

# WEBBASIERTE SATZVORSCHAU MIT TUSTEP

## Ein Praxisbericht aus unterschiedlichen Editionsprojekten mit TEI- und MEI-Daten

05.10. 2021 / 27. ITUG-Jahrestagung

Kai Weber /  pagina GmbH /  @fruehlingstag /  sermo-de-arboribus /  kai-weber

## Ausgangslage: Projekte mit lokaler Installation

- Teamarbeit an Editionen oder Werken
- Dienstleister arbeitet parallel an TUSTEP-basierter Satzvorschau
- hoher Administrationsaufwand für lokale Installationen:
  - unterschiedliche Betriebssysteme bzw. OS-Versionen
  - unterschiedliche TUSTEP-Versionen installiert
  - bei Aktualisierung der Preview-Skripte Verteilung an alle Beteiligte nötig
  - Aufwand wächst linear mit Zahl der Teammitglieder

# Vor- und Nachteile Serverlösungen

## Vorteile

- Einrichtung und Administration kann zentral an einer Stelle vorgenommen werden
- Fixaufwand für Aktualisierungen unabhängig von Teamgröße

## Nachteil

- Bei sehr hoher Auslastung können sich lange Wartezeiten ergeben und eine Skalierung von Ressourcen nötig werden (Load Balancing o.ä.)

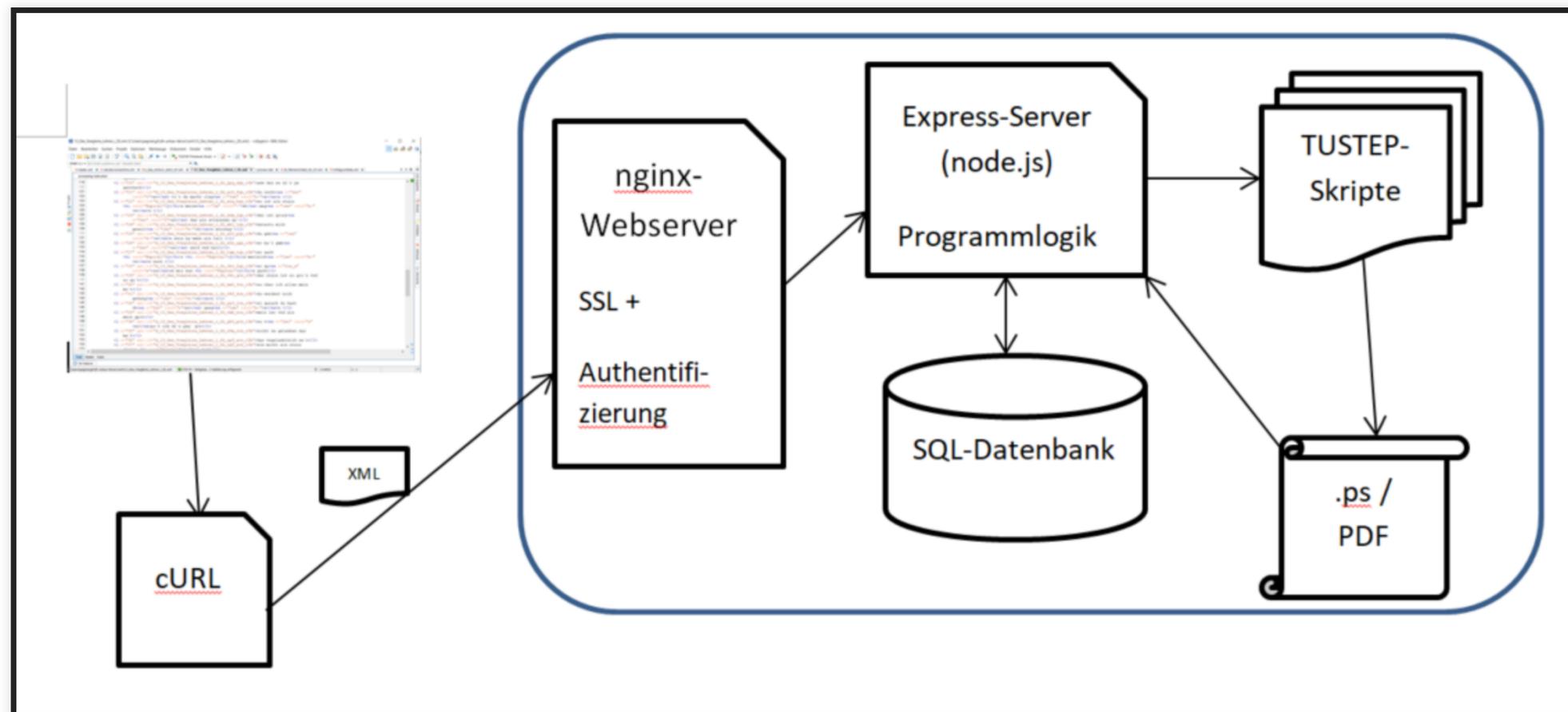
## Integration in Arbeitsumgebung

- Wahl einer guten TUSTEP-Preview-Lösung für ein konkretes Editions-Team hängt neben der Teamgröße auch ab von:
  - Welche Arbeitsumgebung wird verwendet? (TUSTEP-Editor, XML-Editor, XML-Datenbank, ...?)
  - Welche Möglichkeiten bietet die verwendete Arbeitsumgebung für die Integration von Drittsoftware?
  - Über welches Know-How verfügt das Team, das die Preview-Lösung aufsetzt und wartet?

## Vorstellung zweier konkreter Serverlösungen

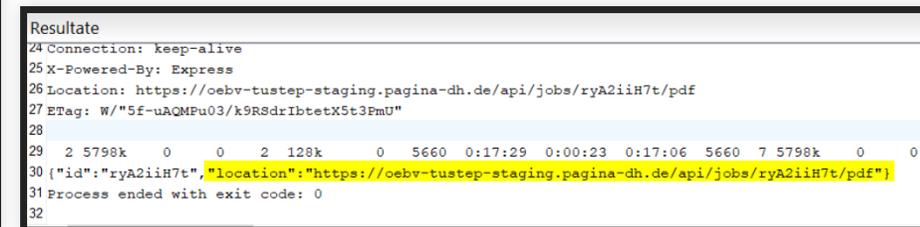
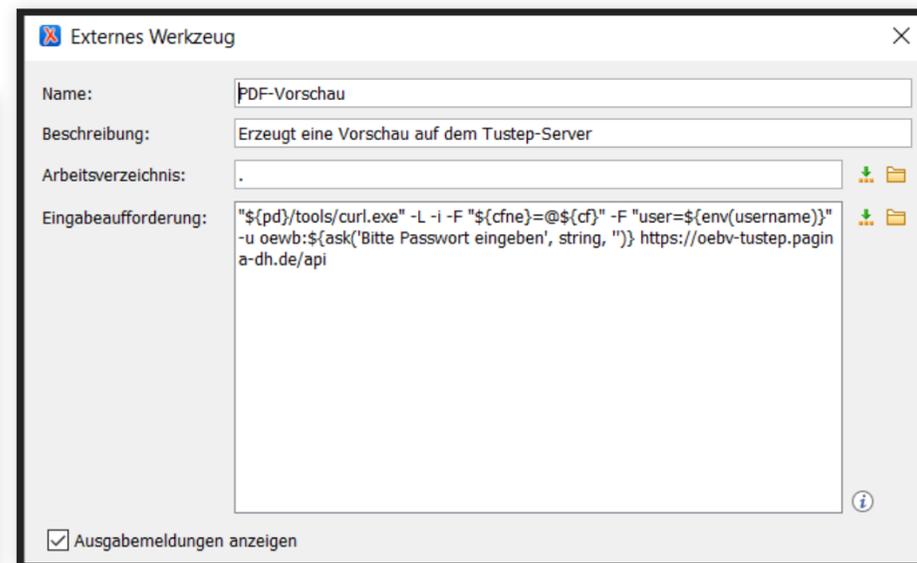
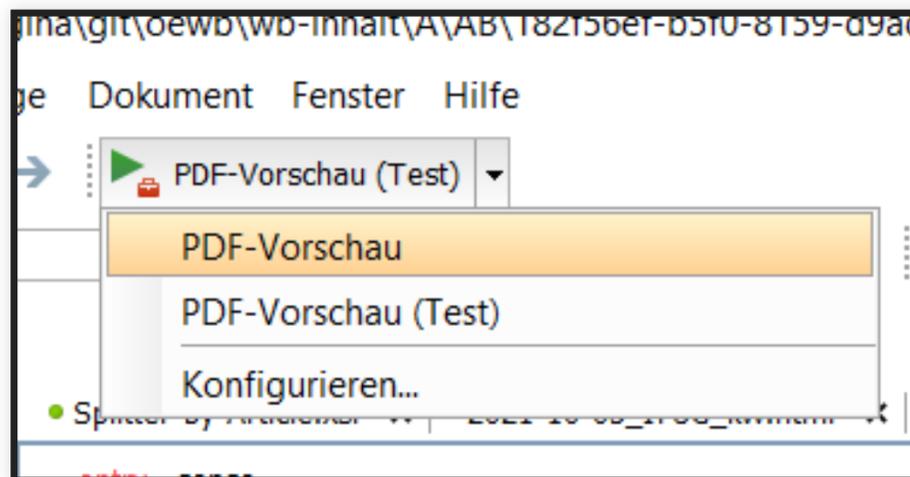
1. Webserver mit „Warteschlange“
2. eXist-Datenbank mit Cache-Funktionalität für Grafiken

# Architektur „Webserver mit Warteschlange“



## <oxygen/>-Integration

- Nutzung des <oxygen/>-Konzepts „externes Tool einbinden“
- Aus Anwender:innen-Sicht: Einfacher Button-Klick zum Auslösen des Previews, Ergebnis als URL
- Aus Admin-Sicht: Versand des aktuellen Dokuments mit cURL an den Server



## Server-Aufgaben

- Entgegennehmen von Preview-Aufträgen: Jeder Auftrag in SQL-Datenbank gespeichert
- Sequentielle Abarbeitung der noch nicht verarbeiteten Aufträge; aufräumen, wenn nötig
- Bereitstellen von Auftragslisten, erstellten PDFs, Logdateien

## Server-API

- /api/run: Nimmt einen Preview-Auftrag entgegen und speichert ihn in der DB
- /api/jobs: Liefert eine Liste aller in der Datenbank gespeicherten Jobs zurück (JSON-Format)
- /api/jobs/{id}/pdf: Liefert die PDF-Datei zu einem Auftrag zurück
- /api/jobs/{id}/log/{logtyp}: Liefert die TUSTEP-Logdatei oder weitere Debug-Ausgaben zu einem Auftrag

## Server: Scheduler

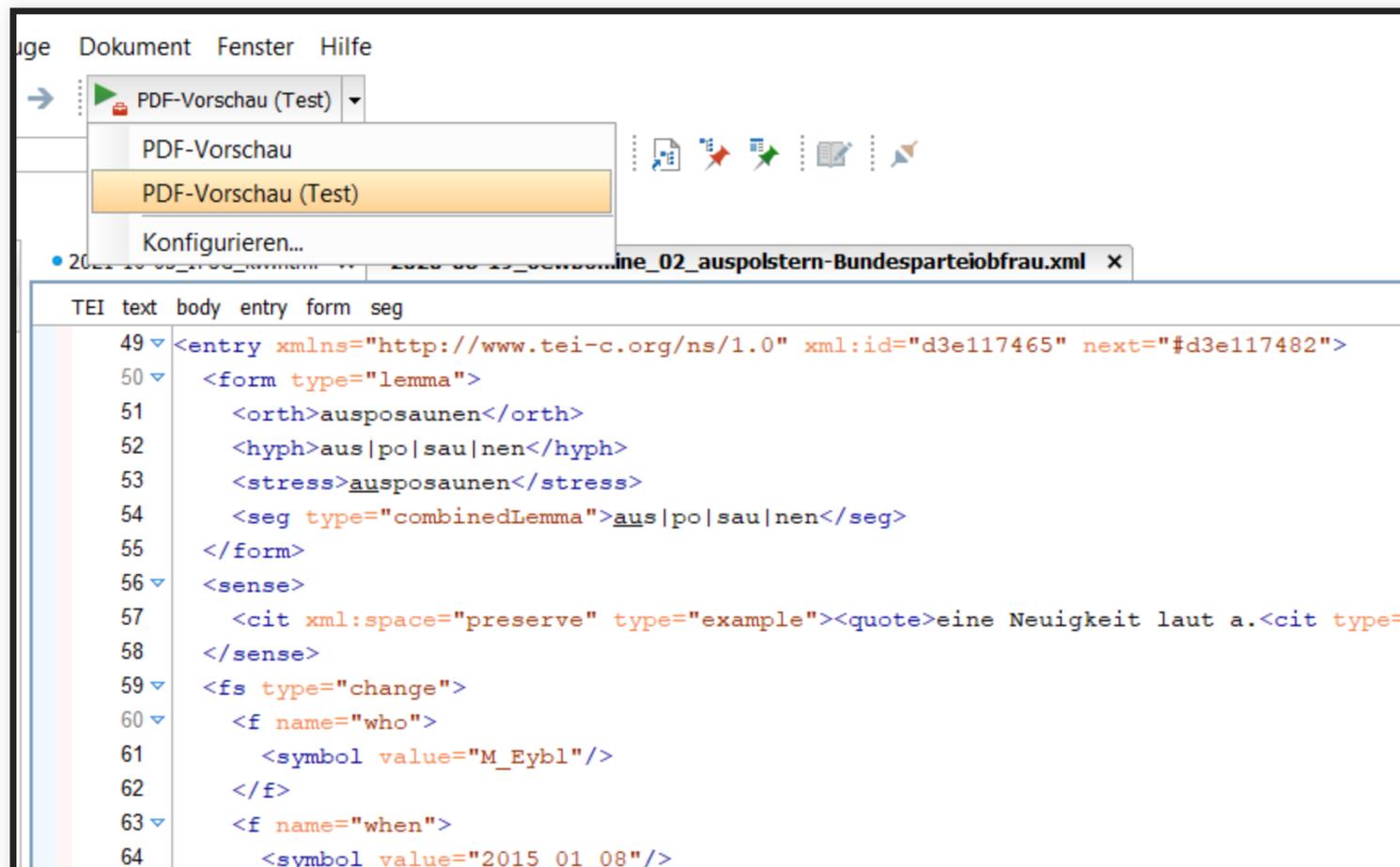
- Server prüft in konfigurierbarem Turnus (z. B. alle 5 sec.), ob noch unbearbeitete Aufträge vorliegen; wenn ja, wird die nächste Quelldatei an das TUSTEP-Skript übergeben
- Server prüft, ob Aufträge älter als konfigurierbares Datum vorliegen (z. B. älter als 2 Monate); wenn ja, werden diese aus der DB und vom Dateisystem gelöscht.

## Server: Umfang

- Die Serverfunktionalität, die auf dem node.js-Server express aufsetzt, ist sehr schlank programmiert: Es genügen 427 Zeilen Programmcode.
- Ein zusätzliches Frontend für die Bequemlichkeit: 268 Zeilen Programmcode (HTML + JavaScript / Vue.js, kein eigenes CSS)
- Skripte für Continuous Integration (automatisches Installieren der nötigen Komponenten auf einem Server): ca. 300 Zeilen

# Demo: Österreichisches Wörterbuch

Schritt 1: XML-Datei aus <oxygen/> heraus an Server schicken



The screenshot shows a software interface with a menu bar (Datei, Dokument, Fenster, Hilfe) and a toolbar. A dropdown menu is open, showing options like 'PDF-Vorschau' and 'Konfigurieren...'. Below the menu, a window titled '2015-01-08\_02\_ausposaunen-Bundesparteiofbrau.xml' is open, displaying XML code. The code is TEI-compliant and defines the word 'ausposaunen' with its orthography, hyphenation, stress, and a sense example. It also includes fields for author ('M\_Eybl') and date ('2015 01 08').

```
TEI text body entry form seg
49 <entry xmlns="http://www.tei-c.org/ns/1.0" xml:id="d3e117465" next="#d3e117482">
50 <form type="lemma">
51 <orth>ausposaunen</orth>
52 <hyph>aus|po|sau|nen</hyph>
53 <stress>ausposaunen</stress>
54 <seg type="combinedLemma">aus|po|sau|nen</seg>
55 </form>
56 <sense>
57 <cit xml:space="preserve" type="example"><quote>eine Neuigkeit laut a.</cit type="
58 </sense>
59 <fs type="change">
60 <f name="who">
61 <symbol value="M_Eybl"/>
62 </f>
63 <f name="when">
64 <symbol value="2015 01 08"/>
```

# Demo: Österreichisches Wörterbuch

## Schritt 2: Auftragsliste auf Website aufrufen

PDF	Dokument	Previewtyp	Empfangen	Start	Ende	Aktionen	Protokolle
	<b>ÖWB Elemente</b> auspolstern, ausposaunen, auspowern, auspowern, ... 2020-08-19_oewbonline_02_auspolstern- Bundesparteiobfrau.xml	einzeln	20. Sept. 2021, 8:35:33 beauftragt von Kai Weber	20. Sept. 2021, 8:35:35	20. Sept. 2021, 8:35:43		<input type="button" value="Previewer"/> <input type="button" value="Tustep"/> <input type="button" value="Debug"/>
	<b>ÖWB, Wörter, die mit „ab“ beginnen...</b> ab, Adverb:, ab, Abakus, ... ab_fest.xml	einzeln	17. Sept. 2021, 16:20:23 beauftragt von Kai Weber	17. Sept. 2021, 16:20:25	17. Sept. 2021, 16:20:27		<input type="button" value="Previewer"/> <input type="button" value="Tustep"/> <input type="button" value="Debug"/>

# Demo: Österreichisches Wörterbuch

## Schritt 3: PDF-Datei abrufen

000001 [redacted] oewb/probe 20.09.21 06:35

1

**ausipolstern:** ein Kästchen a. (*innen mit weichem Material auslegen*)

**ausipolsauinen:** eine Neuigkeit laut a. (*überall nachdrücklich verkünden*)

**ausipoiwern<sup>1</sup>** [-paʊ-] (engl.) → ausgepowert

**ausipoiwern<sup>2</sup>** [-po:v-] (fr.) (ugs., abw., veraltet): *ausbeuten*

**ausipräigen:** *prägen*; Münzen a. | ein ausgeprägter (*stark geformter*) Geschäftssinn *deutlich werden*; ihre Kreativität prägt sich in ihren Arbeiten aus; → ausgeprägt

**ausipreisen:** Waren a. (*mit Preischildern versehen*)

**ausipresisen:** Zitronen a.

**ausiprobieiren:** ein neues Gerät a. (*testen*)

**Ausipuff** der, -[e]s/-e: Bauteil, das bei Verbrennungsmotoren die Abgase

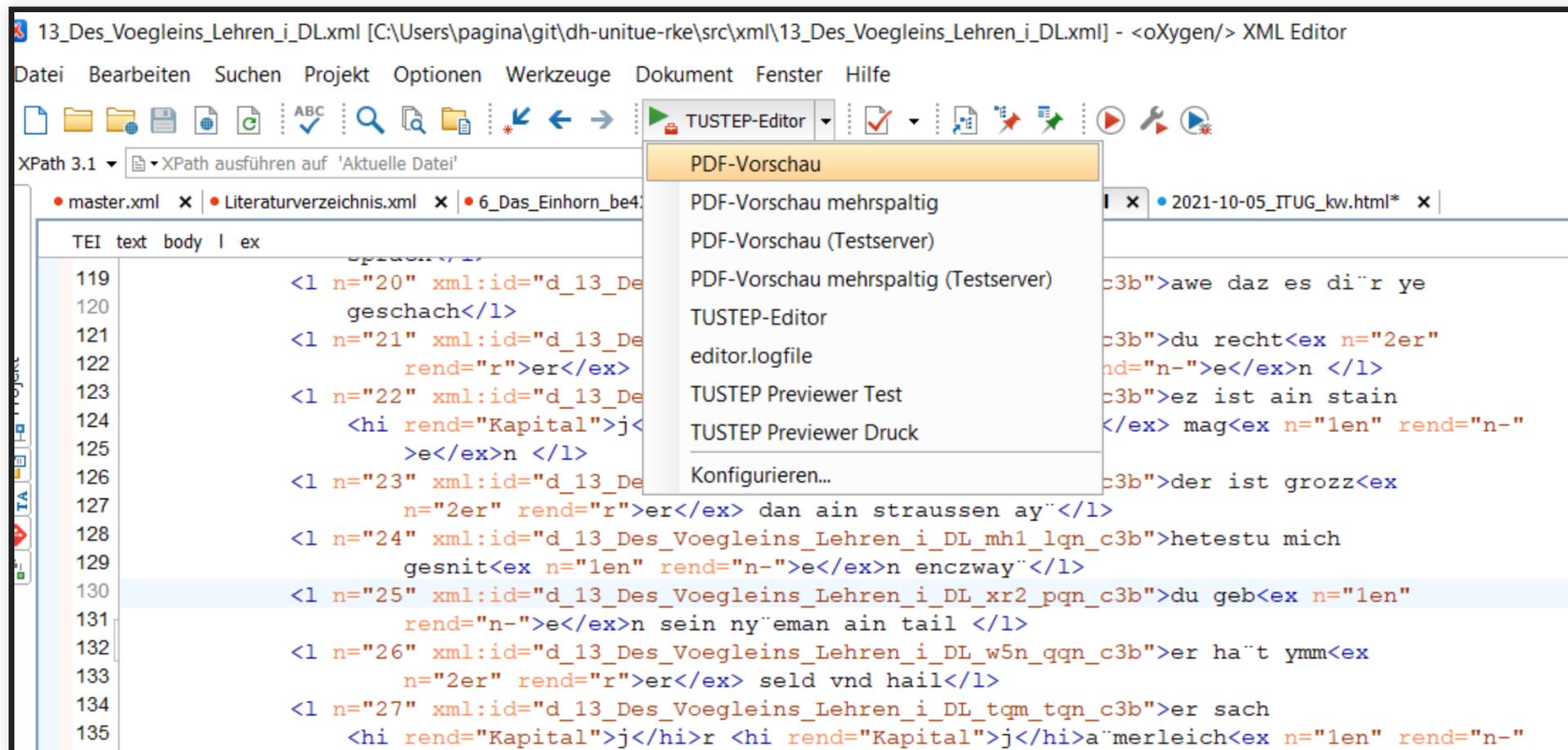
wegleitet || **Ausipuffigas** das, -es/-e

|| **Ausipuffrohr** -[e]s/-e || **Ausipufftopf** -[e]s, Auspufftöpfe



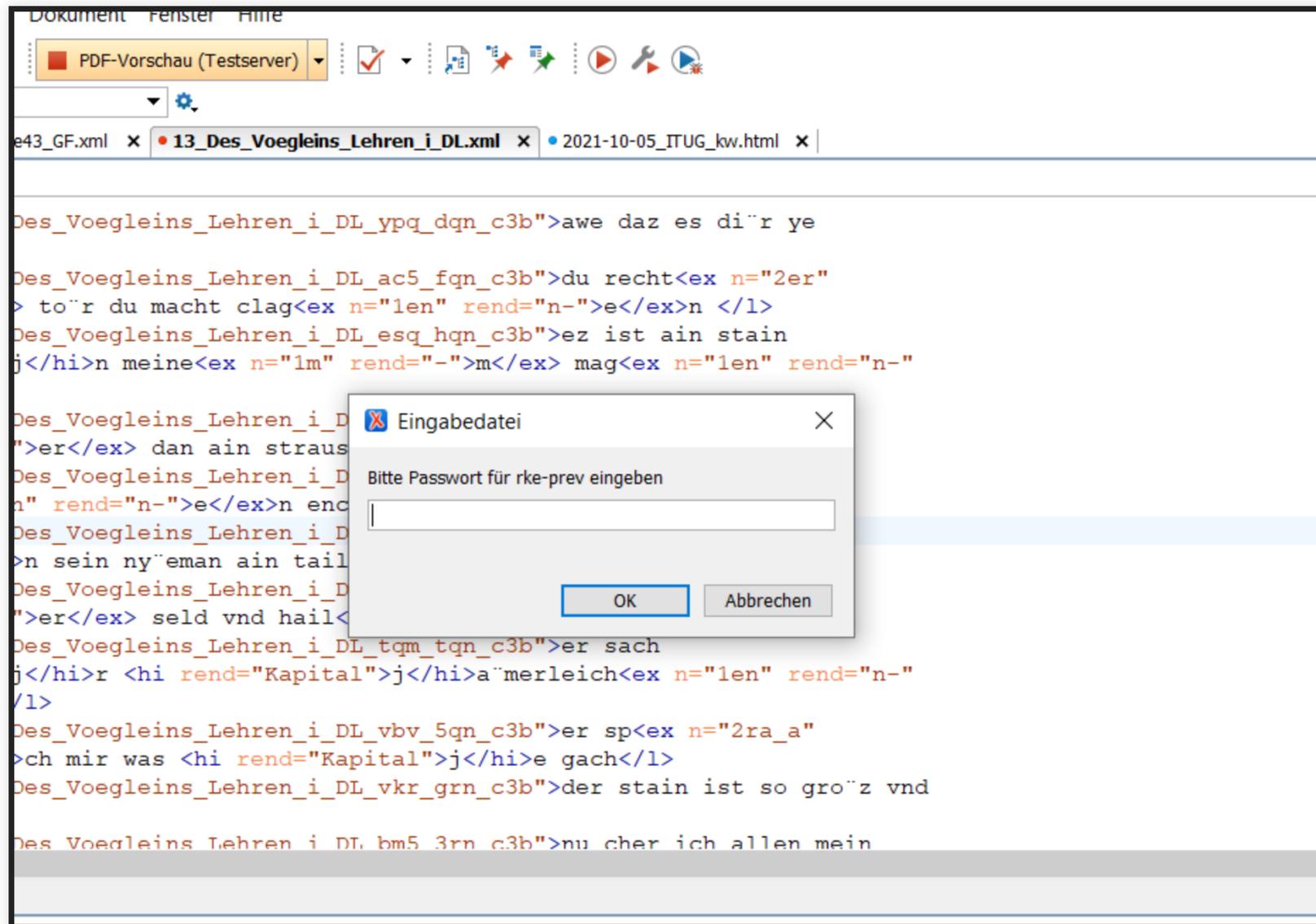
# Demo: Religiöse Kurzerzählungen...

## Schritt 1: XML-Datei aus <oXygen/> heraus an Server schicken



# Demo: Religiöse Kurzerzählungen...

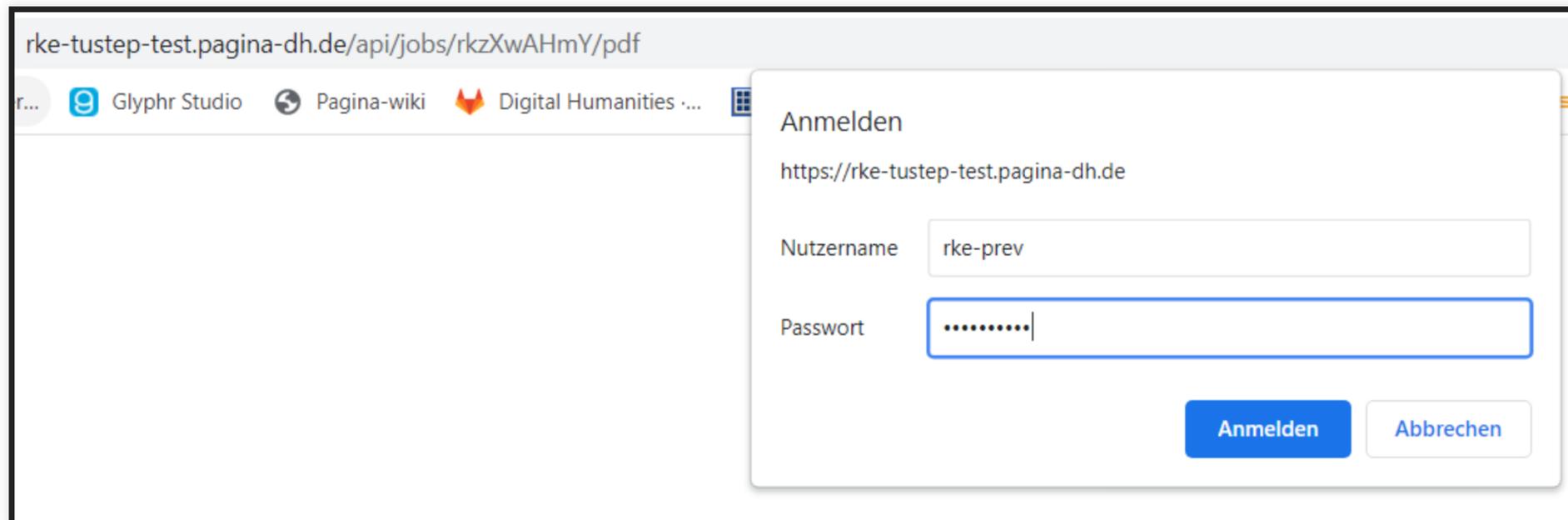
## Schritt 2: Passwort eingeben





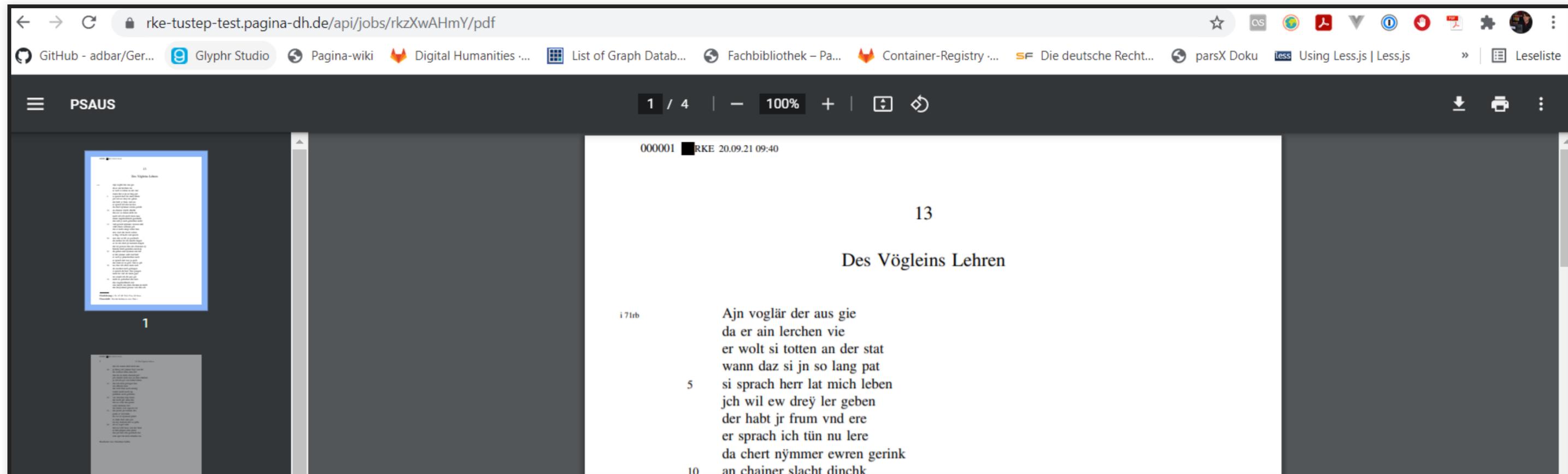
## Demo: Religiöse Kurzerzählungen...

Schritt 4: PDF-URL in Browser-Adresszeile kopieren, Passwort eingeben



# Demo: Religiöse Kurzerzählungen...

## Schritt 5: PDF ansehen



# Demo: Religiöse Kurzerzählungen...

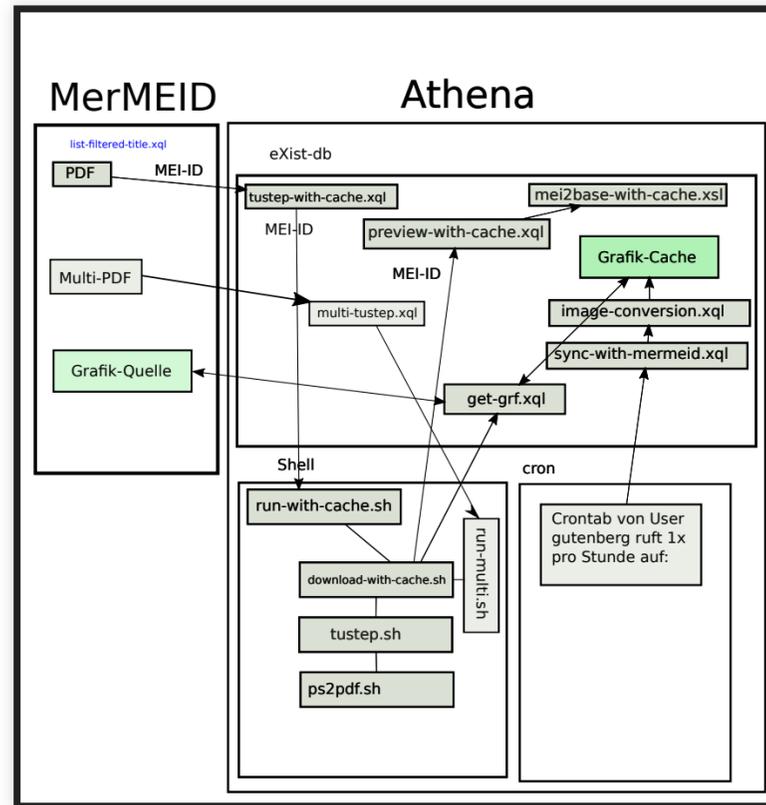
## Schritt 6: Auftrag (manuell) löschen

Browser: rke-tustep-test.pagina-dh.de/#/

Navigation: RKE Tustep Server, API-Dokumentation

PDF	Dokument	Previewtyp	Empfangen	Start	Ende	Aktionen	Protokolle
PDF	<b>Des Vögleins Lehren</b> #ma_dl 13_Des_Voegleins_Lehren_i_DL.xml	einzel	20. Sept. 2021, 11:40:41 beauftragt von Kai Weber	20. Sept. 2021, 11:40:45	20. Sept. 2021, 11:40:45		Previewer Tustep Debug
PDF	<b>Die Jakobsbrüder</b> #ma_dl 42_Die_Jakobsbrueder_bs_DL.xml	einzel	17. Sept. 2021, 13:16:22 beauftragt von Thomas Ziegler	17. Sept. 2021, 13:16:25	17. Sept. 2021, 13:16:26		Previewer Tustep Debug
PDF	<b>Warum Gott sein Haupt neigt</b> #ma_as 25_Warum_Gott_sein_Haupt_neigte_H_SFB.xml	einzel	17. Sept. 2021, 11:19:11 beauftragt von Thomas Ziegler	17. Sept. 2021, 11:19:15	17. Sept. 2021, 11:19:16		Previewer Tustep Debug

# Architektur „eXist-Datenbank mit Grafik-Cache“



# Datenhaltung

- Die Datenhaltung (XML-Dateien und Binärdaten wie z. B. Grafiken) erfolgt über eine eXist-Datenbank
- Der Datenbankserver stellt eine HTML-Übersicht aller Dokumente bereit.

# TUSTEP-Previews

- TUSTEP-Preview für Einzeldokumente durch Klick auf PDF-Link.
- TUSTEP-Preview für beliebige Dokumentlisten: Haken in Checkboxen, dann Klick auf PDF-Vorschau:

**Werklische der Bach-Söhne**

Cache zuletzt synchronisiert: 2021-09-20 12:52:48

alle auswählen

	MEI-ID	MEI	XML	PDF	Komponist	Sigel	Titel
	<input type="text"/>				CPEB ▾	<input type="text"/>	
<input checked="" type="checkbox"/>	23191608312707	<a href="#">mei</a>	<a href="#">xml</a>	<a href="#">pdf</a>	CPEB	<a href="#">A 1</a>	Sonate F-Dur
<input type="checkbox"/>	23191991972547	<a href="#">mei</a>	<a href="#">xml</a>	<a href="#">pdf</a>	CPEB	<a href="#">A 10</a>	Sonate B-Dur
<input checked="" type="checkbox"/>	23232995190881	<a href="#">mei</a>	<a href="#">xml</a>	<a href="#">pdf</a>	CPEB	<a href="#">A 100</a>	Rondo a-Moll
<input checked="" type="checkbox"/>	23233002005601	<a href="#">mei</a>	<a href="#">xml</a>	<a href="#">pdf</a>	CPEB	<a href="#">A 101</a>	Sonate A-Dur

## Interaktion eXist / TUSTEP

- eXist-Datenbank (DB) läuft auf einem Linux-Server
- Datenaggregationen und -transformationen können in der DB mit XQuery- oder XSLT-Skripten besorgt werden.
- aus der DB heraus lassen sich Shellskripte anstoßen, welche benötigte Dateien an die richtige Stelle kopieren, TUSTEP starten und später .ps-Dateien nach PDF konvertieren

## Motivation zum Cachen von Grafiken

- Problem bei Dokumenten mit vielen Grafiken: Konvertierung .tiff -> .eps -> .grf dauert lange
- große Wartezeiten für Anwender:innen der Website
- Annahme: Einmal in der Datenbank hinterlegte Grafiken werden eher selten geändert
- Verbesserungsidee: TIFF-Grafiken vorab nach GRF konvertieren und zwischenspeichern (Cache)
- Beim Previewauftrag prüfen, ob die Grafik im Cache noch aktuell ist; nur nicht aktuelle Grafiken neu konvertieren, sonst gecachte Version verwenden

## Aktualisierung Grafik-Cache

- einmal pro Stunde prüfen, ob in der Quell-DB neue oder geänderte Grafiken vorliegen
- neue Grafiken herunterladen, in einer Cache-DB speichern und dabei eine feste Nummer zuordnen
- neue / geänderte Grafiken via .eps nach .grf konvertieren und unter Dateinamen mit fester Nummer im Dateisystem speichern

## Verwendung gecachter Grafiken zum Preview-Zeitpunkt

- XML-Dokumente enthalten Referenzen auf ursprüngliche TIFF-Dateien
- vor der Übergabe der XML-Dokumente an TUSTEP werden die Original-Dateinamen über die Grafik-Cache-Metadaten in (numerische) GRF-Dateinamen übersetzt
- in der Regel sind aktuelle GRF-Varianten der Grafiken bereits vorhanden, so dass der Preview-Satz via TUSTEP direkt gestartet werden kann und nicht mehr auf Konvertierungen warten muss

# Demo: Bach-Werkverzeichnis

Dateiauswahl, Klick auf „Vorschau“

**Werkliste der Bach-Söhne**

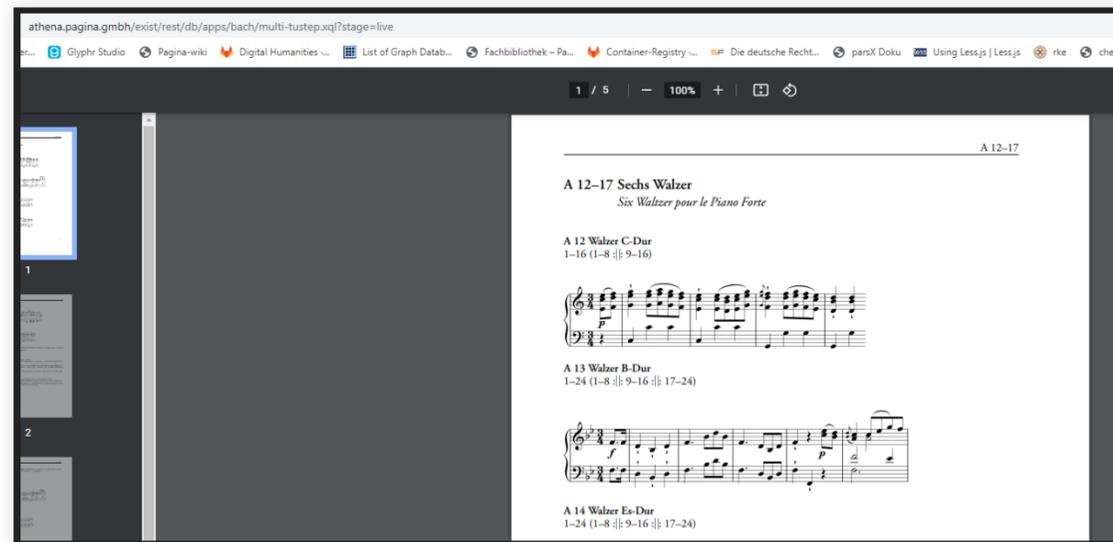
PDF-Vorschau Cache aktualisieren Zur Test-Umgebung Cache zuletzt synchronisiert: 2021-09-20 12:52:48

alle auswählen

	MEI-ID	MEI	XML	PDF	Komponist	Sigel	Titel
					CPEB		
<input checked="" type="checkbox"/>	23191608312707	<a href="#">mei</a>	<a href="#">xml</a>	<a href="#">pdf</a>	CPEB	<a href="#">A 1</a>	Sonate F-Dur
<input type="checkbox"/>	23191991972547	<a href="#">mei</a>	<a href="#">xml</a>	<a href="#">pdf</a>	CPEB	<a href="#">A 10</a>	Sonate B-Dur
<input checked="" type="checkbox"/>	23232995190881	<a href="#">mei</a>	<a href="#">xml</a>	<a href="#">pdf</a>	CPEB	<a href="#">A 100</a>	Rondo a-Moll
<input checked="" type="checkbox"/>	23233002005601	<a href="#">mei</a>	<a href="#">xml</a>	<a href="#">pdf</a>	CPEB	<a href="#">A 101</a>	Sonate A-Dur

# Demo: Bach-Werkverzeichnis

## Anzeige PDF-Datei im Browser



# Ende

Vielen Dank für Ihre Aufmerksamkeit!

Noch Fragen?