DHBW
Duale Hochschule
Baden-Württemberg
Heidenheim

Sommersemester 2023

Neue Datenbankkonzepte

M. Sc. Marvin Scharle

Einkaufszettel

RavenDB

Gruppennummer: 11

Kurs: WWI21 A

Namen:

Alexander Ilg

Isabelle Schanda

Jessica Siedersberger

https://github.com/IlgAlex/RavenDB-Einkaufszettel

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Projektvorstellung

1.1 Thema

Im Laufe dieses Projektes wird eine Webanwendung mit Schnittstelle und einer NoSQL-Datenbank als Datenspeicherung implementiert.
Bei der Webanwendung handelt es sich um einen Einkaufszettel, in dem die Artikel in Form einer Checkliste aufgelistet werden. Die für die Datenspeicherung verwendete NoSQL-Datenbank ist die RavenDB. Diese speichert beispielsweise die hinzugefügten Artikel und ob bei diesen Artikeln Haken gesetzt sind. Die Gründe für die Auswahl der NoSQL-Datenbank RavenDB wird im folgenden Text verdeutlicht.

Im Code gibt es eine README-Datei. Diese beinhaltet Erläuterungen, wie die Anwendung gestartet und die RavenDB inklusive Database heruntergeladen und installiert werden kann.

1.2 Anforderungen

Die Anwendung soll folgende Anforderungen erfüllen:

- Die Anwendung muss dem User die Möglichkeit bieten, durch Klick auf den Plus-Button ein neues Element hinzuzufügen.
- Die Anwendung muss dem User die Möglichkeit bieten, durch Klick auf das vorgesehene Feld bestehende Elemente abzuhaken.
- Die Anwendung muss dem User die Möglichkeit bieten, die gesamte Liste der angelegten Elemente abzuhaken.
- Das System muss fähig sein, nach Abhaken eines Artikels diesen in die Liste der abgehakten Artikel zu verschieben.
- Die Anwendung sollte dem User die Möglichkeit bieten, durch Klick auf den gesetzten Haken diesen wieder zu entfernen.
- Das System muss fähig sein, nach Entfernen des Hakens bei einem Artikel diesen Artikel in die Liste der angelegten Artikel zu verschieben.
- Die Anwendung muss dem User die Möglichkeit bieten, durch Klick auf den Papierkorb-Button bestehende Elemente zu löschen.
- Die Anwendung sollte dem User die Möglichkeit bieten, alle abgehakten Elemente der Liste zu löschen.

2. NoSQL-Datenbank: RavenDB

Zur Durchführung des Projekts wird eine NoSQL-Datenbank verwendet. Diese hat im Vergleich zur relationalen Datenbank folgende Vorteile.

- Die Abfragen bei NoSQL-Datenbanken sind schneller als bei SQL-Datenbanken.
- Die NoSQL-Datenbanken haben ein flexibles Datenmodel. Dies bedeutet, dass die Datenbanken schnell zu ändern oder zu optimieren sind (bei Anpassung der Anforderung).
- Es sind horizontale und vertikale Skalierungen möglich.
- Die NoSQL-Datenbanken fokussieren sich beim CAP-Theorem auf die Availability und auf die Partition tolerance und nicht auf die Consistency. Die Konsistenz der Daten über lange Zeit hinweg wird bei diesem Projekt nicht benötigt.
- Bei NoSQL-Datenbanken werden keine Relationen zwischen den Entitäten benötigt.

Bei diesem Projekt wird RavenDB als NoSQL-Datenbank ausgewählt und verwendet. Diese Entscheidung basiert auf folgenden Merkmalen.

- Bei diesem eher kleinen Projekt ist es nützlich, dass RavenDB durch die Community-Lizenz für den On-Premise-Einsatz kostenlos und auch die Software für die wichtigsten Plattformen beispielweise Windows, Linux und Docker kostenlos herunterladbar ist. Dadurch konnten wir mit RavenDB auf verschiedenen Plattformen arbeiten.
- RavenDB ist durch ein integriertes Dokumentendatenbank-Management-Studio mit grafischer Benutzeroberfläche (GUI) benutzerfreundlich und deshalb gut für dieses Projekt geeignet, da so eine erleichterte Bedienbarkeit gegeben ist.
- Es werden kostenlose Tutorials in GitHub und ausführliche Dokumentationen auf der Webseite zur Verfügung gestellt. Diese werden durch den Support einer großen Community ergänzt. Deshalb ist ein NodeJS-Client vorhanden, der direkt mit RavenDB kompatibel ist. Dadurch konnte sich das Team sehr gut mit dieser NoSQL-Datenbank vertraut machen und das Projekt erfolgreich durchführen.

- RavenDB ist eine dokumentenorientierte Datenbank. Dies bedeutet, dass die Daten im Ganzen in JSON-Dokumenten gespeichert werden und nicht jede Zeile einzeln. Dieser Datenbankenansatz ist für unser Projekt gut geeignet, da unsere Datenstruktur relativ einfach ist und keine Beziehungen zwischen den Datenobjekten bestehen.

3. Anleitung: Database in RavenDB anlegen

Ausgangslage für „Database anlegen“ ist die Installation und das Öffnen der RavenDB.

Man wechselt in den Menüpunkt „Databases“.

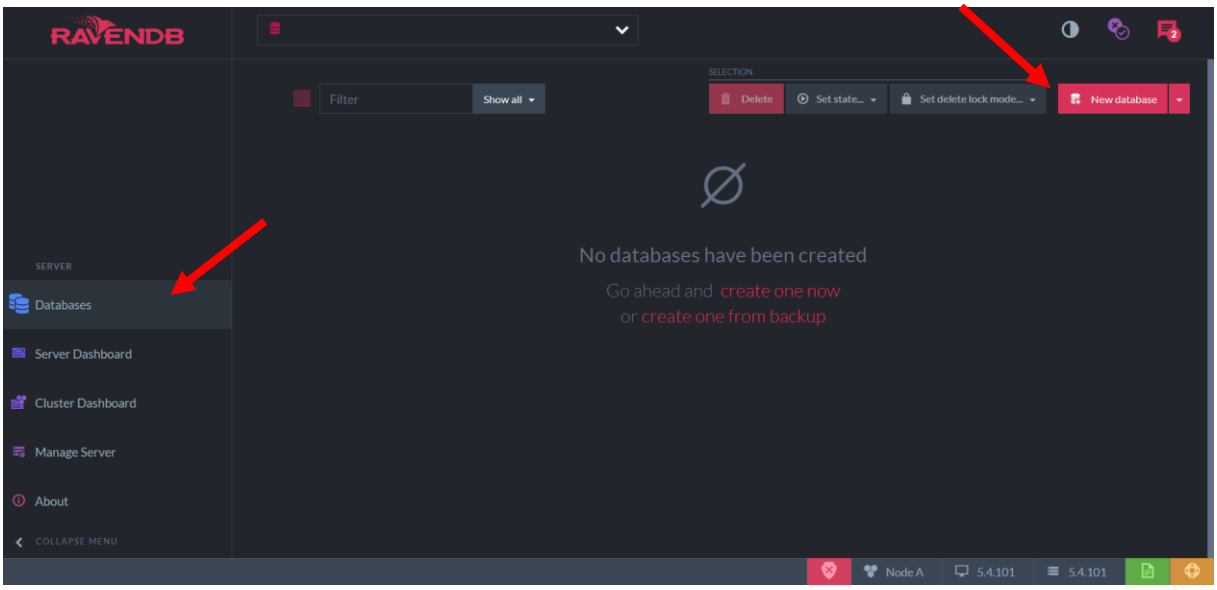

Abbildung 1: RavenDB Start.

Dort klickt man auf den roten Button „New database“ und es öffnet sich ein neues Fenster.

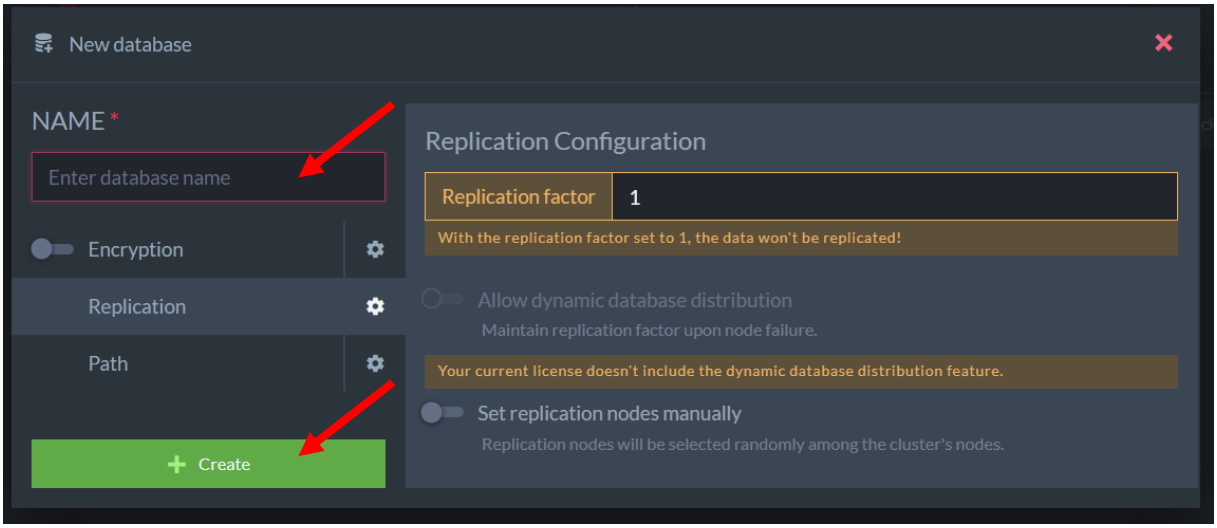

Abbildung 2: New database.

Dort gibt man der Database einen Namen (hier: *Einkaufszettel-DB*) und klickt anschließend auf den grünen Button„Create“.

Eine neue Database wurde angelegt.

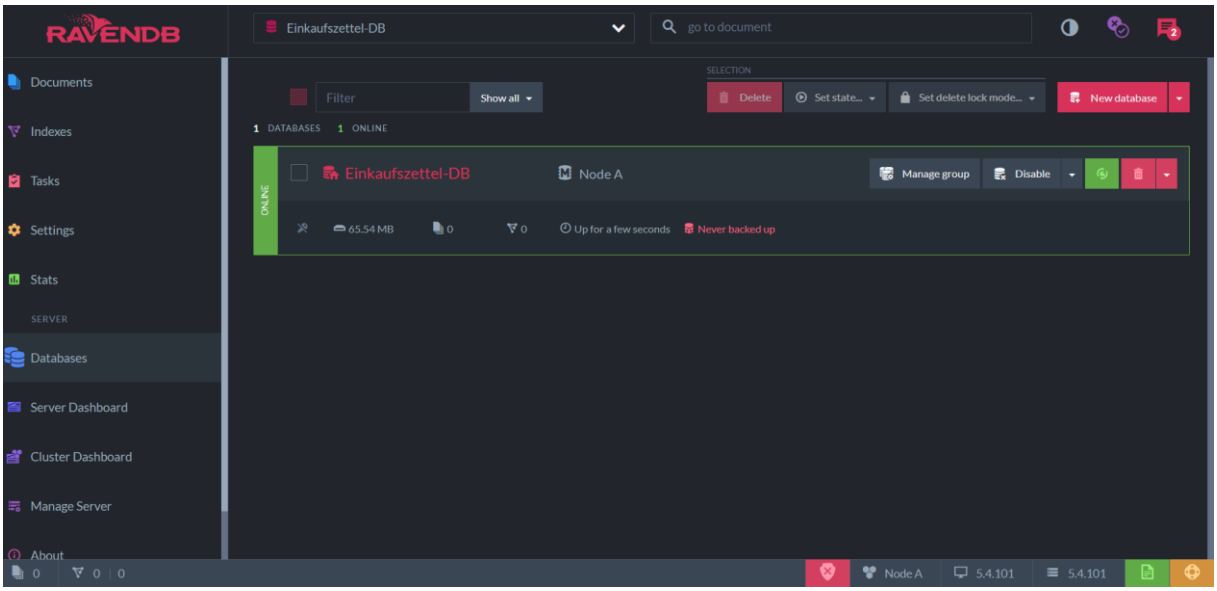

Abbildung 3: Neue Database angelegt.

Man besitzt nun eine (leeren) Database. Beim Import von Daten sollte die Database leer sein, da alle in der Database enthaltenen Daten überschrieben werden.

Man wechselt in die Database durch Klick auf den Namen der gewünschten Database (hier: *Einkaufszettel-DB*).

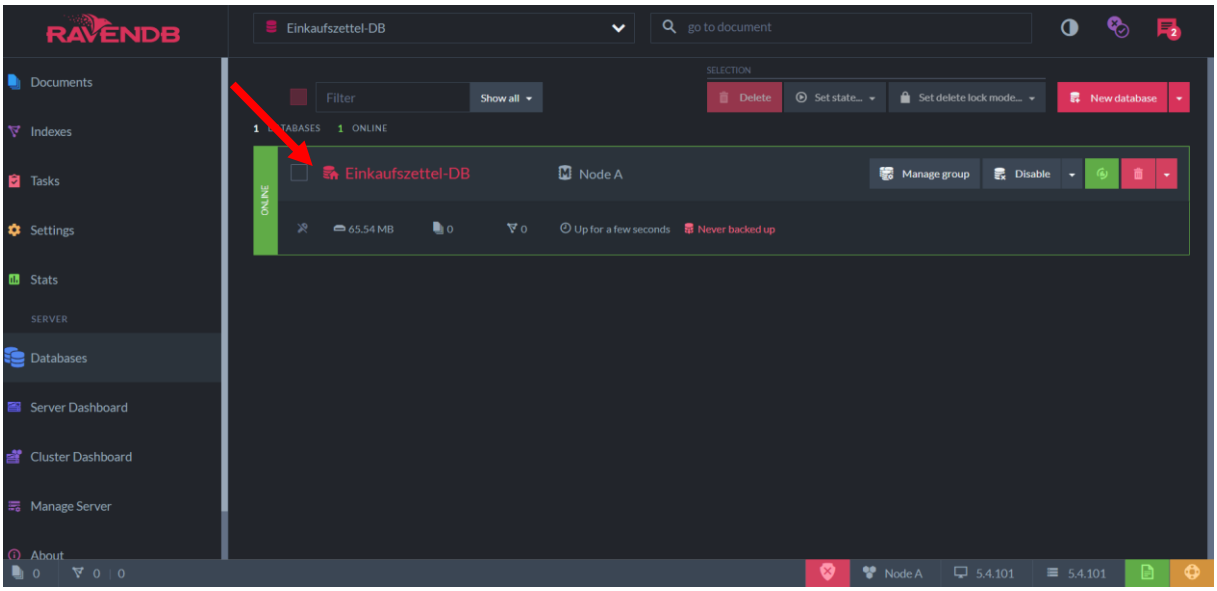

Abbildung 4: Beispieldaten importieren Start.

Es öffnet sich ein neues Fenster. Dort klickt man auf das rote Symbol der Tasks.

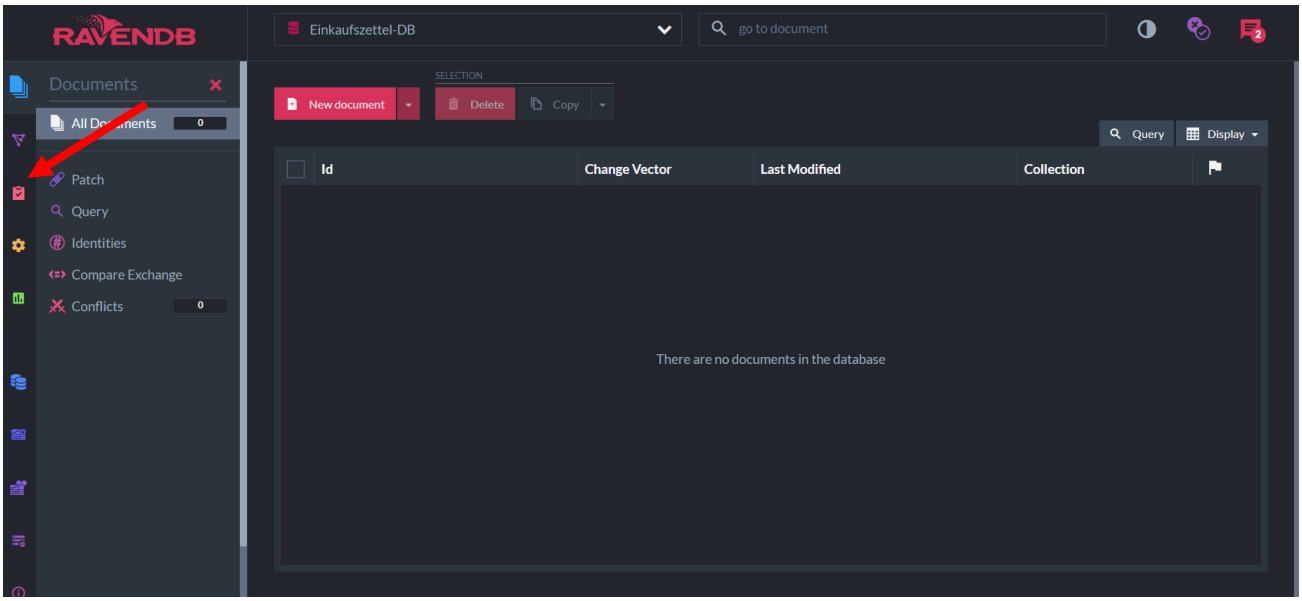

Abbildung 5: Database "Einkaufszettel-DB".

Danach klickt man im Menü auf der linken Seite auf „Import Data“.

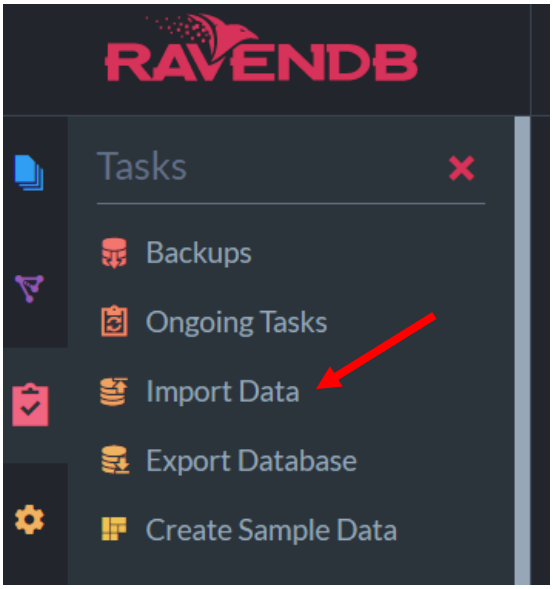

Abbildung 6: Menü Tasks.

Daraufhin öffnet sich ein neues Fenster. Dort gibt man die gewünschte Datei, in der die zu importierenden Daten gespeichert sind, durch Klick auf „Browse“ an (hier: mitgeschickte Datei „*Einkaufszettel-DB.ravendbdump*“).

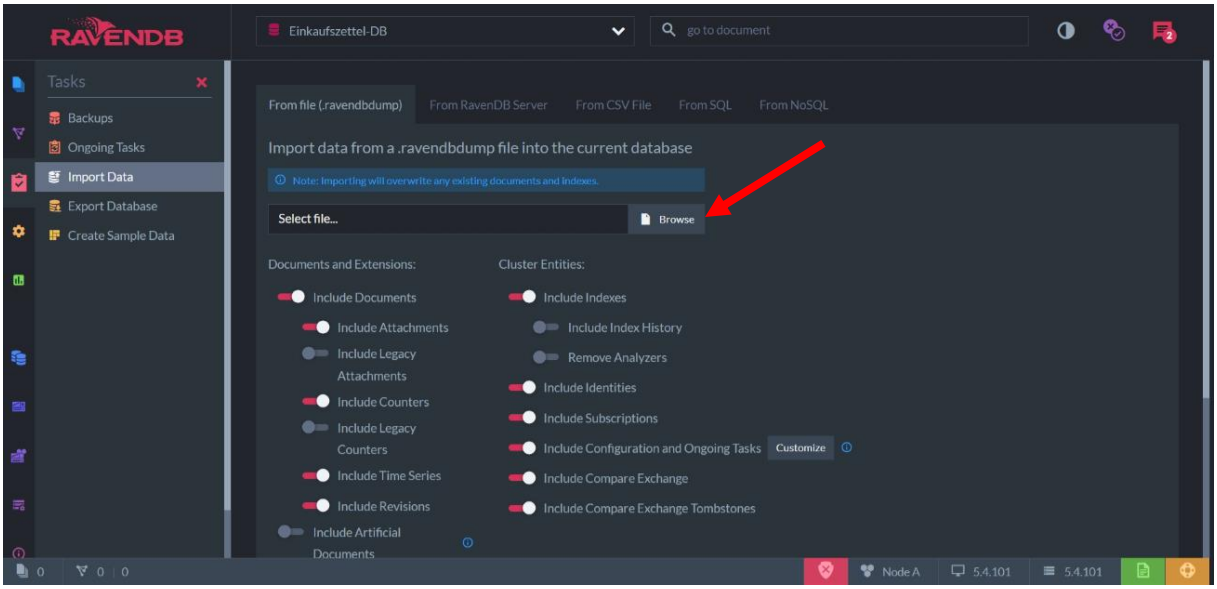

Abbildung 7: Daten importieren oben.

Danach klickt man auf den roten Button „Import Database“.

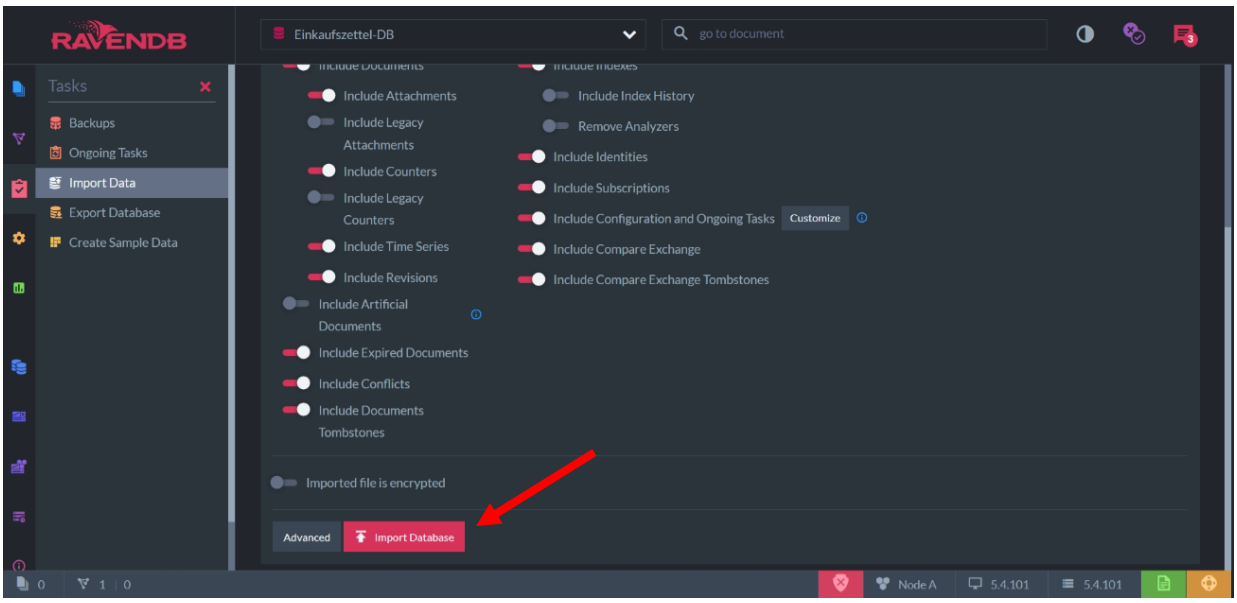

Abbildung 8: Daten importieren unten.

Es öffnet sich eine Übersicht des Datenimports.

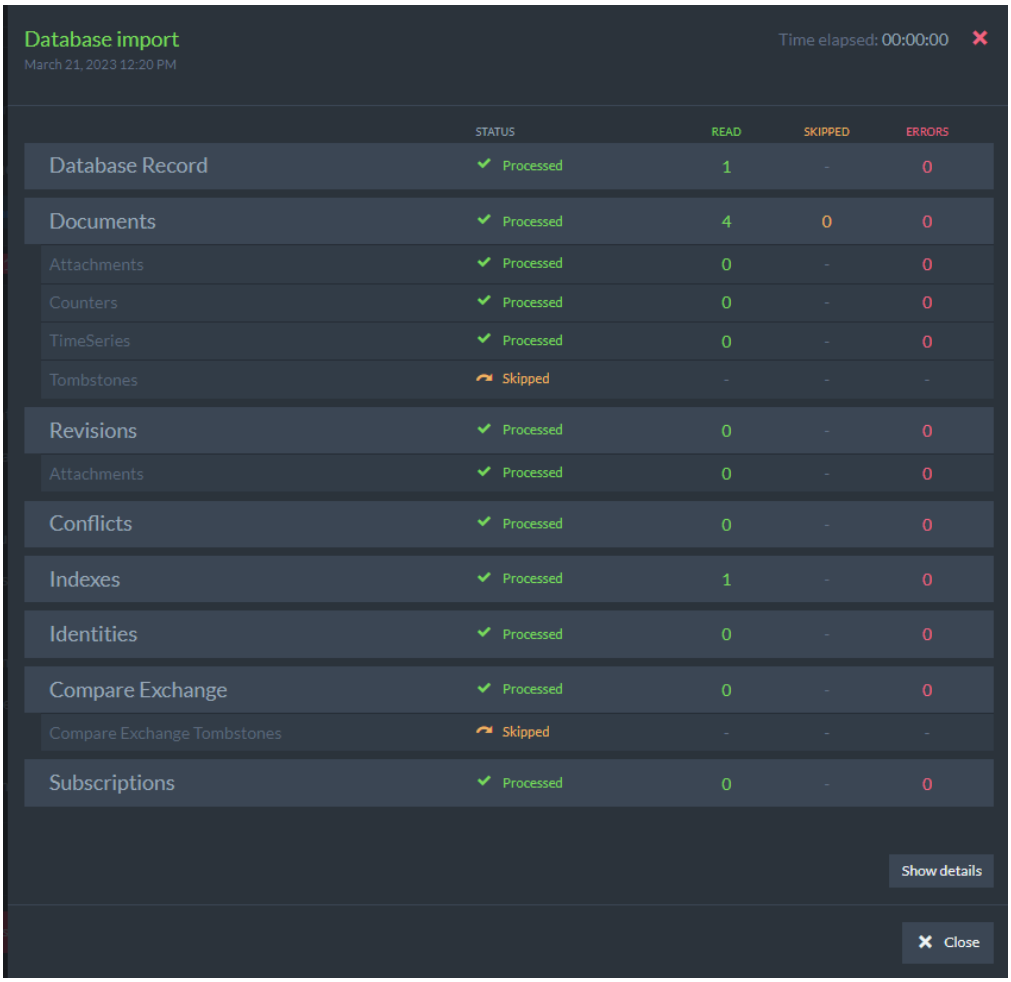

Abbildung 9: Datenimport.

Bei Klick auf den Button „Close“ ist der Importvorgang abgeschlossen und die Database mit den Daten gefüllt.