WIKIPLAG FRONTEND

Frontend in Angular v.6

Inhaltsverzeichnis

Webapp / Frontend

Installation

Voraussetzungen:
Um die Anwendung zu starten, müssen folgende Programme bereits installiert sein:
Node.js 8 (oder höher)

Herunterladen:
Mit folgendem Befehl wird das Repository heruntergeladen:
git clone https://github.com/WikiPlagSS2018/webapp.git

Dann wechselt man in den neuen Ordner:
cd webapp

Nun lädt man die Abhängigkeiten des Node.js-Servers herunter:
npm install oder yarn install

(Hinweis: dieser Befehl ist nur bei dem erstmaligen Start der Anwendung und nach Update des Repositorys mittels git pull nötig.)

Starten:
Nun startet man das Programm durch Start des Node.js-servers:
npm start *wenn Sie* ***npm*** *installiert haben,* oder **yarn start** *wenn Sie* **yarn** *installiert haben***.**

Wenn Fehler durch Node-Module existieren sollte, könnte Module update gestartet werden:
Npm update, oder yarn update

Nun sollte sich im Default-Browser eine neue Seite öffnen, die die Anwendung enthält. Sollte dies nicht automatisch passieren, dann folgendes in die Adresszeile des Browsers eingeben:

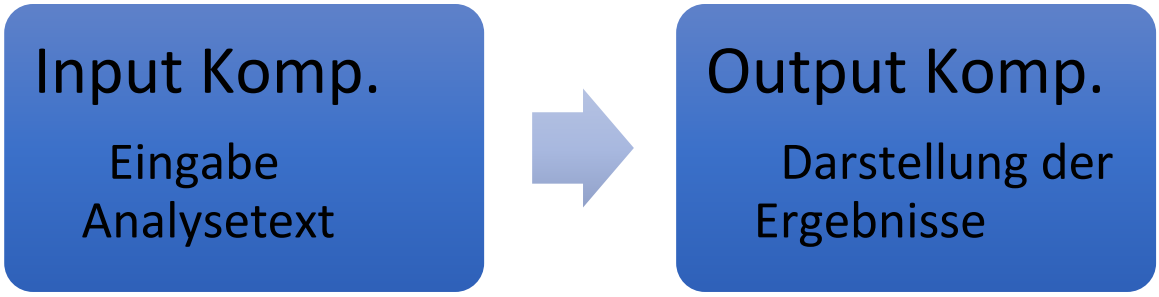
http://localhost:4200
oder
http://localhost:3000

Falls das Frontend ohne Serververbindung gestartet werden soll lässt sich dieses mit einer Mock starten:
Dafür die Datei *plag-positions.service.ts* öffnen und in der Methode *checkforPlag ()* das erste Return-Statement auskommentieren und das bereits auskommentierte Return-Statement in der Zeile darunter entkommentieren.

Grundlegender Aufbau

Das Frontend besteht grob gesehen aus zwei Komponenten.

- Input-Komponente (*input.component*)
    - Hier wird die Möglichkeit geboten einen Text einzugeben, sowie eine schon übermittelte Anfrage aus der History zu laden.
- Output-Komponente (*output.component*)
    - Die Komponente ist verantwortlich für das Darstellen der von Server erhaltenen Analyse, außerdem kann von hier aus eine PDF generiert werden, sowie ein neuer Text als Input eingegeben werden.



Die Komponenten im Detail

Navigation Komponente

Diese Komponente wird in jeder Seite mit eingebunden und beinhaltet den Header der Seite, inklusive Wikiplag Logo usw.
Von hier aus kann über einen Klick auf das Logo wieder zurück zur Startseite gelangt werden (Siehe Routing).

Input-Komponente

In einer Textarea kann hier der Input-Text eingegeben werden. Anschließend wird über den Button „Analysieren“ die Methode send in der *input.component.ts* aufgerufen.
Send()

- Validierung von Input (Leeres Textfeld oder weniger als 100 Zeichen sind invalide)
- Entscheidung ob die Anfrage schon einmal gestellt wurde und aus Local Storage geladen wird, oder ob eine Anfrage an Wikiplag Server gestellt wird

Laden von Local Storage:
Die Daten sind schon vorhanden (Die exakt gleiche Anfrage wurde schon einmal gestellt). Die Daten werden dann aus dem Local Storage geladen (Vgl. LocalStorageManager) und die Output-Komponente wird aufgerufen. Außerdem werden einige Animationen angewandt um das Wechseln der Komponenten optisch aufzupeppen.

Laden von Server:
Der PlagPositionService wird aufgerufen und eine Anfrage an den Server gestellt (Siehe PlagPosService). Anschließend wird die Anfrage + Antwort im Local Storage gespeichert (Siehe LocalStorageManager) und die Output-Komponente wird aufgerufen. Außerdem werden einige Animationen angewandt um das Wechseln der Komponenten optisch aufzupeppen.

Output-Komponente

Es gibt zwei große Bereiche in der Output Komponente.
Im Linken wird der analysierte Text dargestellt (incl. Makierungen der Plagiate). Weiterhin werden die nicht als Plagiat erkannten stellen „eingeklappt“, um die Darstellung der Ergebnisse übersichtlicher zu machen. Mit den blauen Buttons am linken Rand kann man diese Text dann bei Bedarf ausfahren.
Das rechte Feld dient zur Darstellung des im linken Bereich ausgewählten Plagiats. Hier wird der Titel des Wikipedia Eintrages, sowie ein kleiner Auszug des Artikels angezeigt. Mit einem Klick auf ein Plagiat wird man zum jeweiligem Wikipedia-Artikel weitergeleitet.

preparePlagiarismResponse()
Bereitet die Antwort des Servers auf. Via textShorteningService wird der Text zusammengefasst.

alertNumberOfPlags()
Gibt die Nummer der gefundenen Plagiate aus.

changeViewState()
Diese Methode dient zum aus- und einklappen von nicht als Plagiat markierten Textstellen im linken Bereich der Output-Komponente.

pdfExport()
Hierrüber wird der Service pdfGenerator aufgerufen um eine PDF aus den aktuellen Analyseergebnissen zu erstellen.

showPlagOnWikipedia()
Beim Klick auf ein Plagiat wird diese Methode aufgerufen und ein neuer Tab mit dem jeweiligem Wikipedia Artikel aufgerufen.

clickedOnPlagiarism()
Diese Methode wird aufgerufen wenn im linken Bereich der Output Komponente ein Plagiat angeklickt wird. Dieses Plagiat wird farblich markiert und im rechten Bereich der Output-Komponente werden die jeweiligen Wikipedia-Artikel eingeblendet.

Services

Alert Service

Der Alertservice stellt Methode bereit die als Wrapper für die JS Library SweetAlert dienen. Dieses bietet die Möglichkeit optisch attraktive Alerts auf der Website anzuzeigen.

change-to-input-component-guard

Dieser Service stellt sicher, das der User in der Output-Komponente vor dem Wechsel zur Startseite über einen Dialog seine Entscheidung Bestätigung muss. Dies sichert den Nutzer vor eventuellem Datenverlust ab.

local-storage-manager

Der Service bietet eine Schnittstelle zur Verwaltung der im Local Storage gespeicherten, alten Nutzeranfragen. Diese werden Base64 encoded und dann gespeichert.

requestAlreadyInLocalStorage()
Analysiert ob ein Text bereits im Local Storage gespeichert ist und gibt in diesem Falle true zurück.

Clean()
Löscht alle im Local Storage befindlichen Daten

getRecentlyStoredRequests()
Gibt die letzten gespeicherten Plagiate, geordnet nach Anfragdatum zurück.

saveResponseToLocalStorage()
Speichert eine noch nicht vorhandene Anfrage im Local Storage

getResponseFromLocalStorage()
Gibt eine bestimmte Anfrage vom Local Storage zurück.

PDF-Generator Service

Als Export-Möglichkeit bietet Wikiplag die Erstellung einer PDF-Datei an. Hierzu wird die JS Library jsPDF genutzt.

generatePDF(PlagResponse)
Die Methode nimmt das aktuell in der Output Komponente dargestellte PlagResponse Objekt entgegen und kreiert daraus eine PDF-Datei mit Links zu den jeweiligen Plagiaten auf Wikipedia.

Zur Nutzung und Funktion von JSPDF sei auf deren Dokumentation verwiesen
https://rawgit.com/MrRio/jsPDF/master/docs/index.html

PlagPositionService

Der PlagPositionService dient zur Übersendung der Input Daten an den Wikiplag Server.

checkForPlag()
In dieser Methode wird ein Post Request mit dem vom User eingegebenen Text an den Wikiplag Server gesendet. Die Empfangene Antwort wird zu einem PlagResponse Objekt gecastet.

text-Shortening

Dieser Service dient zur Zusammenfassung der von Wikiplag empfangenen Antwort. Um dem User eine möglichst übersichtliche Darstellung zu bieten, werden die Textstellen, die nicht als Plagiat gekennzeichnet sind gefiltert und anschließend in der Output-Komponente

„eingeklappt“. Wichtig hierbei ist, das die Variable charsBeforeAndAfterPlag angibt, wie viele Zeichen von und nach einem Plagiat mit dargestellt werden sollen.
Der Text liegt dann später in einem Array mit Objekten vom Typ SummarizedOutputTextPiece vor und wird so in der Output Komponente verabeitet.

getNextStartTag()
Diese Methode liefert die Stelle, an der der nächste <span Tag beginnt, der den Anfang eines Plagiates kennzeichnet. Hierbei ist wieder die Variable charsBeforeAndAfterPlag zu beachten.

getNextEndPos()
Liefert analog zur Methode getNextStartTag die nächste Endposition eines Plagiates.

splitFirstPlanOccurrence()
Entfernt das erst vorkommende Plagiat aus einem Inputtext und speichert es ab.

removeTextBeforeFirstPlagarism()
Entfernt den Text vor einem Plagiat und speichert ihn entsprechend in der Variablen shortenedPlagarismText.

Update

04.01.2019

Als Update wird die Methode openFile() im input.component.ts dazu geschrieben. Der Inhalt der Datei wird mit Hilfe von FileReader in den Textarea per Drag-and-Drop gelesen.