



Sarah Treyer AP19b








cybernetic.vintage

Profil bearbeiten 


223 Beiträge 856 Follower 653 abonniert

cybernetic vintage
Vintage Store


 hand-picked clothing
 switzerland 
 DM to buy
depop.com/cybernetic_vintage




AVAILABLE





YOU




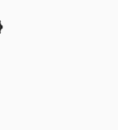
FEEDBACK


 **BEITRÄGE**


 REELS

 GESPEICHT

 MARKIERT







Ich habe auf der Seite rapidapi, die Sie in den Chat geschickt haben, auch Instagram API's gefunden, jedoch waren alle kostenpflichtig:
(<https://rapidapi.com/premium-apis-premium-apis-default/api/instagram85/>)

<https://www.instagram.com/vintage.cybernetic/?a=1>

[illegible]

Was kann ich mit der API tun?

Ich kann den JSON String auslesen und erhalte so unzählige Informationen über den Instagram Account sowie die einzelnen Posts vom Account.

Bsp. über den Account:

```
"show_suggested_profiles"  
"restricted_by_viewer"  
"blocked_by_viewer"  
"external_url" -> Link den ich zu meinem Depop Acc hinterlegt habe  
"edge_followed_by" -> Anzahl Followers  
"fbid"s  
"followed_by_viewer"  
"full_name" -> Voller Name den ich angegeben habe  
"highlight_reel_count"  
"category_name" -> Kategorie, bei mir Vintage Store  
"profile_pic_url_hd"  
"username"  
"biography"
```

Bsp. über einen Post:

```
"taken_at_timestamp"  
"comments_disabled"  
"edge_media_to_caption":{"edges":[{"node":{"text" -> Beschreibung  
"has_public_page"  
"is_video"  
"config_width" -> Bildbreite  
"config_height" -> Bildhöhe  
"edge_media_to_tagged_user":{"edges" -> tagged Benutzer  
"edge_media_preview_like":{"count" -> Anzahl Likes
```

Aufgabenstellung

Einen Report im PDF-Format mit Informationen über die letzten 12 Posts von meinem Instagram «vintage.cybernetic» Account erstellen. Dieser soll etliche Informationen über die einzelnen Posts (ich mache immer für jedes Kleidungsstück einen Instagram Post) aufweisen, welche ich unter dem Punkt Anforderungen genauer definiere. Dann lade ich meinen Report auf einem FTP hoch und sende ihn an einen Empfänger (mich selbst).

Anforderungen an den Report

- Anzahl Likes per Post.
- Bild vom Post.
- Welches war der beste Post (mit Anzahl Likes und ob es schon verkauft ist).
- Welches war der schlechteste Post (mit Anzahl Likes und ob es schon verkauft ist).
- Wie viele Artikel sind schon verkauft, wie viele nicht.

Abläufe

1. JSON von der Website holen
2. JSON auslesen und benötigte Daten herausfiltern
3. Daten aufbereiten
4. PDF erstellen und Daten anzeigen
5. Report auf den FDP hochladen
6. Report als E-Mail versenden

Report Mockup

Cybernetic Vintage Report

10. April 2022



Likes: 10
Verkauft



Likes: 10
Verkauft



Likes: 10
Verfügbar



Likes: 10
Verfügbar



Likes: 10
Verfügbar



Likes: 10
Verkauft



Likes: 10
Verfügbar



Likes: 10
Verkauft



Likes: 10
Verfügbar

Anzahl verkauft: 4
Anzahl nicht verkauft: 7

Bester Artikel: schwarze Hose, 10 Likes
Schlechtester Artikel: grünes T-Shirt, 4 Likes

Implementierung API & Datenaufbereitung

Um die benötigten Daten zu holen habe ich 3 Funktionen:

getData()

Holt die Daten von der JSON API. Leider ist es jedoch so, dass wenn man die API zu oft aufruft, dann gibt es einen Fehler. Darum habe ich in einer .txt Datei den gesamten JSON String von der API hineinkopiert, um so uneingeschränkt testen zu können.

Mann muss einfach die jeweiligen Zeilen ein- auskommentieren (mit URL -> 115, 116, 117, mit txt-File -> 120, 121)

```
113 def getData():
114     # WITH URL
115     url = "https://www.instagram.com/vintage.cybernetic/?__a=1"
116     test = urllib.urlopen(url)
117     parsed = json.load(test)
118
119     # WITH TEXT FILE
120     with open('/Users/sarah/Desktop/api.txt') as jsonFile:
121         parsed = json.load(jsonFile)
```

Danach fülle ich die Daten in einem Array ab:

- Bild-Link
- Beschreibung (wenn ich einen Artikel verkauft habe ändere ich die Beschreibung immer zu SOLD, so kann ich damit herausfinden ob der Artikel verkauft ist oder noch verfügbar ist.)
- Anzahl Likes

statisticData(data)

Hier rechne ich alle zusätzlichen Informationen aus, wie z.B. welcher Post hat die meisten Likes, wie viele Artikel wurden verkauft etc. und gebe diese in einem Array zurück.

```
177 result = {
178     "abiggestLikeUrl": biggestLikeUrl,
179     "abiggestLike": biggestLike,
180     "alowestLike": lowestLike,
181     "alowestLikeUrl": lowestLikeUrl,
182     "aWorstUrl": artikelWorstUrl,
183     "aWorst": artikelWorstLike,
184     "aBestUrl": artikelBestUrl,
185     "aBest": artikelBestLike,
186     "aNotSold": artikelNotSold,
187     "aSold": artikelSold,
188     "goodWeek": goodWeek,
189 }
```

saveIMG(data)

Hier speichere ich, anhand des Bild-Links aus dem der getData() Funktion, alle Bilder der Posts (nur das erste Bild eines Posts) im «images» Ordner, damit ich diese nachher einfach in meinem PDF einbauen kann. Der Grund für den Images Ordner ist, ich sonst immer einen Fehler bekommen habe, dass das Bildformat inkorrekt ist. Ich gebe eine List mit den neuen Bildpfaden zurück.

Damit der neue Bildername eindeutig ist setze ich ihn mit «image» + urlLink – 22 Zeichen + «.png» zusammen (siehe Zeile 123).

➔ image25B5CF0&_nc_sid=7bff83.png

```
122     urlLinkImage = x[0]
123     name = "image" + urlLinkImage[-22:] + ".png"
124     urllib.urlretrieve(urlLinkImage, name)
125     img = Image.open(name)
126     img.save("images/" + name)
```

PDF erstellen

Um ein PDF zu erstellen, benutze ich pyFPDF, eine Library um PDF Dokumente zu generieren.

Class PDF(FPDF)

Ich habe auch eine Class für meine PDF Funktionen erstellt. Diese wäre jedoch nicht unbedingt nötig gewesen, aber es erleichtert die Übergabe der Parameter um z.B. eine Bild zu erstellen.

(Nur noch 3 Parameter anstatt 6)

```
32     pdf.imagex(x, height, width)
```

```
12
13     class PDF(FPDF):
14         def imagex(self, url, height, width):
15             self.image(name=url, x=width, y=height, w=2500 / 80, h=2500 / 80, type='', link=url)
16
```

Um Text hinzuzufügen ist das nicht nötig, da hat es sonst auch nur 3 Parameter.

```
54     pdf.text(25, 190, 'Auswertung:')
```

addDataToPDF(data, pictureUrls, statistics)

In dieser Funktion erstelle ich das PDF und füge alle Bilder und Texte hinzu. Hierzu benötige ich die Daten aus den Funktionen getData, statisticData, saveIMG.

Gespeichert wird das PDF in meinem Projektfolder unter dem Namen «instagramReport.pdf», falls Sie auch eines generieren möchten, bitte den Pfad entsprechend anpassen.

```

14 pdf = PDF(orientation='P', unit='mm', format='A4')
15 pdf.add_page()
16 pdf.set_font('Arial', 'B', 20)
17
18 pdf.text(15.0, 25.0, 'Cybernetic Vintage: Letzte 12 Artikel')
19 pdf.set_font('Arial', 'B', 10)
20

```

Auf Zeile 14 erstelle ich das PDF. Und mit pdf. kann ich nun alle Texte, Bilder, Textart -grösse -farbe etc. hinzufügen. Die Anzeige der 12 Bilder habe ich mit einem Loop gelöst.

FTP

Bplaced & FileZilla

Ich habe Bplaced und FileZilla verwendet um einen Filedownload für meinen Report zu erstellen. Auf Bplaced habe ich mein Verzeichnis erstellt und mithilfe von FileZilla mein File hochgeladen.

Bplaced:

Übersicht FTP-Zugänge

ANZEIGE PRO SEITE 10 | 50 | 200

SUCHE

Benutzername	Heimverzeichnis	Letzter Login	Status	Aktion
sarahtreyer	/	03.05.2022, 15:56 Uhr	aktiv	⚙️ ✕

1 bis 1 von 1 Einträgen

SEITE 1

FileZilla:

sarahtreyer@sarahtreyer.bplaced.net - FileZilla

Server: sarahtreyer.bplaced Benutzername: sarahtreyer Passwort: Port: Verbinden

Status: Empfangen Verzeichnisinhalt für "/"...
 Status: Anzeigen des Verzeichnisinhalts für "/" abgeschlossen
 Status: 'instagramReport.pdf' wird in '/www/instagramReport.pdf' umbenannt
 Status: Empfangen Verzeichnisinhalt für "/"...
 Status: Anzeigen des Verzeichnisinhalts für "/" abgeschlossen
 Status: Empfangen Verzeichnisinhalt für "/"...
 Status: Anzeigen des Verzeichnisinhalts für "/" abgeschlossen

Lokal: /Users/sarah/Desktop/ Server: /www

Dateiname	Dateigröße	Dateityp	Zuletzt geändert
Unbenannt-7_Zeichen...	192018	png-Datei	29.03.2022 14:1...
Unbenannt-900_Zeic...	3934524	png-Datei	05.04.2022 16:3...
Windows 10	796	Datei	17.11.2019 19:41:...
api.txt	291309	txt-Datei	03.05.2022 14:0...
desktop.ini	282	Properties	04.05.2020 11:1...
drawing.png	463555	png-Datei	28.03.2022 09:1...
instagramLog.pdf	9871634	pdf-Datei	03.05.2022 14:5...
rose-tribal.ai	1670913	Adobe Illustrator...	24.04.2022 19:0...
visitenkarte.ai	309572	Adobe Illustrator...	03.04.2022 21:0...
visitenkarte_Zeichenf...	24807	png-Datei	03.04.2022 21:0...
visitenkarte_rückseit...	605571	Adobe Illustrator...	04.04.2022 14:3...

Dateiname	Dateigröße	Dateityp	Zuletzt g	Berechtigung	Besitzer/Gruppi
..					
live access.php	67	PHP	03.0...	0604	31291 5000
instagramReport.pdf	9077067	pdf-Datei	03.05...	0604	31291 5000
instagramLog.pdf	9871634	pdf-Datei	03.05...	0604	31291 5000
PgAdmin.php	72	PHP	03.0...	0604	31291 5000
MyAdmin.php	72	PHP	03.0...	0604	31291 5000

1 Datei ausgewählt. Gesamtgröße: 9871634 Bytes

Server/Lokale Datei	Richtung	Datei auf Server	Größe	Priorität	Zeitpunkt
sarahtreyer@sarahtreyer...					
/Users/sarah/Desktop...	-->	/www/instagramLog.pdf	9871634	Normal	03.05.2022 1...
/Users/sarah/Desktop...	-->	/../instagramLog.pdf	9871634	Normal	03.05.2022 1...
/Users/sarah/Desktop...	-->	/instagraml oa.pdf	9871634	Normal	03.05.2022 1...

Um hier ist die URL um meinen Report herunterzuladen:

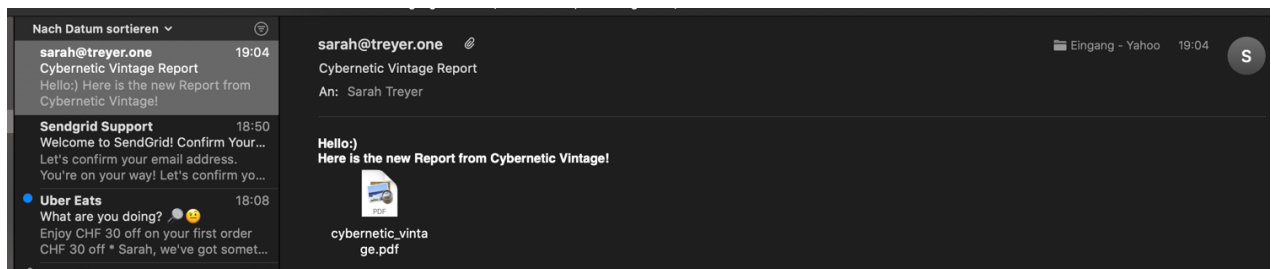
<http://sarahtreyer.bplaced.net/instagramReport.pdf>

E-Mail verschicken Sendgrid

Um den Report automatisch in einem E-Mail zu verschicken habe ich, wie empfohlen, den Service Sendgrid verwendet. Dafür habe ich einen Account erstellt und eine Single Sender Verification gemacht um einen gültigen SENDGRID_API_KEY zu erhalten. Für diese Verification habe ich aber eine eigene kostenpflichtige E-Mail als Absender gebraucht, was ich zum Glück hatte «sarah@treyer.one» (meine yahoo E-Mail wurde nicht akzeptiert).



Danach habe ich mithilfe der Dokumentation von Sendgrid und etwas googeln die passende Funktion `sendMail()` im Code hinzugefügt. Bis die E-Mail ankommt geht es zwar immer einige Sekunden aber es klappt & man kann den Report herunterladen.



```
201 def sendMail():
202     message = Mail(
203         from_email='sarah@treyer.one',
204         to_emails='sarah_treyer@yahoo.com',
205         subject='Cybernetic Vintage Report',
206         html_content='<strong>Hello:) <br> Here is the new Report from Cybernetic Vintage!</strong><br>')
207
208     with open('/Users/sarah/Desktop/instagramLog.pdf', 'rb') as f:
209         data = f.read()
210         f.close()
211     encoded_file = base64.b64encode(data).decode()
212
213     attachedFile = Attachment(
214         FileContent(encoded_file),
215         FileName('cybernetic_vintage.pdf'),
216         FileType('application/pdf'),
217         Disposition('attachment')
218     )
219     message.attachment = attachedFile
220
221     try:
222         sg = SendGridAPIClient(SENDGRID_API_KEY)
223         response = sg.send(message)
224         print('Success! Status: '+response.status_code)
225     except Exception as e:
226         print('Error! ± e.message)
```

Bedienungsanleitung

Falls Sie das Programm testen möchten, müssen Sie noch ein paar Dinge anpassen:

1. Wählen Sie die Methode wie Sie die Daten holen wollen (.txt oder per URL) wie in der Implementation beschrieben (in der getData() Funktion)
2. In der Funktion sendMail() müssen Sie den Arraykey «to_emails» auf Ihre E-Mail ändern.
3. Klicken Sie auf den grünen Playbutton um das Skript zu starten.
4. Wenn die Console «Success! Status:202» ausgibt hat alles geklappt.

Reflektion

Das Projekt lief gut. Alle Ziele/Anforderungen, die ich mir vorgenommen habe, ich erfüllt und ich konnte es innerhalb einem angemessenen Zeitrahmen umsetzen. Jedoch habe ich mich etwas zu sehr auf den Programmierteil vom Erstellen des PDF's konzentriert und die meiste Zeit investiert. Ich hätte vielleicht noch etwas mehr Optionen der API's, Librarys und Services ansehen sollen. Bin aber mit der Wahl der API zufrieden, denn Sie erfüllt genau das was ich wollte und ist nicht kostenpflichtig. Trotzdem würde ich bei einem neuen Projekt vielleicht eine etwas Komplexere API nehmen. Die FTP Dateiablage ist hoffentlich korrekt so, ich habe kein Tutorial für einen automatischen Fileupload mit Python gefunden und bin einfach der Anleitung auf der bplaced Website gefolgt. Mit ServiceGrid eine Mail zu versenden ging erstaunlich einfach. Das ich das PDF mit pyFPDF gemacht habe war etwas umständlich, da ich relativ viele Loop und Wiederholungen im Code habe. Vielleicht hätte es da noch einen eleganteren Weg gegeben. Und leider habe ich das Commiten etwas vernachlässigt, nächstes Mal werde ich die Verbindung gleich am ersten Tag erstellen und so habe ich es immer wieder verschoben. Zum Schluss bin ich mit meinem Report zufrieden und kann ihn sicher auch noch später brauchen.

Der Report

**SYNETIC
JUNGE**



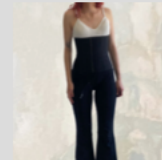
zu haben (Likes 9)



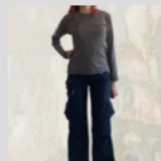
zu haben (Likes 11)



Verkauft (Likes 18)



Verkauft (Likes 16)



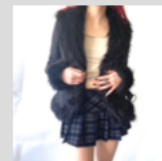
Verkauft (Likes 16)



zu haben (Likes 11)



zu haben (Likes 16)



Verkauft (Likes 29)



zu haben (Likes 34)



Verkauft (Likes 21)



zu haben (Likes 26)



zu haben (Likes 8)

Meiste Likes: 34



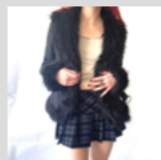
Wenigste Likes: 8



Anzahl verkaufte Artikel:
5 Artikel

Anzahl nicht verkaufte Artikel:
7 Artikel

Bester Artikel: 29 Likes



Schlechtester Artikel: 8 Likes



Wochen Status

