Scrapping stránek prakticky: Dívka dne na TN.cz

Obsah

Kdo se alespoň trochu zajímá o webové dění, tvoření MFA stránek a podobných „neužitečnostech“, tak jste určitě o scrappingu stránek slyšeli. Jedná se o dolování (mining) pro nás užitečných informací z již existujících stránek.
Samozřejmě, někdy se jedná o hotovou vědu, hlavně když jsou potřebné informace uložené v dost prasáckém nestrukturovaném složení.Ale my jsme šikovní kluci od bobří řeky a na všechno máme odpověď.

Praktický příklad scrapping-u v Python-u

V tomto článku si ukážeme, jak prakticky provádět scrapping v Python-u. Vše co k tomu budeme potřebovat je defaultní instalace Python a mou oblíbenou parsovací knihovnu BeautifulSoup, na prvotní analýzu HTML kódu Firefox s pluginem Firebug.

Úkol zní jasně: vybrakovat všechny obrázky Dívek dne ze stránky TN.cz.

Illustration 1: Příklad galerie dívky dne na TN.cz

Analýza galerie Dívky dne

Ještě prvně než se pustíme do psaní kódu si stránku pořádně prohlídneme a zanalyzujeme. Přikládám screenshot z obrazovky galerie (červeným je to co nás bude zajímat):

Illustration 2: Analýza galerie Dívky dne na TN.cz

Číslami jsou označeny odkazy (v HTML kódu je to tag A) na konkrétní galerií. Na každé stránce je pět odkazů na samostatné galerie, pro dalších pět musíme kliknout na link „další“. Stránek je celkem 30 co je zhruba 150 galerií a v každé z nich 6 obrázků.

Krok 1: zjištění URL adres všech stránek s galeriemi

V prvním kroku si zjistíme všechny URL adresy kde se nachází galerie. Začneme tedy od nejnovější galerie, zjistíme všech 5 odkazů na jednotlivé galerie a pak následujeme link označen „další“ a postup opakujeme. Konec prolézání zjistíme, ak žádný link s anchor textem „další“ se na stránce nenachází.

Výňatek kódu v Pythone, který používa BeautifulSoup na rozparsování odkazu „další“:

```
soup.find('div', {'class':'main_page_navigator_navigation'}).find(
     lambda tag: tag.name == 'a' and tag.contents[0] == u'další')
```

V skratke: najdeme tag <div class=“ main_page_navigator_navigation“> a v něm tag a, který obsahuje text „další“ (všimněte si, že do funkce find() můžeme dát jak název tagu který hledáme nebo rovnou funkci).

Další částí skládanky je rozparsování všech odkazů na galerie (na obrázku 2 označená číslicemi 1 až 5):

```
soup.find( 'ul', {'class':'galery_2'} ).findAll(lambda tag:
tag.name == 'a'  and u'Dívka dne' in tag.contents[0] )
```

Táhle část kódu nám vráti všechny odkazy na galerie, samotný odkaz uloží BeautifulSoup v atributu „href“.

Krok 2: získání jednotlivých obrázků z galerie

Takže máme všechny URL na jednotlivé galerie, nyní nám už chybí jenom samotné rozparsování galerie a získání všech obrázků. Není nic jednoduššího:

```
soup.find('div',
{'id':'red_section_girl_of_the_day_photogallery'}).findAll('img')
```

Tenhle kód nám z tagu <div id=“ red_section_girl_of_the_day_photogallery“> vrátí všechny obrázky. Samotný odkaz na obrázek je pochopitelně v atributu „src“.

Závěrem anebo naco je to vůbec dobrý?

Zde představený příklad představuje standardní postup v scrapování běžné stránky: rozparsovat jednotlivé podstránky a z podstránek získat potřebné informace (v případě dívky dne na TN.cz obrázky nahých slečen).

Na co je scrapování dobrý? V prvním řadě na získání informací ke vlastnímu, leckdy nekalému prospěchu (můžeme si scrapper dát do cronu, spojit z databází a postavit si vlastní zdroj RSS z jakékoliv stránky, získávat data na vlastní MFA weby, možnosti jsou neomezené, záleží jenom na fantazií).

Python se svou bohatou standardní knihovnou a rozšířením BeautifulSoup značně ulehčuje práci a programování scrapera bývá i často zábava...

V příštích dílech si můžeme něco povykládat o báječném rozšíření Pythonu `processing` a jak nám může řádově zrychlyt scrapování a využít naplno síťovou linku a naše zahálející multijádrové procesory.

Appendix A: Zdrojový kód scrapper.py

```
# -*- coding: utf-8 -*-


#
# Příklad skrapování obsahu "for fun & profit" ;-)
# Skrapování odkazů na všechny fotky Dívek dne ze serveru TN.cz
#
# Pro víc skvělých příkladů navštivte http://www.kiwwisk.cz
#


#
# Pozor! Pouze pro edukativní účely, kopírování obsahu bez svolení majitele je trestné.
#


import urllib2
from BeautifulSoup import BeautifulSoup
from time import sleep


def get_beautiful_soup( url, pocet_opakovani = 5 ):
    """get_beautiful_soup( url, pocet_opakovani )


        Vrati BeautifulSoup objekt ze zadane adresy (url),
```

Víc informací naleznete na http://www.kiwwisk.cz

```
        kdyz z nejakeho duvodu funkce urlopen() selze, zaspi na 1 s a
        opakuje ji pocet_opakovani krat
    """

    while True:
        try:
            stream = None
            stream = urllib2.urlopen( url )
            soup = BeautifulSoup( stream.read() )
            return soup
        except urllib2.URLError:
            pocet_opakovani -= 1
            if pocet_opakovani < 0:
                raise
            sleep( 1 )
        finally:
            if stream:
                stream.close()

def nasleduj_link_a_vrat_polivku( domain, url, funkce_s_podminkou ):
    """nasleduj_link_a_vrat_polivku( domain, url, funkce_s_podminkou )

    Iterator, kteri vraci BeautifulSoup objekt z adresy v tvaru domain+url a dale vyhleda
    dle funkce_s_podminkou() nasledujici link ktery navstivi a tak dale dokolecka.
    Kdyz funkce_s_podminkou() vrati False, cyklus konci.

    Funkce je vhoda na prechazeni html stranek ktere jsou nejakym zpusobem strankovane
    """

    while True:
        soup = get_beautiful_soup( domain + url )
        yield soup
        next_link = funkce_s_podminkou( soup )
        if next_link:
            url = next_link['href']
            continue
        break

def vrat_mi_url_galerie( start_url ):
    """vrati_mi_url_galerie( start_url )

    Iterator, ktery generuje url galeri s divkami dne.
    Ocakava startovaci url <- nejlepe nejnovejsi galerii.
    Z teto url pak pokracuje ke starsim tak, ze nasleduje link, ktery
    obsahuje text 'další'
    """
```

Víc informací naleznete na http://www.kiwwisk.cz

```
    for soup in nasleduj_link_a_vrat_polivku( "http://tn.nova.cz", start_url, lambda soup:
soup.find('div', {'class':'main_page_navigator_navigation'}).find(lambda tag: tag.name == 'a' and
tag.contents[0] == u'další') ):
        galerie = soup.find( 'ul', {'class':'galery_2'} ).findAll(lambda tag: tag.name == 'a'  and
u'Dívka dne' in tag.contents[0] )
        for link in galerie:
            yield link['href']


def vrat_mi_vsechny_obrazky( url ):
    """vrat_mi_vsechny_obrazky( url )

    Iterator, ktery postupne vrati vsechny obrazky z fotogalerie
    Divky dne na zadane url adese.

    Vrati jmeno divky, link k plnymu obrazku
    """

    soup = get_beautiful_soup( url )
    for img_link in soup.find('div', {'id':
'red_section_girl_of_the_day_photogallery'}).findAll('img'):
        yield img_link['title'], img_link['src'].replace('105x156', '750x750')


def main( ):

    for url in vrat_mi_url_galerie( "/bin/front.php?section_id=280&page=1" ):
        for jmeno, fotka_url in vrat_mi_vsechny_obrazky( "http://tn.nova.cz" + url ):
            print jmeno.encode('cp852', 'ignore'), fotka_url.encode('cp852', 'ignore')


if __name__ == "__main__":
    main( )
```