



M122 LB2

Yahoo Finance API

Inhalt

Einleitung	2
Planung.....	2
Anforderungen.....	2
Meine Vorkenntnisse.....	2
Umsetzung	2
Reflexion	5

Einleitung

Diese Dokumentation beschreibt die Anforderungsdefinition des Projektes sowie das Lösungsdesign in Form von meiner Idee zum Projekt.

Planung

Anforderungen

Geplant ist, die Yahoo Finance API zu verwenden, welche ich auf der rapidapi Webseite gefunden habe. Somit kam ich zur Idee, meinen PDF-Report über eine bestimmte Aktie zu erstellen. Das PDF sollte den Aktuellen Preis, die Preisveränderung der letzten 24 Stunden sowie die Preisveränderung des letzten Jahres beinhalten. Ebenfalls wird eine Firmenbeschreibung hinzugefügt, welche ebenfalls beim API-Request mitkommt.

Meine Vorkenntnisse

Persönlich habe ich keine Vorkenntnisse mit Python, weshalb das Projekt eine neue Erfahrung für mich sein wird.

Umsetzung

Zu Beginn hatte ich die API in mein Projekt zusammen mit dem Code Snippet, welches rapidapi schon zur Verfügung stellt begonnen.

```
import http.client

conn = http.client.HTTPSConnection("yh-finance.p.rapidapi.com")

headers = {
    'X-RapidAPI-Host': "yh-finance.p.rapidapi.com",
    'X-RapidAPI-Key': "ac50f0a264mshb4d50d8737b1a85p178f48jsne098af8d0093"
}

conn.request("GET", "/stock/v2/get-summary?symbol=TSLA&region=US",
headers=headers)

res = conn.getresponse()
data = res.read()
```

Nachdem der Request funktionierte, hatte ich mich an das Generieren der PDF-Datei gewendet. Dies hatte ich mit FPDF gelöst, welches sich aber im späteren Verlauf geändert hatte.

```
from fpdf import FPDF

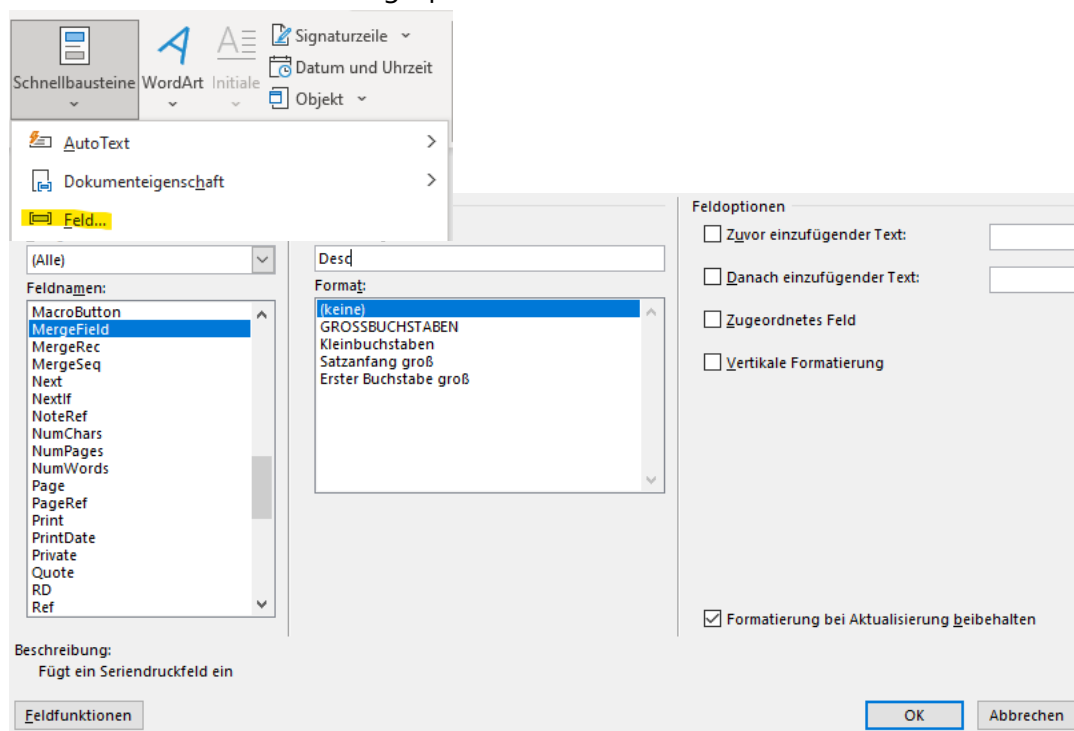
pdf = FPDF()
pdf.add_page()
pdf.set_font("Arial", size = 20)
pdf.cell(200, 10, txt = "Test Title", ln = 1, align = 'C')
pdf.set_font("Arial", size = 14)
pdf.cell(200, 10, txt = "Test Text", ln = 2, align = 'L')

pdf.output("test.pdf")
```

Anschliessend habe ich alle nötigen Werte, welche vom API Output benötige in einzelne Variablen gespeichert.

```
json_data = json.loads(data)
# Put data in variables
title = json_data['price']['shortName']
price = json_data['price']['regularMarketOpen']['fmt']
one_day_change = json_data['price']['regularMarketChangePercent']['fmt']
one_year_change = json_data['defaultKeyStatistics']['52WeekChange']['fmt']
desc = json_data['summaryProfile']['longBusinessSummary']
market_price = json_data['price']['regularMarketPrice']['raw']
pre_market_price = json_data['price']['preMarketPrice']['raw']
difference_price = market_price - pre_market_price
```

Da ich mit dem Designen mit FPDF nicht zufrieden war, hatte ich eine alternative gesucht. Somit kam die Idee, eine Word Datei als Vorlage zu erstellen und diese Vorlage dann mit den nötigen Daten zu befüllen. Bei den Word Files mussten Schnellbausteine eingefügt werden, die einen Bestimmten Text beinhalten. Mit diesem Text werden dann die einzelnen Schnellbausteine im Code angesprochen.



Um die Felder zu befüllen, muss somit eine Funktion erstellt werden, welche diese ersetzt. In der gleichen Funktion wird ebenfalls das neue Word File generiert und dazu ein PDF. Das Generierte Word File wird anschliessend gelöscht, da dies keine Verwendung mehr hat.

```
def create_report(title, price, one_day_change, one_year_change, desc,
difference_price):
    reportTemplate.merge(
        Title=title,
        Price=price,
        oneDayChange=one_day_change,
        oneYearChange=one_year_change,
        differencePrice="{:.2f}".format(difference_price),
        Desc=desc)
    reportTemplate.write('D:/TBZ/M122/Python/Projekt/report/report.docx')
    convert('D:/TBZ/M122/Python/Projekt/report/report.docx',
'D:/TBZ/M122/Python/Projekt/report/report.pdf')
    os.remove('D:/TBZ/M122/Python/Projekt/report/report.docx')
```

Da also das Generieren der PDF-Datei erfolgreich ablief, musste noch der Upload zum FTP implementiert werden. Ich hatte also ein Konto auf bplaced erstellt und mich somit an den Code gewendet. Dies war ziemlich einfach, da ich das Ganze vom Internet finden konnte und es für mich sehr verständlich war.

(Quelle: <https://stackoverflow.com/questions/12613797/python-script-uploading-files-via-ftp>)

```
# FTP upload
session = ftplib.FTP_TLS('merttakil.bplaced.net', 'merttakil', 'Passwort')
# file to send
file = open('D:/TBZ/M122/Python/Projekt/report/report.pdf', 'rb')
# send the file
session.storbinary('STOR report.pdf', file)
# close file and FTP
file.close()
session.quit()
```

Für den Mailversand ist der Sandbox Service von mailtrap verantwortlich. Mit den mitgegebenen Codebeispielen kann sehr einfach ein E-Mail inklusive Anhang gesendet werden.

An example of boarding pass

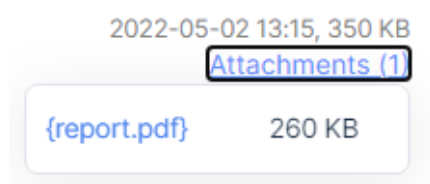
From: <mailtrap@example.com>
To: <new@example.com>

Show Headers

2022-05-02 13:15, 350 KB
Attachments (1)

HTML HTML Source **Text** Raw Spam Analysis Tech Info

Hallo Test!
Hiermit sende ich dir den Report zur Tesla Aktie.
LG Mert



Reflexion

Dieser Auftrag war einfacher als ich das erwartet hatte. Ich hatte jedoch zu Beginn eine Weile gebraucht, Python Syntax zu verstehen, doch trotzdem funktionierte das Projekt so wie ich es mir gewünscht hatte.

Alles in Allem war mein Projekt für mich ein Erfolg und sehr lehrreich.