

Dokumentation zur Projektarbeit



Music Genie™

© Don Promillo

Angewandte Informatik

Web Services Vorlesung

2020

Sascha Vonier | 8917780

Thorben Böer | 1626763

Jan Dietzel | 7737633

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1. Allgemeines zum Projekt.....	3
Ideen und Ansatz	3
Projektplanung.....	3
2. Anforderungen und Modellierung.....	5
Funktionale Anforderungen	5
BPMN Modellierung	5
Web-Oberfläche.....	6
3. Implementierung.....	9
Einbeziehen der APIs	9
Der Web Service	10
Anhang	13

1. Allgemeines zum Projekt

Ideen und Ansatz

Die Idee des Web Service „Music Genie“ entstand aus der Idee, einen erweiterten Spotify Service umzusetzen. Dieser sollte in der Lage sein, die Lyrics aus der Originalsprache in eine beliebige, vom Anwender ausgewählte Sprache zu übersetzen. Damit sollte zum besseren Verständnis des Liedtextes beigetragen werden, um den vermittelten Inhalt des Künstlers besser zu verstehen. Ein weiterer Bestandteil der unserer Meinung nach nicht gut genug in Spotify ausgeprägt war, ist die Menge an Informationen zum Künstler und zum Werk selbst. So kam die Idee, Spotify mit einer anderen API zu unterstützen, um dem Anwender die Recherche zu ersparen.

Des Weiteren stand die Idee im Raum, einen Equalizer zu integrieren, mit dessen Hilfe eine erweiterte Steuerung der Höhen und Tiefen der Audioausgabe möglich wäre. Hier gab es schon erste Bedenken, in wie weit die Audioausgabe von Spotify manipuliert werden kann bzw. manipuliert werden darf. Erst bei der Implementierung fiel auf, dass die Spotify API keine Audioausgabe bereitstellte, geschweige denn einen Musik-Player. Somit verfiel die Idee, Spotify um einen Übersetzer und eine Lyrics-Angabe zu erweitern.

Letztendlich entschieden wir uns dazu, weiter an der Idee zu arbeiten, verloren aber das Interesse an einem integrierten Music Player. Wir kamen zum Entschluss, dem Anwender die Wahl seines priorisierten Musik-Player zu lassen. Ob nun Apple Music oder Spotify, unser Service sollte für jeden Anwender nutzbar sein.

So ergab sich der neue Webservice „Music Genie“, basierend auf drei APIs, für die Realisierung der oben erläuterten Funktionen, dazu mehr in Kapitel 2.1 Anforderungen.

Projektplanung

Der Verlauf des Projekts lässt sich in die folgenden Teile gliedern: Themenfindung, Thema-ausarbeitung, Informationsbeschaffung zu möglichen Technologien und Implementation.

Beginnend mit dem Themenfindungsprozess und der endgültigen Themaausarbeitung am 2. März, erschlossen wir aus drei möglichen Themengebieten und mehr oder weniger sinnvollen Ideen zu möglichen Funktionen unser Thema. Als es zur Recherche für mögliche Technologien und Programmiersprachen für den Web Services kam, sahen wir für den Anfang nur die Möglichkeit zur Umsetzung mit PHP. Nach intensiver Auseinandersetzung mit dem Thema setzten wir noch die Möglichkeit der Umsetzung mit C# und WSDL via IDE Visual Studio auf die Liste. Letztendlich wurde sich einstimmig für die Umsetzung mit REST APIs und PHP entschieden.

Der Plan sah die Aufgabenverteilung, aufgrund vorliegender Informationen zu den jeweiligen Qualifikationen, wie folgt vor:

Sascha:

- Einbindung der APIs
- Erstellung des Web Service
- Ausarbeitung der Implementation (Dokumentation)

Thorben:

- BPMN Modellierung des Web Services
- Performance Tests

Jan:

- Web-Oberfläche (Nutzeransicht)
- Ausarbeitung der Dokumentation

2. Anforderungen und Modellierung

Funktionale Anforderungen

Der Service musste über gewisse Funktionen verfügen, um unsere Vorstellung des „Music Genie“ zu realisieren. Vorrang hat die Bereitstellung der vollständigen Lyrics eines Werkes, welche mittels eines Übersetzers in eine beliebige Sprache übersetzt werden soll. Zudem soll es eine Möglichkeit der Eingabe für Titel oder Interpret geben. Neben der Ausgabe des Liedtextes sollen zusätzlich noch Informationen zum Werk selbst auf der Oberfläche erscheinen. Darunter befinden sich das Erscheinungsjahr, zugehöriges Album und das Cover.

Ergänzend muss die Oberfläche übersichtlich aufgebaut und mit CSS entsprechend gestaltet werden. Eine einfache und selbst erklärende Benutzung der Funktionen via Eingabe und Knöpfen als Möglichkeit der Interaktion ist essentiell.

BPMN Modellierung

Im Folgenden wird der Ablauf des Web-Services mit BPMN-Modellen grafisch dargestellt.

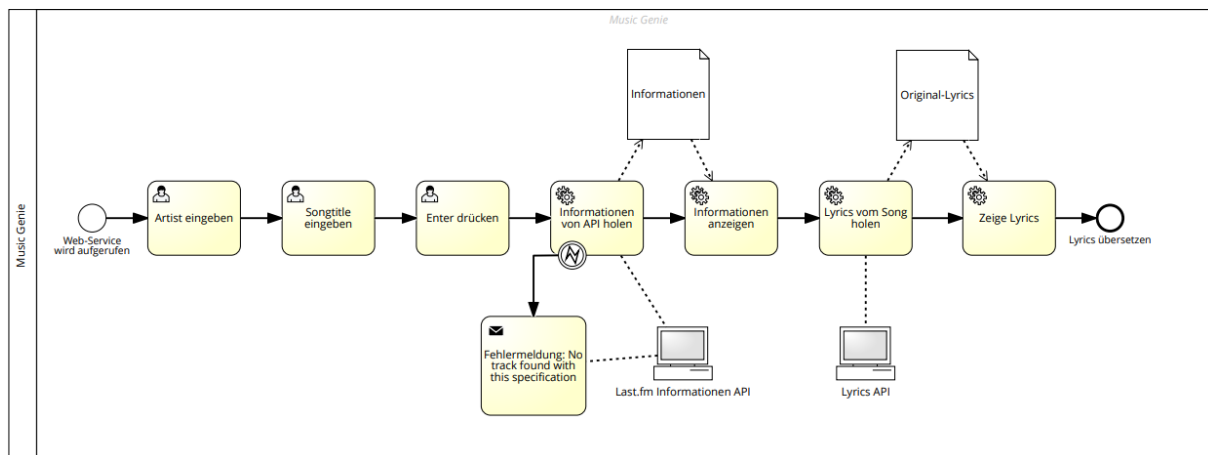


Abbildung 1: BPMN-Modell des Web-Services (1)

Der Nutzer startet den Web-Service, indem er diesen aufruft. Wie in Abbildung 3 gezeigt, kann er nun einen Interpreten und Songtitel eingeben. Nachdem er auf Enter gedrückt hat, versucht der Web-Service die Informationen des gewünschten Songs zu holen, indem dieser auf die *Last.fm Informationen API* zugreift. Hierbei wird der Interpret und Songtitel an die API übergeben und diese liefert die Informationen zurück. Existiert kein Lied mit dem eingebenden Songtitel von dem eingebenden Interpreten, so wird eine Fehlermeldung von der *Last.fm Informationen API* zurückgegeben, welche dann weiter an den Nutzer ausgegeben wird. Diese lautet „No track found with this specification“.

Wurde ein existierendes Lied gefunden, so werden die erhaltenden Informationen dem Nutzer auf der Web-Oberfläche angezeigt. Anschließend wird die Lyrics des gefundenen Songs angezeigt. Hierzu übergibt der Web-Service den Interpreten und Songtitel an die *Lyrics API*. Diese liefert den entsprechenden Lyrics zurück, welcher wieder auf der Web-Oberfläche angezeigt wird.

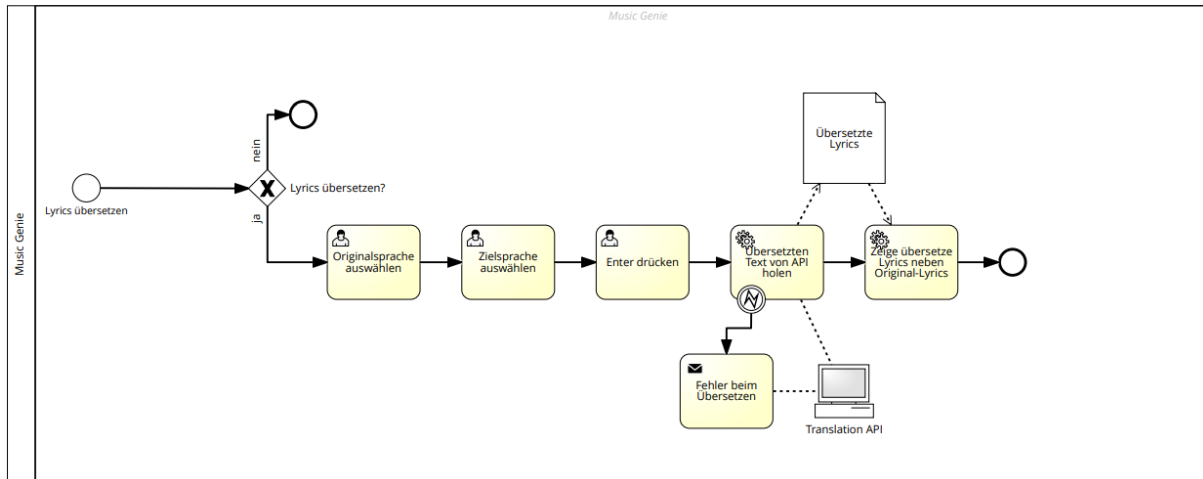


Abbildung 2: BPMN-Modell des Web-Services (2)

Abschließend kann der Nutzer nun noch zusätzlich die Lyrics übersetzen lassen. Hierzu wählt dieser einfach die Originalsprache der Lyrics und die gewünschte Zielsprache. Nach dem Bestätigen mit Enter versucht nun der Web-Service die Lyrics zu übersetzen, indem dieser die Originalsprache, die Zielsprache und den zu übersetzenden Text an die *Translation API* übergibt. Diese gibt im Normalfall den übersetzten Text in der Zielsprache zurück, welcher schlussendlich neben der originalen Lyrics auf der Web-Oberfläche angezeigt wird. Bei einem Fehler wird die entsprechende Fehlermeldung der *Translation API* zurückgegeben, welche ausgegeben wird.

Web-Oberfläche

Abbildung 3 zeigt den Ausgangszustand der Weboberfläche. Nach Eingabe des Titel und des Interpret des gewünschten Musikstücks in die zwei vorhandenen Eingabefelder (Abbildung 4), kann man nun mit dem Button *Enter* fortfahren. Wird keine Eingabe oder eine falsche Eingabe getätigt, wird eine Fehlermeldung aus der Oberfläche ausgegeben.

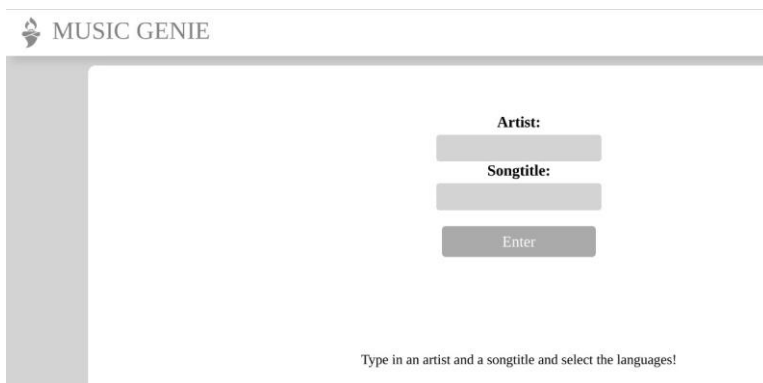


Abbildung 3: Web-Oberfläche im Ausgangszustand

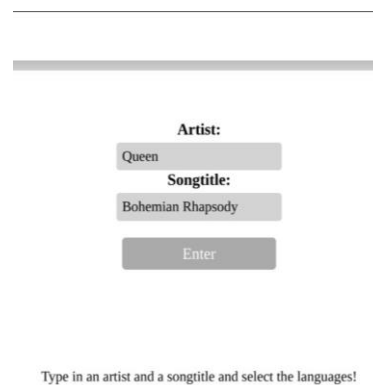


Abbildung 4: Eingabe getätigt

Die Oberfläche aus Abbildung 5 erscheint. Es werden verschiedene Informationen zum Werk angezeigt, wie auch das Cover-Bild. Unter den elementaren Informationen befinden sich Titel, Interpret und Album. Zusätzlich werden noch die Anzahl aller Aufrufe des Titels, die Gesamtzahl der diversen Zuhörer und die entsprechenden Tags (im Grunde Kategorien, die Musikart bzw. Genre angeben). Die Eingabefelder bleiben nach wie vor erhalten, so dass eine weitere bzw. neue Eingabe möglich ist. Momentan enthalten ist die letzte Suche.

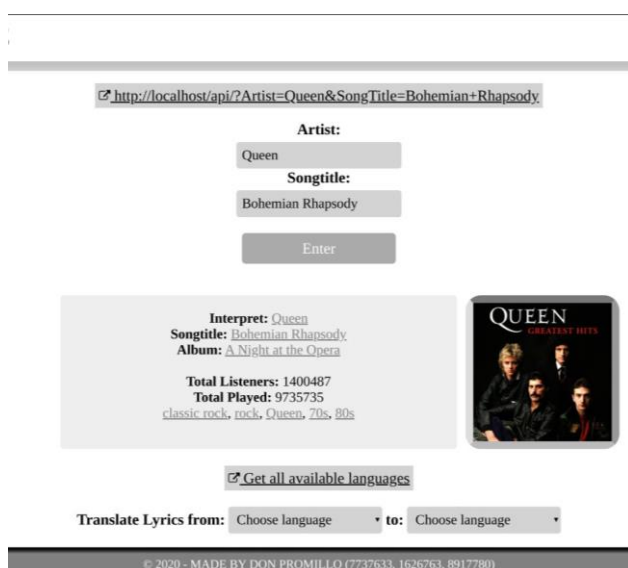


Abbildung 5: Oberfläche nach Eingabe, Informationen

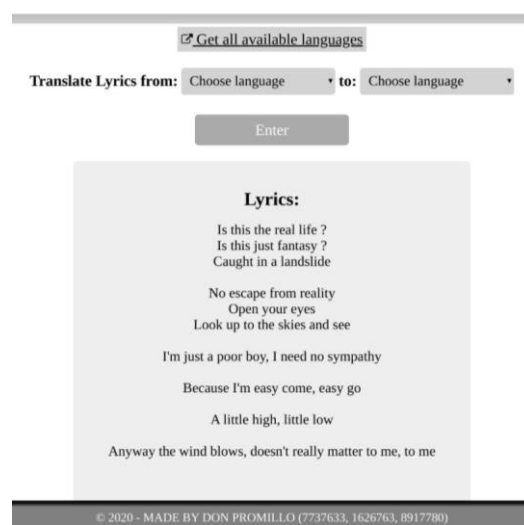


Abbildung 6: Oberfläche nach Eingabe, Lyrics

Weiter unten auf der Oberfläche (Abbildung 6) erscheinen die zugehörigen Lyrics. Darüber befindet sich die dritte und letzte Funktion des Webservices, der Übersetzer. Nach der Eingabe der Ausgangssprache des Lieds kann nun eine neue Sprache ausgewählt werden, die Sprache, in die der Liedtext übersetzt werden soll.

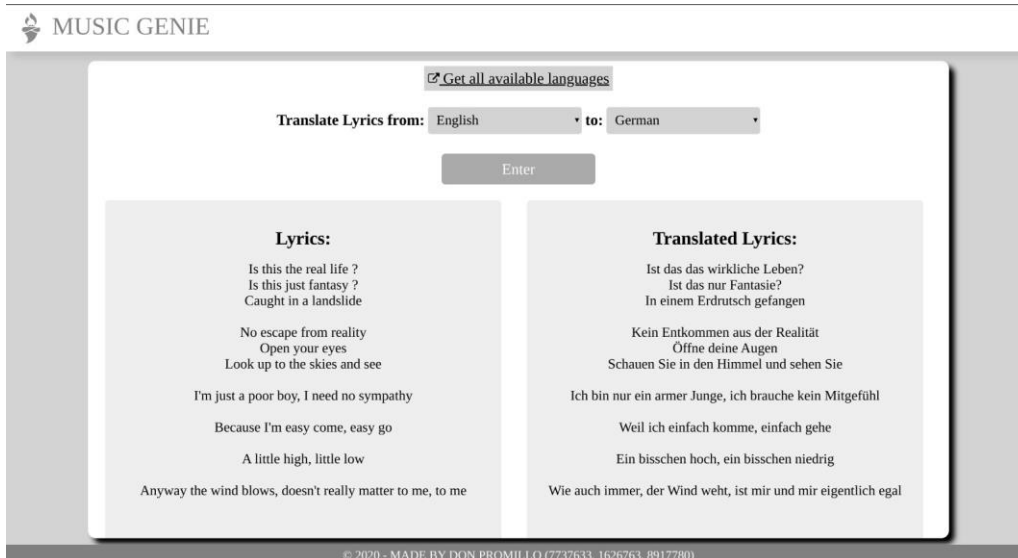


Abbildung 7: Anzeige Lyrics übersetzt

Mit dem Button *Enter* gelangt man zur Ansicht in Abbildung 7. Hier sieht man die Übersetzung des Werks von Englisch nach Deutsch, rechts dargestellt. Um zur Liste aller möglichen Sprachen ist auf den Link *Get all available languages* zu klicken.

3. Implementierung

Einbeziehen der APIs

Translation API

“WhatsMate Translation API enables your application to translate text from one language to another easily via a REST API” (Whatsmate). Bei dieser API müssen die Originalsprache, die zu übersetzende Sprache, sowie den Text übergeben werden. Die API gibt anschließend den übersetzten Text zurück. (Für weitere Informationen:

<https://www.whatsmate.net/translation-api.html>)

Lyrics API

LyricsOvh muss ein Interpret sowie der Songtitel übergeben werden. Anschließend gibt diese den Lyrics zurück. (Für weitere Informationen:

<https://lyricsovh.docs.apiary.io/#reference/0/lyrics-of-a-song/search>)

Last.fm Informationen API

“last.fm ist eine soziale Software und ein Online-Musikdienst. Nutzer erhalten aufgrund ihrer Hörgewohnheiten Empfehlungen für neue Musik, Hinweise auf Konzerte in ihrer Umgebung und Kontaktvorschläge für Nutzer mit ähnlichem Musikgeschmack.” (Wikipedia)

Aufgrund der Beliebtheit der Seite haben wir uns für deren API entschieden. Diese benötigt einen Interpreten sowie den Songtitel und gibt einige Informationen zum Song, Künstler und Alben zurück, sowie Links zu den entsprechenden Last.fm Seiten. (Weitere Informationen:

<https://www.last.fm/api/?lang=de&>)

Der Web Service

Index

Die index.php befindet sich im public-Ordner des Projekts. Sie wird jedes mal, wenn ein neuer Aufruf stattfindet gestartet.. Dies muss natürlich im Voraus in den Webserver-Konfigurationen eingestellt werden (verwendet wurde Nginx). Diesen Vorgang nennt man Routing. Dieses Routing hat jedoch nichts dem PHP-Routing zu tun. Beim PHP-Routing wird in der index.php Datei das Factory-Objekt erstellt und gestartet. Die Application wird nun von der Index aufgerufen. Sie ruft anschließend der Router auf (der dann das eigentliche PHP-Routing betreibt), der den Benutzer auf die richtige Seite leitet.

Factory

Wie der Name schon sagt wird in der Factory die komplette Webseite zusammengebaut. Hier findet die Erzeugung aller Klassen statt. Das hat den Vorteil, dass bei Änderungen der Struktur nur die Factory angepasst werden muss. Die Factory ist zudem die einzige Klasse, in der das new-Schlüsselwort vorkommt, d.h. nur hier werden alle Objekte erstellt.

Application

Mit der Application startet sozusagen die Web-Anwendung. Sie wird in der Factory erzeugt und in der Index aufgerufen. Beim Aufruf wird mit dem Start der Application der Router aufgerufen.

Router

Ein Router in PHP, wird in der Regel von der Application oder von darüber liegenden Routern aufgerufen. Er leitet den Benutzer auf die richtige Page anhand dem Uniform Resource Locator (URL) weiter. Dieses Prinzip wird PHP-Routing genannt. Es dient der Übersichtlichkeit des Projekts. Erkennt der Router nach der Domain das Unterverzeichnis api, so wird anstelle der Page der APICreator aufgerufen, der die API zusammenbaut.

Page

Sie beinhalten die run-Methode, die beim Aufruf des Page-Objekts gestartet wird. Sie holt sich die Daten der selbstgebaute API. Zu der Page gibt es auch einen Projektor, der die notwendigen HTML-Templates zusammenbaut und diese an die jeweilige Page zurück gibt.

Projektor

Jede Page hat ihren eigenen Projektor. Ein Projektor wird von seiner Page aufgerufen. Er baut je nachdem, was die Page anfordert, die entsprechenden HTML-Templates zusammen. Der Projektor hat für die Grundkonstruktion sein eigenes Haupttemplate, bei dem alle Platzhalter dementsprechend ersetzt werden.

APICreator

Der APICreator empfängt vom APIConnector die notwendigen Daten und baut diese zu einer Rest-API zusammen, welche eine JSON-file zurückgibt.

APIConnector

Der APIConnector holt sich die Daten der verbundenen REST-APIs und gibt diese zurück.

Weboberfläche

Die Weboberfläche ist mithilfe eines html-Templates umgesetzt. Dieses bildet die Basis. Durch die php-Klasse Projektor, werden die auf dem Template vorhandenen Platzhalter mit den eigentlichen Informationen ausgetauscht. In der Datei stylesheet.css wird die Weboberfläche noch etwas verschönert.

Zu Beginn kann der Benutzer einen Interpreten seiner Wahl und einen Titel des Interpreten eingeben. Anschließend wählt in den Dropdown-Menüs einmal die Originalsprache des Songs und die zu übersetzende Sprache aus. Sind die Daten korrekt und er klickt danach auf Enter, werden ihm einige Informationen zum Song, sowie Links zu Last.fm und der Lyrics, wie auch der übersetzte Lyrics zurückgegeben.

Die eigenen APIs

Mithilfe der URL "<http://localhost/api/availableLanguages>" bekommt man alle unterstützten Sprachen des Übersetzers als JSON zurück. Dabei ist der Aufbau wie folgt:

```
{
  "languages": {
    "{Ländercode}": "{Ländername}",
    ...
  }
}
```

Ersetzt man in der folgenden URL die Platzhalter, so bekommt man alle restlichen Daten im JSON-Format zurück:

<http://localhost/api/?Artist={name}&SongTitle={titel}&fromLanguage={Ländercode}&toLanguage={Ländercode}>

Der Aufbau der JSON-Datei ist wie folgt:

```
{
  "track": {
    "songTitle": "{Titel}",
    "url": "{URL}",
    "artist": {
      "name": "{Name}",
      "url": "{URL}"
    },
    "album": {
      "name": "{Name}",
      "url": "{URL}",
      "cover": "{URL}"
    },
    "lyrics": {
      "originalLyrics": "{Lyrics}",
      "translatedLyrics": "{Lyrics}"
    },
    "listeners": "{Anzahl}",
    "playCount": "{Anzahl}",
    "tags": [
      {
        "name": "{Name}",
        "url": "{URL}"
      },
      ...
    ]
  }
}
```

Anhang

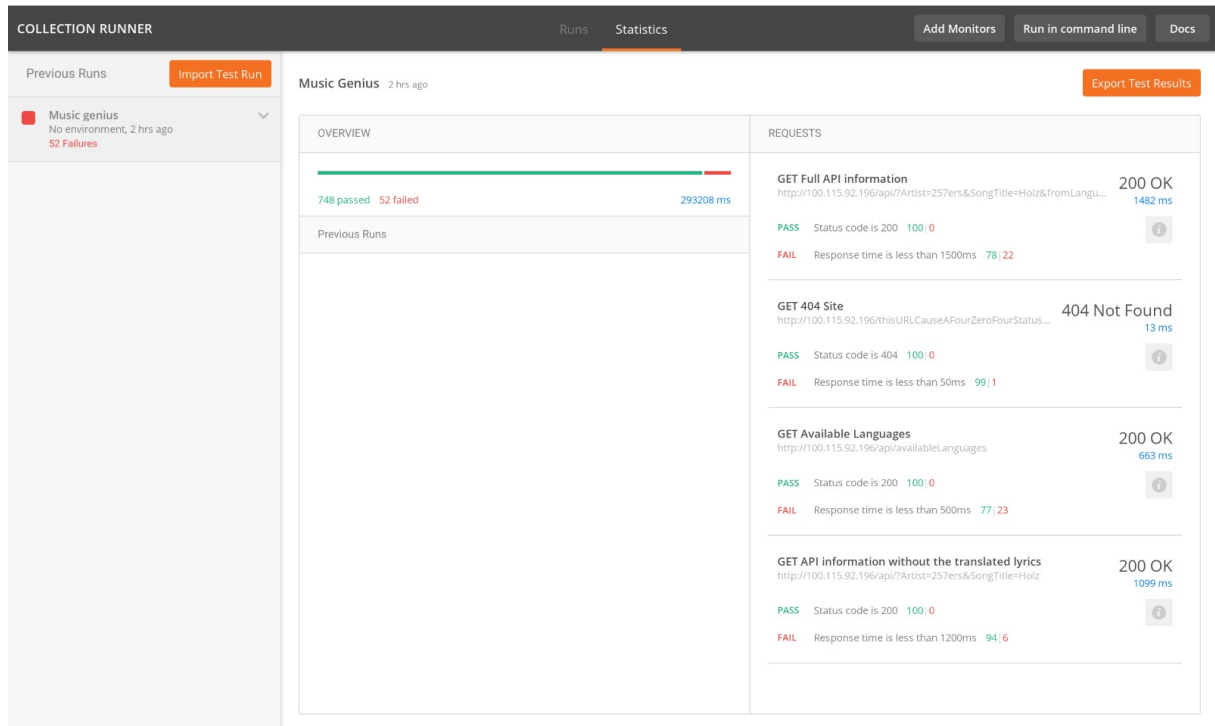


Abbildung 8: Testergebnisse des Webservices Music Genie