

# eXist-db und VueJS für dynamische UI- Komponenten

## Pohl, Oliver

opohl@bbaw.de  
Berlin-Brandenburgische Akademie der Wissenschaften,  
Deutschland

## Dogaru, Teodora

teodora.dogaru@bbaw.de  
Berlin-Brandenburgische Akademie der Wissenschaften,  
Deutschland

## Müller-Laackman, Jonas

jonas.mueller-laackman@bbaw.de  
Berlin-Brandenburgische Akademie der Wissenschaften,  
Deutschland

## Einführung

Für die Erstellung und Bearbeitung digitaler Editionen im Bereich der Digital Humanities werden vorzugsweise XML-basierte Lösungen verwendet. Eine der dabei am häufigsten benutzten Datenbanktechnologien ist eXist-db<sup>1</sup>. Dabei ist eXist-db nicht nur eine Lösung zum Bearbeiten und Speichern von XML-Daten, sondern bietet auch Möglichkeiten zur Programmierung von Webanwendungen auf Grundlage von XQuery. Im weiteren Programmierer\*innen-Umfeld sind X-Technologie-Lösungen sowohl für die Datenhaltung als auch für das Erstellen von Webanwendungen eher eine Nischenlösung<sup>2</sup>.

In den letzten Jahren werden verstärkt Frameworks wie Vue.js<sup>3</sup> oder React<sup>4</sup> für die Erstellung von Frontendmodulen in Webanwendungen verwendet.<sup>5</sup> Ein großer Vorteil solcher Frameworks gegenüber herkömmlichen Lösungen ohne Framework oder mit jQuery<sup>6</sup> ist die Möglichkeit, in sich geschlossene, reaktive User-Interface-Komponenten zu erstellen. Reaktiv heißt in diesem Fall, dass wenn sich das Datenmodell innerhalb einer Komponente ändert, ohne weiteren Aufwand sofort auch die Benutzeroberfläche entsprechend angepasst wird. Gleiches gilt umgekehrt, z.B. wenn eine Nutzerin ein Textfeld ausfüllt, auch das Datenmodell der Komponente sofort den entsprechenden Wert übernimmt, und so weitere Operationen im Hintergrund ausführen kann.

Da diese JavaScript-Frameworks jedoch mit JSON statt mit XML arbeiten, soll der hier vorgeschlagene Workshop eine Einführung in die Verwendung von Vue.js in Verbindung mit eXist-db geben. Im Workshop

soll konkret gezeigt werden, wie man mit eXist-db abgefragte Daten nach JSON serialisieren kann, um diese dann für die Erstellung einer Vue.js basierten, reaktiven Benutzeroberfläche einzusetzen. Grundsätzlich soll den Teilnehmenden hierdurch das Zusammenspiel zwischen einem beliebigen Backend - JSON-API - JavaScript-Frontend-Framework vermittelt werden.

## Workshop

Als praktisches Beispiel für den Workshop werden die Daten und der Quellcode von quoteSalute<sup>7</sup> zur Veranschaulichung benutzt. quoteSalute aggregiert Grußformeln aus verschiedenen digitalen Briefeditionen, bereitet diese auf, sodass NutzerInnen Briefabschlüsse von z.B. Alexander von Humboldt, Friedrich Schleiermacher oder Anna Gräfin von Lehndorff über einen Klick in ihre eigene E-Mail-Korrespondenz einfügen können. Die Daten dafür werden in einer eXist-db-Instanz als TEI-XML vorgehalten und die Benutzeroberfläche über Vue.js realisiert. Dabei kommuniziert die Benutzeroberfläche im Browser mit dem eXist-db-Backend, sodass auf Knopfdruck via XQuery in der Datenbank eine zufällige Grußformel ausgewählt und nach JSON serialisiert wird, sodass letztendlich die Nutzeroberfläche mit der neuen Grußformel angepasst werden kann. Um die passende Grußformel für die eigene E-Mail zu finden, können Nutzer\*innenden Korpus nach Höflichkeitsstufe (formell, informell), Sprache (derzeit deutsch, englisch, französisch, italienisch, spanisch, griechisch, latein) oder nach Geschlecht des Absenders und Empfängers filtern.

Sowohl der Datenkorpus, als auch der Quellcode sowie die Dokumentation sind als freie Software verfügbar.<sup>8</sup>

Der Workshop ist für zwei halbe Tage und für ca. 15-30 Teilnehmende ausgelegt. Dabei ist das Workshopprogramm in insgesamt vier Zwei-Stunden-Blöcke aufgeteilt, die inhaltlich aufeinander aufbauen. Im Folgenden werden die im Vorfeld des Workshops zu verteilenden Materialien, die zu erwartenden Voraussetzungen an die Teilnehmenden, sowie die Inhalte der einzelnen Blöcke und ihre Lernziele beschrieben.

## Voraussetzungen

Von den Teilnehmenden wird erwartet, dass sie grundlegende Kenntnisse in der JavaScript- und Web-Programmierung (HTML, CSS) haben. Weiterhin wird darum gebeten, einen Computer mitzubringen und vorab eXist-db (kostenfrei) sowie einen Texteditor zum Programmieren wie Visual Studio Code oder Atom zu installieren.

## Materialien

Vor dem Workshop stellen wir eine eXist-db-Applikation zusammen, die bereits alle benötigten Daten, Skripte und Datenbankabfragen enthält, um im Workshop sofort in die Arbeit einsteigen zu können, ohne selbst noch alles einrichten zu müssen. Die Teilnehmenden können dann diese App in ihrer eXist-db per Drag & Drop installieren. Eine entsprechende Anleitung wird bereitgestellt.

Sollte die Einrichtung der eXist-Instanz oder die Installation der App wider Erwarten nicht funktionieren, stellen wir auch eine eigene eXist-Instanz für den Workshop mit personalisierten Logins für die Teilnehmenden sowie eine JSON-REST-API bereit, sodass die weiteren Workshopinhalte in jedem Fall praktisch nachvollzogen werden können.

Des Weiteren bereiten wir ein HTML- und VueJS-Code-Skelett vor, sodass auf JavaScript-Seite wenig Zeit auf die Einrichtung der Programmierungsumgebung verwendet werden muss.

Die hier aufgelisteten Daten werden auf Github bereitgestellt und dokumentiert.

## Inhalte

### Block 1

Zu Anfang sollen die im Workshop verwendeten Tools und Konzepte, insbesondere das Zusammenspiel zwischen exist-db, JSON und Vue.js, vorgestellt werden. Es wird den Teilnehmenden noch etwas Zeit eingeräumt, die nötige Software herunterzuladen, zu installieren und ihre Entwicklungsumgebung einzurichten. Anschließend wird der Aufbau der quoteSalute-eXist-db-App inklusive des mitgelieferten Datenkorpus erklärt und gezeigt, wie innerhalb von eXist-db Daten abgefragt und ins JSON-Format serialisiert werden. Diese Schritte sind bereits voreingerichtet, sodass auch Teilnehmende ohne XQuery oder eXist-db-Kenntnisse den Ablauf nachvollziehen können. Aus praktischer Sicht werden die Teilnehmenden den quoteSalute-Grußformel-Korpus abfragen, eine zufällige Grußformel extrahieren und nach JSON serialisieren.

### Block 2

Im zweiten Block sollen die Teilnehmenden anfangs versuchen, die vorgegebenen Beispiele für Datenbankabfragen und JSON-Serialisierungen anzupassen bzw. zu erweitern, um beispielsweise nur Grußformeln, die an Frauen adressiert waren, aus dem Datenbestand herauszufiltern. So soll die Arbeit mit eXist-db und JSON-Serialisierung geübt und gefestigt werden.

Um die aus der eXist-db gelieferten JSON-Daten client-seitig verwenden zu können, muss das JavaScript-Frontend-Framework Vue.js mit der JSON-REST-API kommunizieren. Dafür soll zuerst in die Grundprinzipien von Vue.js (data, templates, model-view-binding, methods,

computed) anhand kleiner, aufeinander aufbauenden Code-Beispiele eingeführt werden.

### Block 3

In Block 3 sollen die in den ersten beiden Blöcken erarbeiteten Methoden und Abläufe miteinander in Verbindung gebracht werden. Über eine asynchrone Abfrage zum lokalen exist-db-Server (JavaScript-Skelett wird gestellt), soll die Vue.js-App eine neue Grußformel vom Server zu erfragen und diese dann im eigenen Datenmodell und so auch in der Benutzeroberfläche abbilden. Um asynchrone Operationen mit JavaScript nachvollziehen zu können, wird an dieser Stelle zusätzlich das Konzept von Promises vorgestellt und erläutert.

Die Teilnehmenden können die verbleibende Zeit in diesem Block dazu nutzen, ihre Vue.js-App zu programmieren und gegebenenfalls visuell anzupassen.

### Block 4

Sollte die Zeit für die Programmierung der Vue.js-App nicht genügen, dient der vierte Block als zusätzlicher Puffer. Für bereits fertige Teilnehmer und Interessierte bieten wir noch ein optionales Programm mit zusätzlichen Aufgaben an. So soll das Zusammenspiel zwischen eXist-db und Vue.js noch an einem weiteren, konkreten Fallbeispiel für die Digital Humanities geübt werden. Die Teilnehmenden erhalten einen Auszug aus einer digitalen Edition und sollen mit den vorgestellten Technologien ein Personenregister erzeugen. Durch diese Transferleistungsübung sollen die erlernten Vue.js-Methoden und Prinzipien gefestigt werden.

## Übersicht

Das Grundkonzept des Workshops lässt sich in folgender Tabelle zusammenfassen:

	Technologien & Konzepte	Lernziel
Block 1	Überblick eXist-db, VueJS & JSON	JSON-Serialisierung von XML-Daten mit eXist-db verstehen
Block 2	eXist-db, JSON, XPath, XQuery	eXist-db Datenbankabfragen modifizieren und Ergebnisse nach JSON serialisieren
VueJS	Grundkonzepte von VueJS verstehen und anwenden	
Block 3-4	VueJS, Asynchrone Anfragen, Promises	VueJS-App programmieren und an JSON-REST-API anbinden

Nach dem Workshop sollen die Teilnehmenden in der Lage sein

- nachzuvollziehen, wie man mit eXist-db XML-Daten nach JSON serialisiert,
- die Bedeutung von der Vue.js-Begriffe data, methods, model-view-binding und templates zu verstehen, anwenden und selbst erklären zu können,
- eine einfache, reaktive Vue.js-Applikation zu programmieren,
- diese Vue.js-Applikation an eine JSON-REST-Schnittstelle anzubinden, und
- das grundlegende Prinzip im Zusammenspiel von Backend - JSON-API - JavaScript-Frontend auch auf andere Technologien und Frameworks übertragen zu können.

## Technische Ausstattung

Für die Durchführung des Workshops wird ein Beamer (vorzugsweise mit VGA-Anschluss) sowie Internetzugang benötigt. Aufgrund der Länge des Workshops wären mehrere Mehrfachsteckdosen zum Aufladen der Geräte der Teilnehmenden wünschenswert.

## Fußnoten

1. <http://exist-db.org>
2. In ProgrammiererInnen-Umfragen von JetBrains und StackOverflow oder Programmier- und Datenbank-Beliebtheits-Indizes sind eXist-db, XQuery und verwandte Technologien nicht einmal genannt.  
<https://www.jetbrains.com/research/devecosystem-2018/>  
<https://insights.stackoverflow.com/survey/2018#technology>  
<https://www.tiobe.com/tiobe-index/>  
<http://pypl.github.io/DB.html>
3. <https://vuejs.org/>
4. <https://reactjs.org/>
5. <http://2016.stateofjs.com/2016/frontend/>  
<https://2017.stateofjs.com/2017/front-end/results>
5. <http://2016.stateofjs.com/2016/frontend/>  
<https://2017.stateofjs.com/2017/front-end/results>
6. <https://jquery.com/>
7. <http://quotesalute.net>
8. <https://github.com/telota/quoteSalute>