

OTH Amberg-Weiden, Fakultät EMI, Studiengang Medieninformatik

Web-Anwendungsentwicklung

Studienarbeit im Sommersemester 2020

Prof. Dr. Ulrich Schäfer

2. Juni 2020

1 Aufgabenstellung

Auf Basis von NodeJS und Apache Solr auf der server-Seite sowie jQuery, jQueryUI und – je nach Geschmack – weiteren Webframeworks auf der client-Seite soll eine interaktive Webanwendung zur Suche und Aufnahme-Programmierung im "seriellen" digitalen TV-Programm (DVB; Digital Video Broadcasting – europ. Standard für Digitalfernsehen) erstellt werden.

Datengrundlage der Anwendung sind EPG-Daten über ca. 6 Wochen im XMLTV-Format¹. Diese wurden von mir bereits in ein Import-Format für Apache Solr transformiert. Die Anwendung soll selbstverständlich auch für andere Zeiträume funktionieren.

Die Webanwendung soll es ermöglichen, nach Stichworten in Titeln, Untertiteln und Beschreibungen von TV-Sendungen zu suchen und dabei nach Kriterien wie Sendedatum und -zeit (Intervall), Sender u.ä. zu filtern. Die Suchfelder sollen dynamisch ausgewählt und kombiniert werden können (und, oder, Negation). Für Suchkomfort sollen Autocompletion in Suchfeldern, Fazetten, Datepicker und Sortierkriterien sorgen. Fazettierte Suche soll z.B. für Datumsbereiche wie heute, morgen, nächste n Tage; Sendergruppen öffentlich-rechtlich, privat, Radio); Tageszeit Vormittag/Nachmittag/Abend/Nacht angeboten werden.

Außerdem sollen mit Hilfe von regulären Ausdrücken automatisiert Hyperlinks für Darstellernamen u.ä. in der Internet Movie Database (als IMDB-Suche) sowie tentative Wikipedia- und Mediathek-URLs eingefügt werden.

Der Servercode soll mit NodeJS implementiert werden und auch als "Proxy" für den Apache-Solr fungieren, der aus Sicherheitsgründen NICHT direkt aus dem Webclient-Code angesprochen werden darf.

Die Webanwendung darf von Ihnen wahlweise als mobile oder Desktop-Anwendung implementiert werden. Echt responsive, für beide Plattformen funktionsfähige Anwendung, z.B. mit Hilfe von jQuery mobile oder Bootstrap führen zu Pluspunkten bei der Bewertung.

2 Hinweise zur Durchführung

Aus den Belegungen der oben beschriebenen Suchfelder sollen dynamisch Queries für Apache Solr zusammengebaut (jQuery-Beispielcode zum Zugriff auf Solr-Suchergebnisse s. Moodle), und die Queries anschließend z.B. mit http post oder get an den Solr-Server geschickt werden.

¹EPG = Electronic Program Guide; XMLTV ist ein Pseudostandard für EPG- und Senderdaten im XML-Format.

2.1 Solr installieren, konfigurieren, Index befüllen

Solr 8.5.2. herunterladen von <https://lucene.apache.org/solr> (binary distribution) Auspacken, dann Solr starten mit

```
cd solr-8.5.2
bin\solr start
```

Falls Java unter Windows nicht gefunden wird, wird empfohlen OpenJDK von <https://github.com/ojdkbuild/ojdkbuild> zu installieren, ggf. Systemumgebungsvariable PATH und JAVA_HOME setzen.

Dann core mit dem Namen epg erzeugen. Bitte benennen Sie Ihren Solr-Core epg und nicht anders, um das Testen Ihrer Webanwendung zu ermöglichen.

```
bin\solr create -c epg
```

Befüllen wie im Moodle angegeben:

```
java -Dc=epg -jar example/exampledocs/post.jar solrunique.xml
```

bzw. unter Linux

```
bin/post -c epg solrunique.xml
```

Es wird eine Datei `server\solr\epg\conf\managed-schema` angelegt. Diese bitte kopieren unter dem Namen `schema.xml` im selben Verzeichnis. Dann mit

```
bin\solr stop -p 8983
```

den Solr-Server beenden, und die Datei `schema.xml` mit einem Texteditor anpassen, so dass die aus den post-Daten inferierten Felder passendere Datentypen erhalten: Alle erhalten das Attribute `multivalued="false"`, die Felder `start` und `end` werden als `pdate` getypet, die Textfelder `title`, `subtitle` und `desc` erhalten statt `text_general` die Typen `text_de` bzw. `text_fr`, `channel` sollte als Typ `string` haben. Sie dürfen gern mit `<copyField>`s arbeiten, vgl. Abschnitte 2.6, 2.7 weiter unten. Beispiel für einen Eintrag (gekürzt an ...):

```
<doc>
  <field name="id">3sat@20200514114500</field>
  <field name="channel">3sat</field>
  <field name="start">2020-05-14T11:45:00Z</field>
  <field name="stop">2020-05-14T12:15:00Z</field>
  <field name="title_ger">Der Geschmack Europas</field>
  <field name="subtitle_ger">Sizilien</field>
  <field name="desc_ger">"Der Geschmack Europas" erkundet ...</field>
</doc>
```

Anschließend die Datei `managed-schema` und den bisherigen Index löschen über das Web-Interface oder das data-Verzeichnis unterhalb von `solr-8.5.2/server/solr/epg` löschen.

Dann nochmal neu indizieren mit `post`. Ob es funktioniert hat, sehen Sie im Webinterface daran, dass die eckigen Klammern um die Werte der Felder verschwunden sein sollten, da sie nun nicht mehr `multivalued` sind.

Suchkonfiguration können Sie über das Solr-Web-Interface testen (<http://localhost:8983/solr/#/epg/query>), die Suchanfrage als URL mit Parametern, Abholen per AJAX, am besten im JSON-Format (kann in der query als Parameter konfiguriert werden).

Dokumentation zu Solr: <https://lucene.apache.org/solr/resources.html>

2.2 Client: Benutzerführung und Gestaltung

Die Gestaltung der Benutzerführung ist Ihnen im Wesentlichen freigestellt. Sie soll vor allem ansprechend, funktional und sinnvoll sein. Sie können mehrere Seiten oder alles auf einer Seite als single page-Anwendung realisieren (ggf. dynamisch laden oder ausblenden, Wechsel/Ein-/Ausblenden mit jQuery-Animation, vgl. jQuery-Folien und sowie <http://jqueryui.com/demos/> bzw. <http://demos.jquerymobile.com/1.4.5/>). Z.B. könnten Sie ein accordion zum Ein-/Ausklappen der manchmal recht umfangreichen Sendungsbeschreibungen (bitte dort die Zeilenumbrüche mit anzeigen, d.h. für HTML in `
`-Tags umwandeln!) verwenden.

2.3 Link-Generierung mit Hilfe von regulären Ausdrücken

Es sollen, falls im desc-Feld Kandidaten für Schauspielernamen identifiziert werden, Suchanfragen-Links an die IMDB generiert werden. Beispielmuster, die erkannt werden sollen (nur mit den hier aufgeführten Triggerworten wie Regie:, Darsteller:, usw.).

1. Regie: Griffin Dunne

Drehbuch: Akiva Goldsman, Robin Swicord, Adam Brooks

Komponist: Alan Silvestri

Kamera: Andrew Dunn

Buch/Autor: Axel Bold, Edin Hadzimahovic

Schnitt: Michele Gentile

Musik: Uwe Schenk

Regie: Tapaas Chakravavarti, Siddharth Vasudeva

Darsteller:

Sandra Bullock (Sally Owens)

Nicole Kidman (Gillian Owens)

Außerdem existiert für Darsteller eine Variante bei bestimmten Sendern (es soll aber nicht senderspezifisch implementiert werden), zu erkennen am Bindestrich anstelle von runden Klammern und der vertauschten Reihenfolge:

Darsteller:

Vater Niccolo - Boris Aljinovic

Marco - Elia Francolino

Luigi - Linus Drews

Für alle Namen(-Kandidaten) soll immer nur jeweils ein Suchlink in die IMDB erzeugt werden. Beispiel: <https://www.imdb.com/search/name/?name=Sandra%20Bullock>.

Analog sollen Suchanfragen-Links für title, subtitle in Mediathekviewweb (nur Öffentlich-Rechtliche Sender), z.B. <https://mediathekviewweb.de/#query=Die%20Sendung%20mit%20der%20Maus>, eingefügt werden.

Es sollen jeweils nur Suchlinks erzeugt werden; es soll nicht überprüft werden, ob dafür auch eine Antwort gefunden wird.

2.4 Solr in der Anwendung hinter NodeJS als Proxy

Sie können zum Ausprobieren direkt mit dem Solr-Server interagieren, allerdings darf Ihr Webclient-Code später nicht direkt auf Solr zugreifen, sondern Solr muss hinter NodeJS als Proxy "versteckt" werden (sonst Punktabzug!). Normalerweise wird Solr serverseitig z.B. mittels eines Webservers wie Apache oder nginx, einem Proxy, PHP oder Node.js gewrapt, da sonst der Index per URL beliebig modifiziert bzw. gelöscht werden könnte.

2.5 Aufnahme-Programmierung

Die Aufnahme-Programmierung besteht lediglich in der Übernahme der Daten aus den gewählten Suchergebnissen in eine JSON-Datei auf dem Server. Dazu bitte bei der Anzeige von Suchergebnissen jeweils ein Optionsfeld zur Auswahl der Aufnahme anbieten. In einer JSON-Datei auf dem Server sollen ALLE Felder der Sendung gespeichert werden. Diese Daten sollen nachträglich in einer Liste "Gespeicherte Aufnahmen" angezeigt (subtitle/desc-Feld wieder als accordion) und Aufnahme-Einträge aus der Liste wieder gelöscht werden können.

2.6 Textuelle Inhalte mit Stemming, Mehrsprachigkeit, copyField

Um Mehrsprachigkeit in Webanwendungen und mit Solr zu adressieren, enthalten die EPG-Daten eines Senders französischsprachige Texte (ich konnte leider keinen Sender mit nichttriviale englischsprachigen EPG finden) in den Feldern `title_fra`, `subtitle_fra`, `desc_fra`, für die deutschsprachigen Einträge lauten die Feldnamen `title_ger`, `subtitle_ger`, `desc_ger` (Erklärung: diese dreibuchstabigen ISO 639-2 Suffices stammen aus der originalen XMLTV-Feld `lang`). Sie können den Feldsuffix global z.B. in einem Dropdownfeld für die Sprache oder durch klickbare Flaggensymbolen vom Benutzer wählen lassen und im JavaScript-Code zum Zusammenbauen der Solr-Queries automatisch anhängen.

2.7 Autocompletion-Felder

Wenigstens die Suchfelder zu `channel`, `title`, `subtitle` sollen mit autocompletion unterstützt werden. Dazu mit `<copyField>` arbeiten, um ungestemmte und gestemmte Varianten zu erhalten. Vgl. Abschnitte "Field Properties by Use Case" und "Copying Fields" im Apache Solr Reference Guide.

z.B. <https://examples.javacodegeeks.com/enterprise-java/apache-solr/solr-autocomplete-example/>

2.8 Datetimepicker

Für Komfortable Eingabe von zeitlichen Bereichen bei den Suchanfragen können Sie z.B. das jQuery-Plugin <https://plugins.jquery.com/datetimepicker/> verwenden.

Außerdem sollen voreingestellte Bereiche Vormittag/Nachmittag/Abend/Nacht (jeweils 6 Stunden, beginnend mit 6:00 Uhr) verwendet werden können.

Bitte fügen Sie in Ihre Webanwendung ein zusätzliches Feld für eine Datumswahl ein, mit das Datum für "heute" festgelegt werden kann, da die EPG-Daten nur bis Mitte/Ende Juni reichen.

2.9 Sendernamen und -Logos

Das Mapping der Sendernamen auf Senderlogo-Dateien (+ Info, ob Radio- oder TV-Station) erfolgt mit Hilfe der Datei `logonames.json`. Diese sowie fertige Logo-Bitmaps finden Sie als png-Dateien im Verzeichnis `senderlogos` in der Datei `Abgabeverzeichnisstruktur.zip`. Die Logos könnten Sie z.B. in den Suchergebnissen und verkleinert im Autocompletion-Feld für die Sendernamen anzeigen (vgl. jQueryUI-Doku zu `autocomplete`).

Quellenhinweis: Die Senderlogos stammen von <https://github.com/picons/picons>, genauer aus der Datei `snp-full.100x60-86x46.light.on.transparent_DATUM.hardlink.tar.xz`, zu finden unter <https://github.com/picons/picons/releases/>.

2.10 Fazettierte Suche, Datums- und Zeitfelder

Bitte Solr-Dokumentation zur fazettierten Suche konsultieren.

Das von Solr korrekt interpretierte Datums-/Zeitformat (pdate-Felder) hat folgende Struktur: YYYY-MM-DDTHH:MM:SSZ, wobei T und Z fest sind.

Mehr dazu finden Sie im Apache Solr Reference Guide, Abschnitt "Working with Dates".

Hier nochmal ein Beispiel für eine Bereichssuche im DateTime-Feld:

```
title_ger:Corona AND start:[2020-05-20T08:00:00Z TO 2020-05-21T10:00:00Z]
```

2.11 Browser-Kompatibilität

Die Anwendung sollte unter Firefox und Chrome lauffähig sein (auf beiden Browsern testen). Mit Ctrl+Shift+M können Sie (mobile) Ansichten in verschiedenen Bildschirmformaten testen.

2.12 Hervorheben der Suchanfragen im Suchergebnis

Bei der Anzeige der Resultate sollen die (positiven) Suchbegriffe in den Feldern title, subtitle, und desc hervorgehoben werden. Auch dieses highlighting erledigt Solr weitestgehend für Sie (hl-Parameter in der Query).

2.13 Beispiel Abholen von Solr query-Resultaten per jQuery-AJAX mit JSON

s. Beispielcode im Moodle

Für die Implementierung der Proxy-Funktionalität für Solr-Queries durch NodeJS hindurch wird (im server-Code auf NodeJS) die request-Bibliothek empfohlen. Vgl. auch das fetch-Beispiel im Moodle und piping requests: <https://github.com/request/request>. Aber bitte nicht die query vom client mit pipe ungesehen durchreichen, sondern wie in diesem Beispiel nur die Antwort (und die query z.B. in NodeJS-Funktionen zusammenbauen aus Parametern vom client):

```
http.createServer(function (req, resp) {
  if (req.url === '/doodle.png') {
    if (req.method === 'PUT') {
      req.pipe(request.put('http://mysite.com/doodle.png'))
    } else if (req.method === 'GET' || req.method === 'HEAD') {
      request.get('http://mysite.com/doodle.png').pipe(resp)
    }
  }
})
```

3 Abgabe

Der Quellcode ist getestet, frei von Fehlern nebst Dokumentation, Beispieldaten, Konfigurationsdateien usw. abzugeben. Neben HTML-, JavaScript- und CSS-Dateien sind das auch die Konfigurationsdatei (schema.xml, aus managed-schema abgeleitet und erweitert um Feld-, Filter- und Typdefinitionen für die EPG-Daten) sowie erweiterte Synonym- bzw. Stopword-Dateien für Solr aus solr-8.5.2\server\solr\epg\conf\.

Falls Sie zusätzliche Web-Frameworks verwenden, liefern Sie bitte ein Installationsskript (möglichst auf Basis von **npm**) im Verzeichnis server/install/ (s.u.) mit.

Der Quellcode darf keine obsoleten oder auskommentierten Anweisungen, Methoden oder Variablen enthalten (Ausnahme: nicht benötigte Teile in schema.xml können auskommentiert werden). Also vor der Abgabe nochmal checken, ob wirklich alles benötigt wird, was kodiert wurde, Code bereinigen.

Ihre Abgabe erfolgt in einer zip-Datei Nachname_Vorname.zip mit der folgenden, in der Datei Abgabeverzeichnisstruktur.zip vorgegebenen Verzeichnisstruktur:

```
Nachname_Vorname.zip          # Bitte Nachname durch Ihren Nachnamen
  +-- Nachname_Vorname/        # und Vorname durch Ihren Vornamen ersetzen
    +-- doc/
      |  +-- Nachname_Vorname.pdf # Ihre Dokumentation (PDF, KEIN Word)
    +-- src/
      |-- server/               # Quellcode für NodeJS-Server
      |  |-- logofiles.json     # Mapping der Senderlogo-Dateien
      |  |-- server.js          # Server (NodeJS; leere Datei)
      |  |-- install/           # Installationsskript(e) für Server (NodeJS)
      |  +-- static/            # Quellcode für Client
      |     |-- css/            # CSS-Dateien
      |     |-- img/            # Bilddateien, auch Ihre eigenen
      |     |  +-- senderlogos/ # Senderlogos (png)
      |     |-- index.html      # Ihre Webanwendung (Client; leere Datei)
      |     +-- js/             # Ihre JavaScript-Quelldateien (nur client)
    +-- solr/
      |-- solrunique.xml        # EPG-Daten im Solr-XML-post-Format
      +-- conf/                 # Ihre Solr-Konfiguration
          +-- lang/              # nur geänderte sprachspezifische Dateien (de/fr)
```

Sie dürfen die Dateistruktur um eigene Dateien und Unterverzeichnisse erweitern. Die Grundstruktur mit den Verzeichnisnamen sowie client (index.html) und server-Code (server.js) soll jedoch gleich bleiben. Vor der Abgabe bitte die EPG-Daten (Datei solrunique.xml; 28 MB groß) löschen und natürlich auch den Solr-Index nicht mit abgeben.

Eine mindestens 6-seitige Beschreibung (exklusive Screenshots) Ihrer Studienarbeit ist mit abzugeben. Kern der Beschreibung soll die Web-Anwendung selbst sein. Es sind alle verwendeten Quellen in einem Quellenverzeichnis am Ende der Dokumentation anzugeben.

Die Erklärung, dass Sie keine Teile der Arbeit (außer den angegebenen) von anderen übernommen, sondern das Programm selbst geschrieben haben, ist per Moodle abzugeben.

Nach der Abgabe bitte regelmäßig email checken, falls Rückfragen nötig sind.

4 Bewertungsgrundlagen

- Studienarbeits-Dokumentation (PDF), Screenshots, Quellenangaben, Aufbau
- Usability der Anwendung, Benutzerführung, Layout
- Code: Korrektheit, Vollständigkeit, Eleganz, Lösungswege
- Code-Dokumentation: inline
- Kurzvortrag mit Demonstration der Anwendung

Viel Spaß und viel Erfolg!