturnFormat(JSON)
    return sparql.query().convert()

def get_text(textitem_id):
    q = ""
    if textitem_id:
        q = query_txt.replace("<placeholder>", "cps:"+textitem_id+" cpsp:P46 ?kuratorStatement.")
    else:
        q = query_txt.replace("<placeholder>", "")

    results_txt = run_query(endpoint_url, q)
    for item in results_txt["results"]["bindings"]:
        # print(item)
        print('Wikibase link: ' + '[' + item['textItem']['value'] + ']' + '(' + item['textItem']
        print('Kurator: ' + item['kuratorLabel']['value'] + '\n')
        headers = {'User-Agent': 'Ex_Books_conference_bot/0.0 (https://github.com/SimonXIX/Experi
        r = requests.get(item['textUrl']['value'], headers=headers, stream=True)
        text = str(r.content)
        text = text.replace("ä", "&auml;")
        text = text.replace("Ä", "&Auml;")
        text = text.replace("ö", "&ouml;")
        text = text.replace("Ö", "&Ouml;")
        text = text.replace("ü", "&uuml;")
        text = text.replace("Ü", "&Uuml;")
        text = text.replace("ß", "&szlig;")

```

```

        text = text.replace('\\n', "<br>")
        text = str(text)
        text = text.removeprefix("b'<!DOCTYPE html>").removesuffix("")
        print(text)

def get_delay(date):
    try:
        date = datetime.datetime.strptime(date, '%a, %d %b %Y %H:%M:%S GMT')
        timeout = int((date - datetime.datetime.now()).total_seconds())
    except ValueError:
        timeout = int(date)
    return timeout

def fetch_image_by_url(url, headers):
    r = requests.get(url, headers=headers, stream=True)
    if r.status_code == 200:
        im = Image.open(r.raw)
        return im
    if r.status_code == 500:
        return None
    if r.status_code == 403:
        return None
    if r.status_code == 429:
        timeout = get_delay(r.headers['retry-after'])
        print('Timeout {} m {} s'.format(timeout // 60, timeout % 60))
        time.sleep(timeout)
        fetch_image_by_url(url, headers)

def get_img(partOfItem_id):
    q = ""
    if partOfItem_id:
        q = query_img.replace("<placeholder>", "?partOfStatement cpsps:P6 cps:"+partOfItem_id)
    else:
        q = query_img.replace("<placeholder>", "")
    results_img = run_query(endpoint_url, q)
    for item in results_img["results"]["bindings"]:
        #print(item)
        print('Wikibase link: ' + '[' + item['imgItem']['value'] + '] + (' + item['imgItemLabel']['value'] + ')')
        print('Title: ' + item['itemLabel']['value'] + '\n')
        print('Year: ' + item['publishDate']['value'] + '\n')
        print('Description: ' + html.unescape(item['itemDescr']['value']) + '\n')

    # get image from image URL and display resized version
    image_url=item['imgUrl']['value']
    headers = {'User-Agent': 'Ex_Books_conference_bot/0.0 (https://github.com/SimonX...

```

```

    im = fetch_image_by_url(image_url, headers)
    im.thumbnail((500, 500), Image.Resampling.LANCZOS)
    display(im)
    print('\n\n')

def get_graph():
    import VizKG.visualize as vkg
    results_graph1 = run_query(endpoint_url, query_graph)
    #print(results_graph1)
    #print('---')
    results_graph2 = run_query(endpoint_url, query_graph2)
    #print(results_graph2)

    chart = vkg(sparql_query=query_graph2, sparql_service_url=endpoint_url, chart='wordcloud')
    chart.plot()

```

How to use your own text for processing

1. Add a new Text item to the wikibase. link to wikibase new item the item should contain the following statements:
 - P57 (external link): link to the html file containing the new text
 - P46 (kurator): Item of the curator. you may use an existing item like Q210 (Ulrike seeger) for test purposes
 - P53 (license): Item of a license for the text. e.g Q203 (CC BY-NC-ND 4.0 DEED)
 - P6 (is part of): set value to Q218 (Schlossanlage Weikersheim)
2. check if your new text item occurs in the list of selected text items: Link to wikibase query service
3. set parameter of `get_text()` to the id of your new text item e.g.: `get_text("Q209")`

```
get_text("Q229")
```

Wikibase link: <https://computational-publishing-service.wikibase.cloud/entity/Q229>

Kurator: Seeger, Ulrike

b'<!DOCTYPE html>

Die Saaldecke der Renaissance von Balthasar Katzenberger

Vertragsbedingungen

Der Vertrag zu den 69 Deckengemälden des Großen Saals zwischen Graf Wolfgang und Balthasar Katzenberger hat sich erhalten.[1] Darin wurde am 22. September 1601 festgelegt, dass der Maler Balthasar Katzenberger

8CHAPTER 2. DIE SAALDECKE DER RENAISSANCE VON BALTHASAR KATZENBERGER

aus Würzburg „Ihren gnaden [Graf Wolfgang] die deckh im Neuen Saal mit Wasserfarb auff Tuech von allerley Jagden, Waydtwerkh und andern was Ire g. Ime jedesmals fürgeben und beuehlen laßen, aufs schönst Säuberst, Künstlichstelichen und frech aussehendt mallen soll, alle Simbs der gannzen Deckh sowoll auch neben herrumb das Simbs alles mit brauner nus oder sonsten ein Dunckhel holz färb, wie es Iren gnaden gefellig anstreichen“.[2]

Graf Wolfgang scheint sowohl das Thema der Jagd vorgegeben als auch die zugehörigen druckgraphischen Vorlagen zur Verfügung gestellt zu haben. Den Passus „allerley Jagden, Waydtwerkh und andern was Ire g. Ime jedesmals fürgeben und beuehlen laßen“ hat man wahrscheinlich dahingehend zu deuten, dass der Auftraggeber in absehbarer Zeit noch weitere Vorlagen liefern könnte. Die zur Anwendung gelangte Technik „mit Wasserfarb auff Tuech“ scheint nur die zweite Wahl gewesen zu sein. In den Vertrag wurde der Zusatz aufgenommen, dass, sollte Graf Wolfgang sich doch noch für Ölfarben entscheiden, er anstatt der vereinbarten 195 Gulden 260 Gulden zu zahlen habe, jeweils zuzüglich der täglichen Verpflegung:

„Da [= Falls] es aber Ihren Gnaden gefellig wer solche deckh mit öll färb zuverferttigen soll Ihme für seine belohnung gegeben werden, Zway hundert und Sechzig gülden. Die Cost und Suppen wie gemelt“.[3]

Laut Restaurierungsbericht malte Katzenberger in Leimfarben auf grober, hellgelb grundierter Leinwand.[4] Erst Christian Thalwitzer, der die Gemälde 1710/11 überarbeitete, verstärkte ihre Leuchtkraft mit einer roten Grundierung und Ölfarben, was ebenfalls die Restaurierung der Jahre 1982–1989 erbrachte. Eines der quadratischen Gemälde (Q1) überliefert auf der Rückseite die originale Maltechnik. Katzenberger hatte das Gemälde angelegt und in der rechten Bildhälfte nahezu fertiggestellt, als sich für die in der linken Bildhälfte angelegte Figur eine Änderung ergab. Da Leimfarben schlecht decken, verzichtete er auf eine Übermalung und drehte die Leinwand kurzerhand um.[5]

Die Gemälde entstanden in der Werkstatt, wobei für die achteckigen Gemälde mit einer Höhe von 3,65 Metern ein Gerüst gezimmert werden musste. Da der Vertrag zu Beginn der dunklen Jahreszeit Ende September abgeschlossen wurde, legte Graf Wolfgang vorsorglich fest, dass Katzenberger nur bei Tageslicht malen dürfe: „In Summa solche Deckh wie gemelt [= wie oben vereinbart] er selbst alles bej tag und nit bej nacht aufs Künstlichst und schönst machen und verferttigen“. Der Auftraggeber stellte die Leinwand, die Farben, Gold und Steinöl für die Gesimse. Gemalt hat Katzenberger die Bilder unter Aufsicht des Grafen in Weikersheim, da sein Lohn neben den 195 Gulden aus morgendlicher und abendlicher Verpflegung mit Brot und Suppe ohne Fleisch bestand.

Katzenberger benötigte für die Arbeit, die er ganz allein, also ohne Kompagnon, nur mit Malergehilfen leistete, dreizehn Monate. Die Fertigstellung quittierte er am 22. November 1602.[6] In die zahlreichen Künstlersignaturen von Graf Wolfgangs Renaissanceausstattung reihte er sich auf dem Achteck-Gemälde Nr. 13 ein, das sich knapp östlich der Mitte der Decke befindet. Sinnfällig nutzte er

das Thema der Wildkatzenjagd für ein Selbstporträt mit Pinsel, Malstock und Palette. Rechts unten notierte er in schwarzer Schrift: „Balthasar Katzenberger vo[n] Wurtzburg maler hat diese gantze Decken in 13 monat alleins gemalet 1602“.[7] Rechnet man sechs Arbeitstage pro Woche, so entfallen fünf Tage auf ein Bild, wobei freilich die 12 Blumenbilder deutlich weniger Zeit in Anspruch nahmen als die 19 großen Achteckbilder.

Balthasar Katzenberger schuf für Schloss Weikersheim seine umfangreichsten erhaltenen Werke. Über weitere Anhaltspunkte zu seinem Oeuvre und seinem Lebensweg unterrichtet der Eintrag im Allgemeinen Künstlerlexikon.[8]

How to select images for processing

Images are selected via the sparql query. The method `get_img()` is capable of using a wikibase item id as parameter to select images with the property P6 (is part of) linking to the given item id.

1. select a valid location id from the query result: Link to wikibase query service

2. set parameter of `get_img()` to the id of your selected location item e.g.:
`get_img("Q217")`

```
get_img("Q229")
```

Wikibase link: <https://computational-publishing-service.wikibase.cloud/entity/Q230>

Title: Ceiling Decoration of the Knights' Hall – Eastern Part of the Ceiling

Year: 2018

Description: Balthasar Kazenberger, painter, 22.09.1601/22.11.1602 - Jan van der Straet, painter - Christian Thalwitzer, restorer, 1710/1711