Waves Mashup

Samr Alakrad
Hochschule Ruhr West
Wirtschaftsinformatik
10013285
Essen, Deutschland
samr.alakrad@stud.hs-ruhrwest.de

Leif Plewe
Hochschule Ruhr West
Wirtschaftsinformatik
10004565
Mülheim an der Rhur, Deutschland
leif.plewe@stud.hs-ruhrwest.de

Mohand Almohammad
Hochschule Ruhr West
Wirtschaftsinformatik
Oberhausen, Deutschand
line 4: City, Country
mohand.almohammad@stud.hs-ruhrwest.de

Romuald Ngongang Djugouo
Hochschule Ruhr West
Wirtschaftsinformatik
10013531
Hamm Deutschland
romuald.ngongang-djugou@stud..hs-ruhrwest.de

Abstrakt: In diesem Dokument wird die Entwicklung eines Mashups von dem oben genannten Team im Rahmen eines Hochschul-Moduls namens "Verteilte Systeme" dokumentiert. Es wird beschrieben wie das Team die Anforderung der Aufgabe gelöst hat.Keywords—Waves, API, Mashup, Moving Average, Sharpee Ratio, Sentimentanalyse.

I. Einleitung

Das Mashup aggregiert unterschiedliche Daten von verschiedenen Quellen, die im Backend verarbeitet werden, um ein bestimmtes Problem zu lösen. Dabei wird nach dem Prinzip "Same-Origin-Directive" implementiert. Das heißt die Daten werden durch den eigenen Server geleitet und das Frontend bezieht die Daten ausschließlich aus dem eigenen Backend. Bei den Daten handelt es sich um Informationen bezüglich der Kryptowährung "Waves". Es werden Preis- und Zinsdaten aggregiert sowie eine Twitteranalyse zu bestimmten Hashtags durchgeführt.

II. Anlagen

- Deployment: https://wavesmashup16.herokuapp.com/waves
- GitHub-Repository: https://github.com/Leif-hash/WavesMashup
- SwaggerUI: https://wavesmashup16.herokuapp.com/swagger-ui/index.html

III. Genutzte Services

- https://wavescap.com/api/
- https://api.coingecko.com
- https://wavescap.com
- https://text-sentiment.p.rapidapi.com/analyze

A. Benutzerszenarien

Im folgendes werden verschiedene Benutzerszenarien für die Nutzung der Wavesmashup Funktionen dargestellt.

- Als User möchte ich mir die Marktpreise von Waves einsehen, um mich über die Preisentwicklung auf dem Laufenden zu halten.
- Als User möchte ich die wöchentliche Kursentwicklung von Moving Average von Waves in USD ansehen, damit ich voller Zugriff auf die Preisdaten von Waves, indem ein ständig aktualisierter Durchschnittspreis erstellt wird.
- Als User möchte ich mir die periodische Kursentwicklung von Sharpee Ratio von Waves in USD ansehen, damit ich das Risikoniveau und die angepasste Rendite aller Investmentfonds ermitteln kann. Dies gibt mir ein klares Bild und lässt mich erfahren, ob Verhältnis von Risiko und Rendite vorteilhaft ist.
- Als User möchte ich alle Tweets mit bestimmten ,“Waves” bezogenen Hashtags, gesammelt angezeigt bekommen.
- Als Nutzer möchte ich eine Sentimentanalyse aufrufen können, um ein Gefühl für die aktuelle Stimmung gegenüber der Kryptowährung “Waves” zu bekommen. Aus der Stimmung lassen sich Prognosen für die Preisentwicklung ableiten.
- Als User möchte ich die Waves-Zinssätze für verschiedene Finanzprodukte nach Höhe sortieren, um beurteilen zu können, welche Kryptowährung im Vergleich zu Waves die stärkste ist.

B. Use Case Diagram

Die in der Aufgabe gennanten Anforderungen sind auf folgendes Use Case Diagram zu sehen.

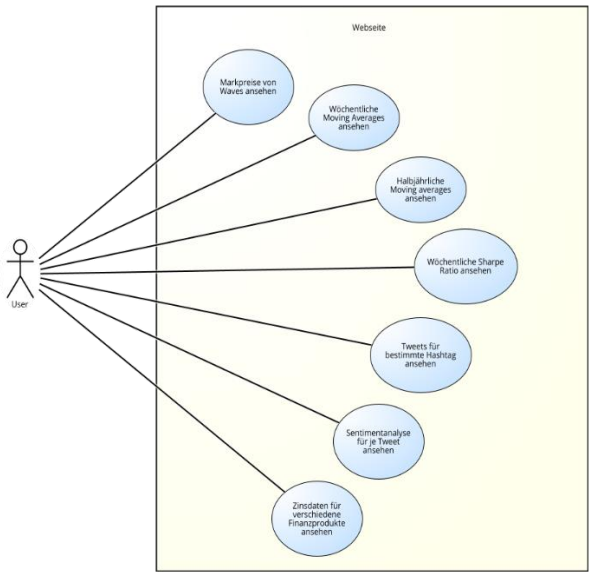

Abbildung 1

Veranschaulicht Reihe von Ereignissen, die auftreten, wenn ein Akteur ein System zum Abschließen eines Prozesses verwendet

C. Ausgewählte Architektur

Wir haben uns für die Architektur Model, View, Controller (MVC) entschieden, wobei Model die Anwendungs-/Geschäftslogik ist, die den aktuellen Zustand speichert und alle Berechnungen durchführt. View soll den aktuellen Zustand anzeigen und der Controller ist dann für die zu steuernde Applikation und die zu vermittelnde Kommunikation zwischen Model und View da.

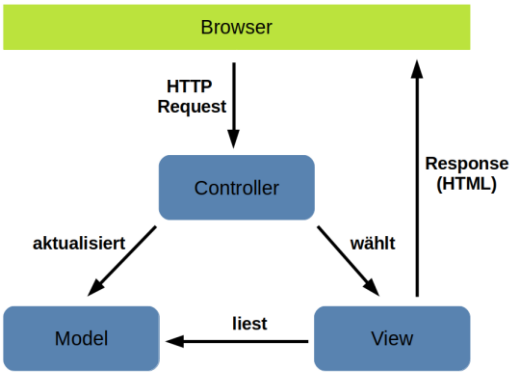

Abbildung 02

Veranschaulicht die Verfahrensweise eines MVC

Der Controller verarbeitet die HTTP-Anfragen, führt eine Datenvalidierung durch und fordert den Model dann eine Aktualisierung auf. Die Geschäftslogik benachrichtigt die View über Änderungen, die dann als dynamische Seite angezeigt werden.

Obwohl die Implementierung dieser Architektur Aufwendig ist, gibt es zahlreiche Vorteile daran.

Einige davon sind:

- Die Arbeitsteilung ist leicht möglich,
- Die Erweiterbarkeit der Funktionalität ist leicht,
- Der Austausch von einzelnen Komponenten ist einfach.

Des Weiteren haben wir uns bei der Entwicklung des Backends für "Java" mit dem Framework "Spring Boot" entschieden. Zum Starten des Projekts ließen wir den Code unter https//start.spring.io/ generieren. Dabei ist es direkt mögliche die nötigen dependencies hinzuzufügen. Für das Frontend fiel unsere Wahl auf "html" mit "Thymeleaf" und "Bootstrap".

D. Beschreibung der Implementierung

- Die aktuellen Preise für die Waves/USD:

  Zuerst wird eine Verbindung mit der API (https://wavescap.com/api/asset/WAVES.json) hergestellt, dann wird eine if-else-Statement durchgeführt, um herauszufinden, ob die Verbindung hergestellt wurde oder nicht. Wenn die Verbindung reibungslos war, werden alle Variablen der Klassen Waves und PricesList mithilfe des Parser initialisiert. Die Objekte der Klassen sind mit der Methode getPricesList() aufzurufen.

- Moving Average für die Waves/USD (1)

  SMA: Simple Moving Average

  d: Tage für die das MA berechnet werden soll.

  p(x) = Letzter Preis an dem Tag x

  SMA(d) = (p (1) + p (2) + ... + p(d)) / d

  Beispiele:

  - 7 Day Simple Moving Average SMA (7) = (p (1) +p (2) +p (3) +p (4) +p (5) +p (6) +p (7)) / 7
  - 21 Day Simple Moving Average SMA (21) = (p (1) +p (2) +...+ p (21)) / 21

- Sharpee Ratio für die Waves/USD (2)

  Es ist ein Maß für die Rendite in Relation zum Risiko, um die Rendite einer Investition im Vergleich zu ihrem Risiko zu verstehen und damit eine sinnvolle Kaufentscheidung zu treffen.

  Für die Berechnung des Sharpee Ratio wurde folgende Formel verwendet.(3)

$$S = \left(\frac{R_p - R_f}{\sigma_p} \right)$$

Abbildung 03

Formel für Sharpee Ratio

Wobei:

R_p = Rendite des Assets (Waves)

R_f = Risikofreie Rendite (U. S. Staatsanleihe)

σ_p = Volatilität von Waves

Die Zeiträume der R_f und R_p müssen mit dem Zeitraum der Rendite übereinstimmen. Bei dem Sharpee Ratio für Wavesmashup werden die Zeiträume auf 7 Tage eingesetzt.

- Schritte der Berechnung:

1- Zuerst wird die Rendite des Assets berechnet:

Rendite= ((Verkaufspreis - Einkaufspreis in bestimmten Zeitraum) * 100) / Einkaufspreis

2- Risikofreie Rendite festsetzen:

Es ist ein Zinssatz, der ohne Risiko erzielt werden kann, bei dem also nach Auffassung der Marktteilnehmer kein Risiko besteht, dass es zu Verzögerungen oder Ausfällen bei Zinszahlungen und Rückzahlung allgemein kommt. Im August 2019 lag der risikolose Zinssatz für zehnjährige Anleihen in der Eurozone bei – 1,95%, der Umlaufrendite der entsprechenden Bundesanleihen.

Die 10-jährige risikofreie Rendite wird in der Regel als Indikator für viele wichtige Finanzangelegenheiten verwendet und daher wurde bei der Berechnung der risikofreien Rendite bei WavesMashup betrachtet.

3- Alsl letztes wird die Volatilität des Waves über denselben Zeitraum berechnet:

Die Formel für die Berechnung lautet:

Die Wurzel aus: (1/n) *((a-i) ²+(b-i) ²)

Wobei:

- Die Variable n ist die Anzahl der enthaltenen Kurswerte innerhalb des betrachteten Zeitraumes.
- Das i ist der Durchschnittswert
- a und b sind an dieser Stelle die Kurswerte. Die Kurswerte werden immer Anfang des Monates und am Ende des Monats notiert.

Beispiele:

1. Monat Anfang: 20€ 1. Monat Ende:30€ Volatilität des Assets: 10%
2. Monat Anfang: 20€ 2. Monat Ende: 16€ Volatilität des Assets -4%
3. Monat Anfang: 30€ 3. Monat Ende: 36€ Volatilität des Assets: 6%

Es ergibt sich eine durchschnittliche Volatilität von i = 12/3= 4%

Die Werte in die Formel eingesetzt würden Folgendes ergeben:

Die Wurzel aus (1/3) *((20%-4%) ²+(-4%-4%) ²+(6%-4%) ² =

Die Wurzel aus (1/3*((256%) +(64%) +(4%)) =

Die Wurzel aus (1/3*(322%)) =

Die Wurzel aus 107.3 = 10.4%

E. Sentimentanalyse

Mithilfe der Klasse TweetsSearcher werden alle Tweets, die mit einem bestimmten Hashtag verknüpft sind, durchsucht und anschließend in einer Liste vom Datentyp (String) gespeichert.
Die in der Liste gespeicherten Texte werden einzeln von der Klasse TextAnalyzer analysiert und dann in einer Liste vom Datentyp (TextSentiment) gespeichert.
Der User kann dann Sortierte Informationen der Twitterstimmung genießen Z.B.

- Den Quelltext.
- Wie positiv, negativ oder neutral der Text verfasst ist.
- Auf welcher Sprache wurde der Text verfasst.

F. Waves‘ Zinsdaten für verschiedene Finanzprodukte

Nach der Verbindung mit der API (https://dev.pywaves.org/neutrino/json) werden die Daten in Form von der Klasse WavesInterestDto in einer Liste gespeichert und mit Hilfe des Controllers zurückgegeben.

Der beste Waves‘ Zinsdaten für die verschiedene Finanzprodukte werden in Form einer Tabelle verglichen und anschließend sortiert.

[1] https://lessonsfinancial.com/2020/04/16/edu-campaign-six-indicators/?gclid=Cj0KCQiAuP-OBhDqARIsAD4XHpczCLmZ_br29FgnfStX5Pl7UwSSQVV2DUccDCl185GJ2sUUgLeLOoEaAr-KEALw_wcB

[2] https://www.investopedia.com/terms/s/sharperatio.asp

[3] https://finance.yahoo.com/quote/%5ETNX/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAMXy7y6zQ9mSt63uBnmK2HQpd15G8ub6FkRrW43PSgDeu0qfGgYEktQhfm3K7gc9YNcZeTKtaYaVH4geAGzuMBCBrh24aKKY_v0BAzKzs4j6N3puv16NM9K6ZWUhYuVx5k5UqHHqVwCGriJmaClguVQmKDlx8KStgZISJd9xUM_8