Webscraping met Python

februari 3, 2021 · Software

Auteurs

ZEH Otten

Webscraping is de techniek om geautomatiseerd gegevens van het internet te vergaren. Zonder browser!
Stel je voor dat een grasmaairobot weet wat de weercondities zijn en dat het voorlopig droog genoeg blijft om het gras te gaan maaien? De benodigde informatie die nodig is om deze beslissing te nemen is op allerlei (weer)websites te vinden. De robot hoeft alleen maar de

informatie te web scrapen.

Ik denk dat het volgende voorbeeld van webscraping veel duidelijk maakt en als inspiratie kan dienen om onze robots en onze programma's 'iets intelligenter' te maken.

Probleem
In mijn woonplaats heeft elk huishouden minstens een viertal afvalcontainers; een voor plastic afval, een voor GFT (composteerbaar) afval, een voor papier en een voor 'rest' afval. Elke container wordt minstens een keer per maand geleegd. Soms op maandag, soms op dinsdag en ook op donderdag, mits er geen feest- of vrije dagen zijn. Dan wordt het weer anders.

Je begrijpt het al. Ik raak in de war van de 'regelmaat' en het gebeurt dus dat ik te laat ben met het aanbieden van mijn afval. De hulp van een jaarlijkse 'afvalkalender' is weliswaar een oplossing maar hoe leuk is het om dit probleem te laten oplossen door een 'web scraping' programma, geschreven in Python?

Doel van het scrape programma
Ik wil een dag van te voren een berichtje ontvangen op mijn telefoon (mail, tweet of ander berichtje) waarin wordt vermeld welke afvalcontainer ik moet klaarzetten. In dit geval kies ik voor een e-mail naar robodomo@gmail.com omdat ik de procedure om mail te sturen al vaker in Python heb toegepast.

Werkwijze
Ik gebruik een Raspberry Pi Zero (circa 10 euro) om mijn Python programma op te draaien. Tevens heeft de Zero Wifi en daardoor internet toegang. Een volledige Python3 omgeving is standaard aanwezig op de Linux distributie Raspbian. Om het programma te schrijven kan ik kiezen uit meerdere editors, zoals Geanny , idle of Thonny.

Het programma zal via de Linux taakmanager 'crontab' één keer per dag worden aangeroepen om te checken of er een mail moet worden gestuurd.

Ik gebruik de website van de lokale afval verwerkingsdienst die de afhaaldata ergens op hun website publiceert. Ik maak gebruik van een paar extra, maar zeer veel toegepaste python libraries:

Smtplib voor het sturen van e-mail, **urllib** voor het bezoeken van de website en voor het scrapen van data en **Beautiful Soup**. Deze laatste is geen standaard Python librarie en moet apart worden geïnstalleerd. BeautifulSoup helpt bij het formatteren en organiseren van html opgemaakte internet pagina's.

Beautiful soup, so rich and green,
Waiting in a hot tureen!
Who for such dainties would not stoop?
Soup of te evening, beautiful Soup!
(*gedicht van Lewis Carroll*)

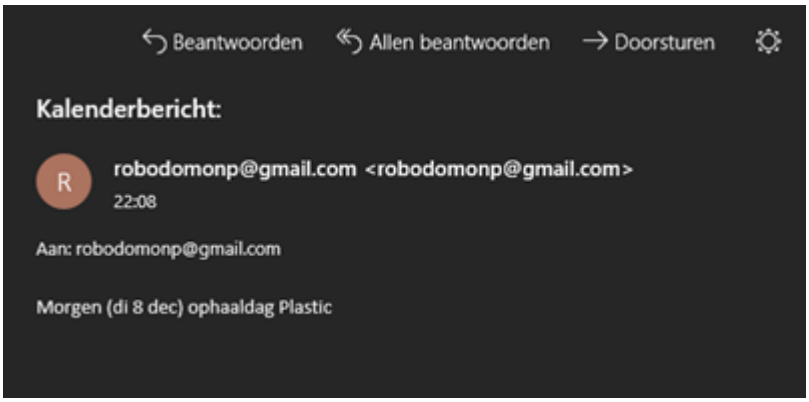
Op de website url = 'https://afvalkalender.dar.nl/adres/postcode:nummer’ is de gewenste data te vinden.
html = urlopen(url) opent de site en de regel bs = BeautifulSoup(html.read(), 'html.parser') leest de website in de variabele bs.
De regel s = bs.find_all(class_='date') zoekt in bs alle html <date> tags en plaatst die in s.
De procedure analyseer kijkt in de gevonden s om welke ophaaldata het gaat.
Indien het de volgende dag een ophaaldag is dan stuurt de procedure 'sendmail' mij een email.

Dit web scraping voorbeeld gaat ervan uit dat bekend is waar de gewenste data te vinden is op de website, namelijk achter de <date> tags.
De google chrome browser beschikt over een tool om html pagina's te bekijken (via de menu optie 'hulpprogramma's voor ontwikkelaars'). Die kun je gebruiken om gewenste data te vinden.

Het gehele programma ziet er dan als volgt uit:

```
#
# webscraping programma voorbeeld met beautiful soup !
# Webscraping met Python, Ryan Mitchell
#
# (c) Z.E.H. Otten
# november 2020
#

import datetime
import smtplib
from locale import setlocale, LC_ALL
from datetime import date, datetime, timedelta
from email.mime.text import MIMEText
from urllib.request import urlopen
from bs4 import Beautiful Soup

debugPrint = False
setlocale(LC_ALL, "nl_NL.utf-8")
url      =  'https://afvalkalender.dar.nl/adres/postcode:nummer'
html  = urlopen(url)
bs       = BeautifulSoup(html.read(), 'html.parser')
afval = ["Plastic","Rest-afval","GFT-afval","Papier"]
#-----------------------------------------
def sendMail(subject,body):
    msg = MIMEText(body)
    msg['Subject'] = subject
    msg['From']    = "mailadres"
    msg['To']      = "mailadres"

    s = smtplib.SMTP('smtp.gmail.com')
    s.ehlo()
    s.starttls()
    s.ehlo()
    s.login("mailadres", "password")
    s.send_message(msg)
    s.quit()
    return


#-----------------------------------------
def analyseer():

    s[0]=  str(s[0])[16:25]
    s[1]=  str(s[1])[16:25]
    s[2]=  str(s[2])[16:25]
    s[3]=  str(s[3])[16:25]

    #een dag van te voeren waarschuwen
    nextDay = datetime.now() + timedelta(days=1)
    if debugPrint: print (nextDay.strftime('%a %-d %b'))

    message = ""
    for d in range(4):
        if debugPrint: print (d, s[d], afval[d])
        if nextDay.strftime('%a %-d %b') == s[d]:
            message = message + "Morgen" +  " ("+ s[d] +") " + "ophaaldag " + afval[d]
            message = message + "\n"
        else:
            if debugPrint: print ('morgen geen ophaaldag')

    if message != "":
        sendMail("Kalenderbericht:",message)
        print (message)
        if debugPrint: print ('Mail gestuurd')
    return

#-----------------------------------------
# main loop

print ("Running scrape program...")
s = bs.find_all(class_='date')

if len(s) == 4:  # vier afvalstromen
    analyseer()
else:
    print ("error..")
```

Het resultaat van het programma: Indien de komende dag een afvalcontainer aan de weg moet worden gezet dan wordt dit met een email aangegeven:



Gerelateerd

Raspberry Pi (6): Let's e-mail en twitter met onze robots
21 december 2016
In "Raspberry Pi"

Eerste data uit Wifly in de browser
27 november 2012
In "Software"

Raspberry Pi (4): Luxe deurbel
10 oktober 2015
In "Electronica"

1 reactie

Dré Jansen
3 februari 2021

hoi Zeno,

inderdaad, een mooi programma, maar wij hier in de Hoeksche Waard hebben al zo'n app. keurig een dag van tevoren ontvang ik een mailtje met de juiste container die moet worden aangeboden.
dus ook bij feestdagen wordt op de afwijkende datum de herinneringsmail verzonden.

groeten, Dré