

Bookr - Dokumentation

Einleitung

Bookr ist ein Dienst, der ermöglicht, Bücher zu suchen und nähere Informationen zu ihnen zu bekommen. Es können nach unterschiedliche Feldern, wie zum Beispiel ISBN, Autoren oder Titel, gesucht werden.

Das entworfene Bookr-Projekt besteht aus einer Datenbank, einer API und mehreren Clients. Im Laufe der Arbeit wurden ein Web, iOS und Desktop-Client entworfen und implementiert.

Architektur

Die vorhandenen AWS¹-Komponenten sind mehrere Node.js-Server, eine MongoDB-Datenbank und ein Loadbalancer.

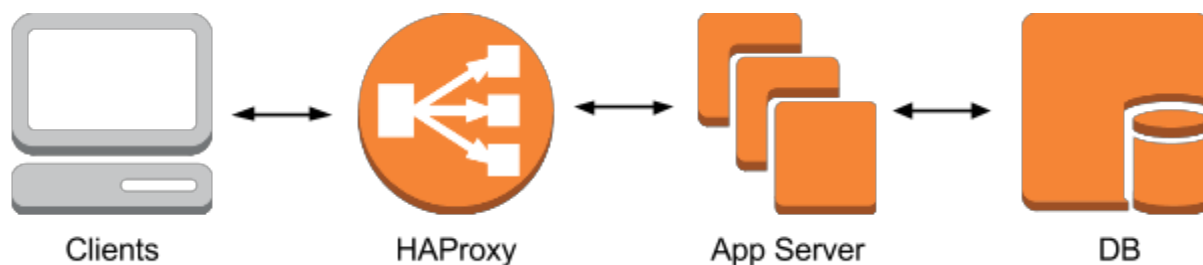


Fig1: Kombination der genutzten AWS-Komponenten

Loadbalancer

Der HAProxy²-Loadbalancer wird im API-Server-Stack gestartet. Alle Anfragen der Clients werden an die Elastic-IP des Loadbalancers gesendet. Dieser verteilt die Anfragen auf die API-Server. Je mehr Nutzer die Anwendung nutzen, umso mehr API-Server können gestartet werden, um die Last zu verteilen.

Node.js-API-Server

Als Software für den Server wird Node.js³ in Kombination mit Express⁴ eingesetzt. Node.js wurde gewählt, da bereits Erfahrungen in JavaScript vorhanden waren und durch die nicht blockierende Architektur eine performante Anwendung geschaffen werden kann.

¹ Amazon Web Services

² <http://haproxy.1wt.eu/>

³ <http://nodejs.org/>

⁴ <http://expressjs.com/>

Der Server hat die Aufgabe einkommende Anfragen zu evaluieren und an weitere Software-Komponenten zu leiten. So wird bei Suchanfragen zuerst nach einem Ergebnis in der Datenbank gesucht. Sollte kein Resultat gefunden wird die Webcrawler-Komponente gestartet. Ergebnisse dieser Komponente werden in der Datenbank gespeichert und als Antwort zurückgegeben.

Die implementierten Webcrawler bieten eine Schnittstelle, über die ein Suchwort genutzt werden kann, um auf mehreren externen Diensten, Suchen nach Büchern zu starten. Zu diesen Diensten gehören Google Books API⁵, ISBNdb⁶ und Open Library⁷. Nachdem Resultate aller Anfragen eingetroffen sind, werden die für die Weiterverarbeitung normalisiert und kombiniert.

Ergebnisse von Google Books API und ISBNdb umfassen nur einzelne Bücher. Open Library bietet die Möglichkeit für ein Buch verschiedene ISBN-Nummern aufzulisten. Diese Information wird genutzt, um Resultate der anderen APIs zu kombinieren und Auflagen eines Buches zuzuordnen.

Der Mehrwert der Vorgehensweise besteht darin, dass der Endnutzer besser gruppierte Ergebnisse mit weniger Duplikaten hat.

Anfragen an externe Server sind langsam und sollten vermieden werden.

Durch den implementierten Ablauf, wird bei vielen verschiedenen Suchanfragen die Datenbank wahrscheinlicher Ergebnisse finden und weniger auf die Webcrawler zurückgreifen müssen. Somit kommt der Endnutzer schnell an Ergebnisse, da die Verbindung mit der Datenbank schnell ist.

⁵ <https://developers.google.com/books/>

⁶ <http://isbndb.com/>

⁷ <https://openlibrary.org/>

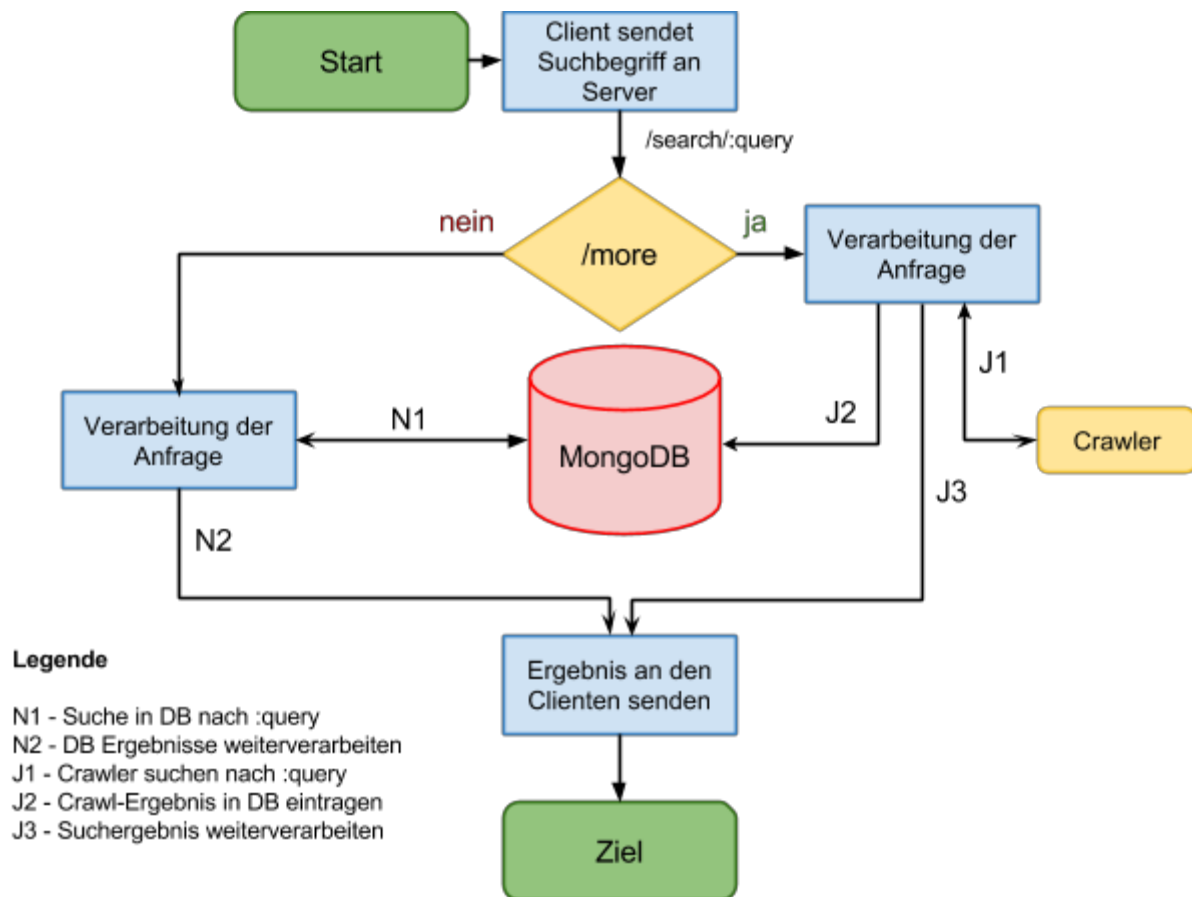


Fig2: Ablauf der Verarbeitung einer API-Anfrage zum Suchen eines Buches.

MongoDB

Als Datenbank-System wurde MongoDB⁸ gewählt. MongoDB ist eine dokumentorientierte, schema-freie Open-Source-Datenbank.

Die Wahl der Datenbank ist auf MongoDB gefallen, da das Datenmodell keine Relationen erwartet und MongoDB bereits mit Daten als JSON-Objekte arbeiten kann.

Das Datenmodell besteht aus Auflagen, bzw "Version" eines Buches und einem groben Buch-Objekt, bzw. "SuperBook". Hier wurde eine Relation mit einem Fremdschlüssel aufgebaut, da eine "Version" einem "SuperBook" zugewiesen ist.

⁸ <http://www.mongodb.org/>

Model

SuperBook

Wie bereits erwähnt stellt das SuperBook ein Buch dar, welches verschiedene Auflagen hat. Das SuperBook hat weniger Felder, als deren Versionen, da von Auflage zu Auflage unterschiedliche Publisher, Umschlagtexte oder Buchcover vorhanden sind.

Es besteht aus einem eindeutigen Identifier, der durch das Ergebnis der Open Library Schnittstelle gesetzt wird.

Attribut	Inhalt
_id	eine eindeutige ID
year	Jahr der Ersterscheinung
title	Titel des Buches
subtitle	Untertitel des Buches
authors	Autoren des Buches
isbns	Liste aller ISBNs, als ISBN 13 und 10 Tupel, die als Auflagen bekannt sind

Version

Die Version eines SuperBooks entspricht einer Auflage. Über das “superBook”-Feld wird eine Relation zum dazugehörigen SuperBook hergestellt.

Coverbilder der Versionen werden durch die Google-API zur Verfügung gestellt.

Attribut	Inhalt
superBook	Referenz auf das zugehörige SuperBook
year	Erscheinungsjahr der Auflage
title	Titel der Auflage

subtitle	Untertitel der Auflage
authors	Autoren der Auflage
publisher	Herausgeber dieser Auflage
textsippet	Textauszug des Buches
thumbnail	Link auf Auflagencover in unterschiedlichen Größen

REST-Schnittstelle

Der Server bietet verschiedene Schnittstellen an, über den die Clients Daten abfragen können. Das Datenformat für Antworten ist JSON.

```
GET /book/:_id
returns Object<SuperBook>
```

Gibt ein SuperBook zurück, wenn die ID in der Datenbank gefunden wurden, ansonsten ein leeres Objekt.

```
GET /book/version/:isbn10-isbn13
returns Object<Version>
```

Gibt eine Version zurück, wenn ISBN10 und ISBN13 in der Datenbank gefunden wurden. Wenn die Datenbank keinen Eintrag hat, werden die Crawler gestartet und suchen nach den ISBN-Nummern.

Wenn die Crawler ein Ergebnis finden, wird es als Version abgespeichert und zurückgegeben, ansonsten ein leeres Objekt.

```
GET /search/:query
returns Array<SuperBook>
```

Gibt ein Array von SuperBooks zurück, wenn der String im Query, im Index eines Buches vorhanden ist, ansonsten ein leeres Array.

```
GET /search/:query/more
returns Array<SuperBook>
```

Gibt ein Array von SuperBooks zurück, die über die Crawler, beim Suchen nach dem String in Query, gefunden wurden. Wenn kein Buch gefunden wurde, wird ein leeres Array zurückgegeben.

Clients

Da die Clients der Bookr-API die gleichen Daten erhalten, wird hier eine Übersicht zur verschiedenen Anzeige auf den unterschiedlichen Geräten und Systemen gegeben.

Webclient

Ember.js⁹ JavaScript-Anwendung welche per AJAX auf die API zugreift und Ergebnisse rendert.

Bookr

Masters of Doom

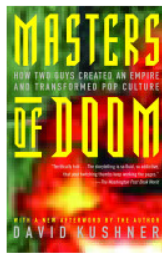
1 Ergebnisse

[Masters of Doom \(2003\)](#) David Kushner

Weitere Bibliotheken durchsuchen

Bookr

Masters of Doom



Masters of Doom (2004)

How Two Guys Created an Empire and Transformed Pop Culture

Autoren

David Kushner,

Publisher

Random House LLC

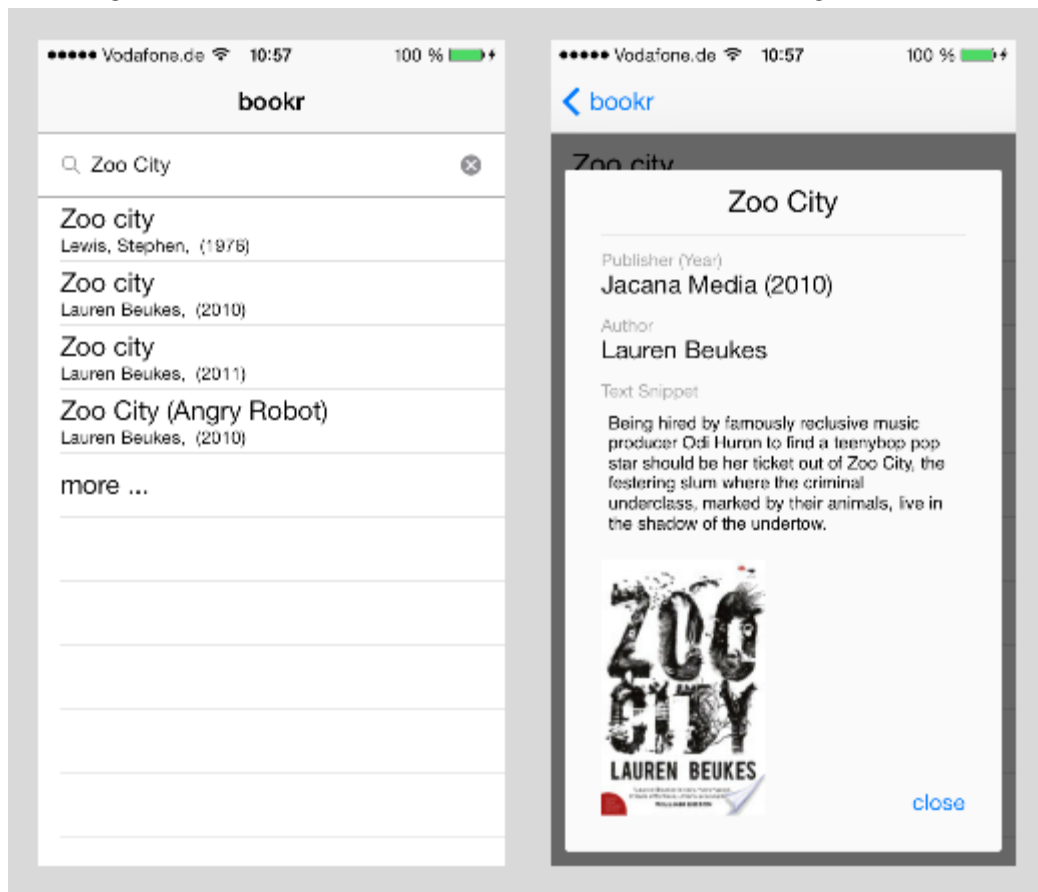
ISBN 10

0812972155

⁹ <http://emberjs.com/>

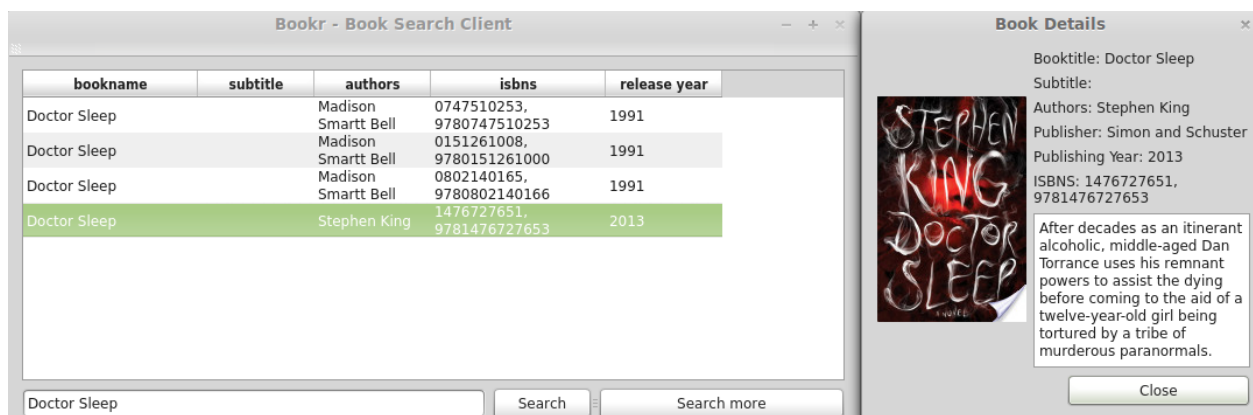
iOS-Client

iOS-Anwendung, die mithilfe des Restkit-Frameworks auf die API zugreift.



Desktop-Client

C++-Anwendung, mit Qt5 gebaut und lauffähig auf Linux, Mac und Windows.



Git Repositories

Bookr-iOS	https://github.com/romair/bookr-client-ios
Bookr-Desktop	https://github.com/christian-wieden-1314-fhb/bookr-desktop-client
Bookr-Web	https://github.com/makepanic/bookr-web
Bookr-API	https://github.com/makepanic/bookr
Bookr-Crawler	https://github.com/makepanic/bookr-crawler
Chef-Recipes	https://github.com/makepanic/opsworks-cookbooks/tree/bookr
Bookr-AWS-CLI	https://github.com/makepanic/bookr-aws

Travis-ci Repositories

Bookr-iOS	https://travis-ci.org/romair/bookr-client-ios/
Bookr-Desktop	https://travis-ci.org/christian-wieden-1314-fhb/bookr-desktop-client
Bookr-Web	https://travis-ci.org/makepanic/bookr-web
Bookr-API	https://travis-ci.org/makepanic/bookr
Bookr-Crawler	https://travis-ci.org/makepanic/bookr-crawler

Installation via CLI-Tool

Bookr-AWS-CLI clonen und via nodejs ausführen. Über Kommandozeilenparameter können bestimmte Funktionen aktiviert oder deaktiviert werden. Mit --all werden alle Aktionen ausgeführt. Vorher müssen die AWS-Credentials in die Konfigurationsdatei (aws-credentials.json) gepackt und weitere Einstellungen in der aws-setup.json vorgenommen werden.

Um ein Entwicklungssystem bereitzustellen, wird zusätzlich eine Vagrant box gegeben, die alle Anforderungen stellt ohne Pakete auf dem lokalen System installieren zu müssen.

Für mehr Informationen, die README im Repository anschauen.

`bookr-aws [master●] » node aws-setup.js --help`

Bootstraps AWS environment for bookr components. Version 1.0

Options:

<code>--all</code>	run all tasks
<code>--create-db-sec</code>	creates the database security group
<code>--create-web-sec</code>	creates the web security group
<code>--launch-db</code>	launches database instance
<code>--ops-api</code>	adds opsworks configuration for api stack, layer
<code>--ops-api-instance</code>	adds opsworks configuration for api instance
<code>--lb-api</code>	adds opsworks HAProxy configuration for api
<code>--launch-api</code>	launches api instance

--launch-lb	launches loadbalancer instance
--add-api-db-sec	adds the api ip to database security group
--deploy-api	deploys api application to api instance
--ops-web	adds opsworks configuration for webclient
--launch-web	launches web instance
--deploy-web	deploys web application to web instance

Das Tool geht alle Aufgaben nacheinander durch und am Ende (sollte) alles erfolgreich eingerichtet sein. --all ist dafür gedacht, auf einem komplett vanilla System alle Komponenten einzurichten und zu starten.