**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU ELEKTROTEHNIČKI FAKULTET**

**Sveučilišni studij**

**ALATI ZA UČITAVANJE NESTRUKTURIRANIH PODATAKA**

**Diplomski rad**

**Slaven Galić**

Mentor: doc.dr.sc. Josip Job

**Osijek, 2017.**

**SAŽETAK**

U ovom radu prezentirat će se se teme kao što su tipovi podataka po strukturi, njihovo učitavanje i prikazivanje pomoću gotovih usluga i biblioteka. Osim teorijskog dijela, u radu je prezentiran i konkretan primjer alata za učitavanje podataka iz nekog izvora. Kao rezultat ovakvog alata dobiveni su kvalitetni podatci koji su iskoristivi u praksi. Na temelju uspoređenih usluga, biblioteka i vlastitog alata može se vidjeti razlika u kvaliteti učitavanja nestrukturiranih podataka iz vanjskih izvora te njihovo spremanje u raznim formatima.

Ključne riječi: strukturirani/polustrukturirani podatci, nestrukturirani podatci, UIMA, web scraping

**SUMMARY**

# 1. UVOD

Porastom korištenja interneta javno dostupna količina podataka je dosegla velike razmjere. S tolikom količinom koja ima određenu snagu i potencijal, pretpostavlja se da bi na svako pitanje trebao postojati odgovor, ali sama dostupnost ne nužno donosi i riješenje. Eksplozija podataka je neizbježan trend s obzirom na veliki tehnološki napredak, ljudi i organizacije sve više ovise o računalnim uređajim i izvorima informacija na Internetu. Danas raspolažemo sa golemim količinama sveprisutnih podataka. Većina podataka se nalazi u nestrukturiranom ili polustrukturiranom obliku gdje se podatak nalazi na mjestu predviđenom samo za otvaranje i čitanje određenog dokumenta. Današnja interakcija uglavnom se temelji na dokumentima. Dokumente primamo i šaljemo, generiramo ih iz vlastitih sustava i aplikacija ili ih izrađujemo ručno. Nestrukturirani i nepohranjeni podatci smanjuju iskoristivost, mogućnost analize i korištenje tih dokumenata. Veći dio dokumenata je nestrukturiran i pohranjen u različitim formatima unutarnjih izvora kao što su papir, e-mail poruke, MS Word dokumenti, tehnička dokumentacija itd. Osim unutarnjih izvora, nestrukturirani podatci nalaze se i u vanjskim izvorima kao što su blogovi, forumi, internetske stranice te društvene mreže. Podatci koji se nalaze u ovim izvorima mogu odgovoriti na razna pitanja i pomoći pri rješavanju gorućih problema s kojima se susreću razni poslovni sustavi.

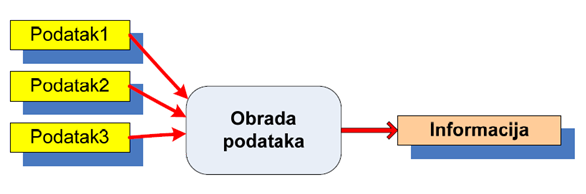
Za masivne količine strukturiranih i nestrukturiranih podataka upotrebljava se naziv „veliki podatci“ (eng. Big Data). Tako velike i kompleksne količine podataka gotovo je nemoguće obrađivati tradicionalnim alatima za upravljanje bazama podataka i aplikacijama za obradu. Koncept „velikih podataka“ ne odnosi se samo na velike opsege podataka, nego i na tehnologije potrebne za rad s velikim količinama podataka, njihovo prikupljanje i skladištenje. Prikupljanje podataka je jedna od bitnih stavki funkcioniranja cijelog koncepta. Podatci koji se nalaze na vanjskim izvorima prikupljaju se alatima za učitavanje nestrukturiranih podataka (eng. Web-Scraping tools). Učitavanje nestrukturiranih podataka je proces ekstrakcije informacija iz izvora koji nisu obrađivani i nisu strukturirani na pravilan način. Trenutno je najveći problem nedostatak prikladnih alata specijaliziranih za integraciju svih tehnologija potrebnih za izvođenje procesa analize podataka. Kompletan alat bi trebao sadržavati cjelokupni proces od pronalaženja točnih pojmova i zahtjeva, prikupljanja informacija, obrade i skladištenja u ispravnom obliku.

# 2. MODELI PODATAKA

## 2.1 PODATAK

„Poznata ili pretpostavljena činjenica na osnovi koje se oblikuje informacija. Sastoji se od skupa kvantitativnih parametara koji se mogu zapisati kao nizovi znakova ili nizovi brojeva. U računalima, koja su do nedavno nazivana i strojevima za automatsku obradu podataka, ti se nizovi, radi pohrane, obrade i sl. pretvaraju u nizove bitova. Podatak je nematerijalne prirode, on jednostavno postoji u našim mislima i nema značenje unutar ili izvan svog postojanja ili o samom sebi,  pa se pridružuje značenju kojim opisujemo svojstva objekata. Može postojati u bilo kojem obliku bio upotrebljiv ili ne. Oblici podataka su zvučni, slikovni, brojčani i tekstualni.“[[1]](#footnote-1) Podatak je jednostavna misaona činjenica koja u svom neobrađenom obliku ima nekakvo značenje. On sadrži opis svojstva nekog entiteta, zapažanja tokom nekog procesa, registrirane činjenice ili događaje. Sama struktura podatka je apstraktna jer sadrži više svojstava, značenje, vrijeme i vrijednost. Podatci koji su kombinirani u strukturi čine informaciju.

Riječ informacija potječe od lat. *Informatio* što znači predodžba, tumačanje, pojam. „Skup podataka s pripisanim značenjem, osnovni element komunikacije koji, primljen u određenoj situaciji, povećava čovjekovo znanje.“[[2]](#footnote-2) Informacija je skup logički povezanih podataka, odnosno organiziranih i obrađenih činjenica koje predstavljaju nekakvu vijest. Nakon što je interpretirana, stavljena u kontekst ili kad joj je dano značenje, ona postaje znanje. Od samog početka korištenja računala, obrada različitih vrsta podataka bila je jedan od osnovnih zadataka. U slučajevima kada je jako bitno donijeti kvalitetnu odluku, veliku ulogu igra organizacija podataka sa kvalitetnim informacijama. Informacije se mogu strukturirati kako bi imale određenu namjenu, taj proces se naziva strukturiranje informacija. Informacije su često strukturirane prema kontekstu u interakciji s korisnicima ili većim bazama podataka. Baza podataka je organizirana zbirka određene količine podataka.



**Slika 1.** Podatci nakon obrade postaju informacija

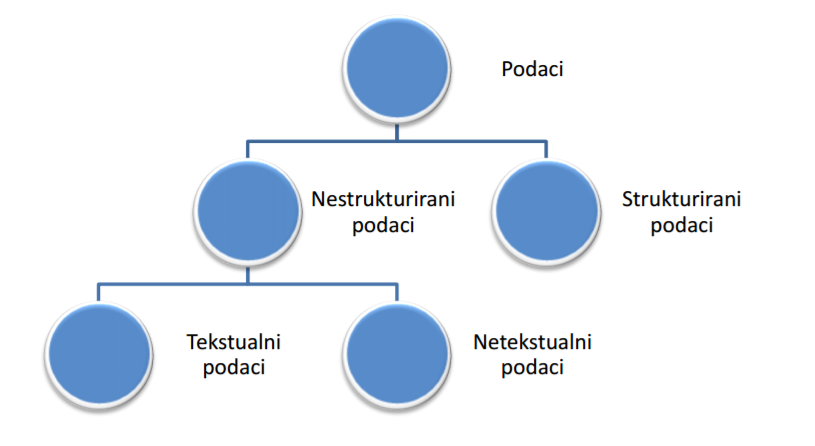
Tradicionalne baze podataka imaju dugotrajnu poziciju u većini procesa vezanih za jednostavno skladištenje podataka. Klasični relacijski model, gdje se baza sastoji od relacija, je star preko 40 godina, izumio ga je Edgar Frank Codd, 1970. godine u svom *“A Relational Model of Data for Large Shared Data Banks”*. Dizajn klasičnih relacijskih baza podataka nije se značajno mjenjao posljednjih desetak godina i danas sve teže može pratiti zahtjeve velikih količina podataka, tj. koncepta „velikih podataka“ s kojima se moraju nositi sadašnje i buduće aplikacije. Također, porastom broja korisnika Interneta sve više i više podataka se razmjenjuje putem mreže, a ti podatci nisu baš strukturirani te su heterogeni i često nepotpuni.

## 2.2. NESTRUKTURIRANI PODATCI

Nestrukturirani podatci su „sirovi“ i neorganizirani podatci. Nedostatak strukture čini kompilaciju vremenom i energetski zahtjevnim zadatkom. Nestrukturirani podatci dolaze u mnogim oblicima i veličinama. Može ih se pohraniti u dokumente, izvješća, proračunske tablice, web stranice ili digitalne medije(slike, audio i video). Prvi korak u obradi tih podataka je dokumentiranje, konsolidacija i upravljanje. Iako se većina baza podataka danas može nositi s nestrukturiranim podatcima, smjer industrije je razvoj aplikacija koje bi upravljale sadržajem baze i tako upravljale nestrukturiranim podatcima. Bilo bi korisno za tvrtku u svim poslovnim slojevima pronaći mehanizam analize podataka kako bi se smanjili troškovi nestrukturiranih podataka koji dodaje organizaciji.

Postupak dodavanja strukture nestrukturiranim podatcima sastoji se od:

* Upotreba uzoraka teksta kao što su redovni izrazi za male ili velike strukture
* Upotreba tabličnog pristupa za prepoznavanje zajedničkih odjeljaka
* Upotreba tekstualne analize kako bi se razumio tekst i povezao sa ostalim informacijama



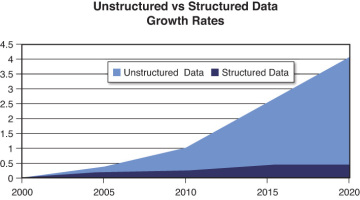
Slika 2. Prikaz vrste podataka

### 2.2.1. UIMA standard

UIMA (eng. Unstructured Information Management Architecture) je standard koji koristi arhitekturu upravljanja nestrukturiranim informacijama i komponenta je arhitekture i implementacije programskog okvira za analizu nestruktuririranog sadržaja kao što su tekstualni, video i audio podatci. Napravljen je od strane IBMa i pruža kompletnu programsku arhitekturu za razvoj, otkrivanje, sastav i implementaciju analitike za analizu nestrukturiranih podataka i integraciju sa tehnologijama pretraživanja. Motivacija za razvoj takvog prograsmkog okvira bila je izgradnja zajedničke platforme za nestrukturirane informacije, poticanje ponovne uporabe dijelova analize i smanjenja dupliciranja analize. Jednostavna UIMA arhitektura omogućuje uključivanje vlastitih dijelova analize i kombinaciju sa postojećim.

Glavni cilj je transformacija nestrukturirane informacije u strukturiranu informaciju koristeći analizu kako bi se otkrili entiteti i odnosi i izgradio model koji ima približno strukturiranu ili polu-strukturiranu arhitekturu.

### 2.2.4. BUDUĆNOST NESTRUKTURIRANIH PODATAKA



**Slika 3.** Prikaz rasta strukturiranih i nestrukturiranih podataka

Budućnost pripada nestrukturiranim podacima i vrijednim poslovnim spoznajama koje sadrži. Tvrtke trebaju razvijati i ažurirati svoje procese poslovne inteligencije kako bi uključile nestrukturirane podatke i otključali njegovu vrijednost. Provedbom pet gore navedenih koraka i razvijanjem odgovarajuće poslovne strategije - u kombinaciji s pravim praksama podataka i alatom za analizu - možete otkriti kako rješavati trenutne poslovne izazove i prikazati budućnost vaše tvrtke.

## 2.3. OSTALI TIPOVI PODATAKA PO STRUKTURI

### 2.2.1. STRUKTURIRANI PODATCI

Strukturirani podatci su bazirani na shemi što znači da je struktura unaprijed isplanirana i strogo poznata. Nakon što se definira shema, podatci se mogu unositi na temelju kriterija zadanih u njoj. Najjednostavnije rečeno, strukturirani podatci su svi oni koji su organizirani u oblikovani repozitorij, obično je to baza podataka, tako da se njeni elementi mogu adresirati radi učinkovitije obrade i analize. Struktura podataka je vrsta spremišta koja organizira informacije u tu svrhu. U bazi podataka, na primjer, svako polje je diskretno i njezine se informacije mogu dohvatiti odvojeno ili zajedno s podatcima iz drugih polja, u različitim kombinacijama. Snaga takve strukture je njena sposobnost da obuhvati sve podatke i pruži korisne informacije. Strukturirani podatci se uglavnom odnose na informacije sa visokim stupnjem organizacije, tako da je uključivanje u relacijsku bazu neprekinuto i lako se može pretražiti jednostavnim, izravnim algoritmima određene tražilice ili pomoću drugih pretraživačkih operacija. Kada su informacije dobro strukturirane i predvidljive, tražilice mogu lakše organizirati i prikazati podatke na kreativan način. Kod označavanja strukturiranih podataka obično se koristi shema.org vokabular.[[3]](#footnote-3)

Ovakvi podatci su najzastupljeniji u razvoju aplikacija i pružaju najjednostavniji način obrade podataka. Bitno je naglasiti da strukturirani podatci čine 5-10% ukupnog postotka svih informatičkih podataka.

### 2.2.2. POLUSTRUKTURIRANI PODATCI

Kod polustrukturiranih podataka(eng. semi-structured data) ne postoji rigorozno definirana shema nego je način označavanja opcionalan. U nekim oblicima polustrukturiranih podataka shema uopće ne postoji, dok kod nekih postavlja samo labava ograničenja nad podatke. Podatke koji imaju labava ograničenja karakterizira nepravilna i implicitna struktura te fleksibilnost.

Primjer polustrukturiranih podataka su XML(eng. eXtensible Markup Language), JSON( eng. Javascript Object Notation), NoSQL(eng. Not Only SQL) baze podataka. Slično kao i kod strukturiranih, na polustrukturirane otpada samo 5 do 10% informatičkih podataka. Prednost polustrukturiranih podataka je u tome što uključuju mogućnost prikaza podataka koje nije lako ograničiti shemom te sama felksibilnost u pogledu prijenosa podataka, sposobnost prikaza strukturiranih podataka kao i sposobnost mijenjanja strukture tijekom određenog vremena.

Polustrukturirani podatci uglavnom imaju sljedeće karakteristike:

* Podatci su modelirani u obliku stabla ili grafova
* Podatci mogu postojati sa ili bez sheme

# 3. UČITAVANJE PODATAKA

# 

Učitavanje podataka(eng. Data Scraping) je tehnika koja se koristi za ekstrakciju velikih količina podataka s lokalnog izvora, baze podataka ili Interneta. Obično se prijenos podataka između programa postiže pomoću podatkovnih struktura pogodnih za automatsku obradu podataka. Učitavanje je postupak dohvaćanja podataka, obično nestrukturiranih ili strukturiranih, iz izvora podataka kako bi se oni iskoristili za daljnju obradu ili pohranu. Uvoz u sustav međustrukturiranja obično prati pretvorba podataka i dodavanje metapodataka. Ekperimentalni podatci se prvo uvoze u računalo iz primarnih izvora, kao što su uređaji za mjerenje ili snimanje. Današnji elektronički uređaji obično predstavljaju električni priključak(npr. USB) pomoću kojeg se „sirovi podatci“ mogu prenijeti u osobno računalo. Izdvajanje podataka iz nestrukturiranih izvora

## 3.1. TEHNIKE UČITAVANJA PODATAKA

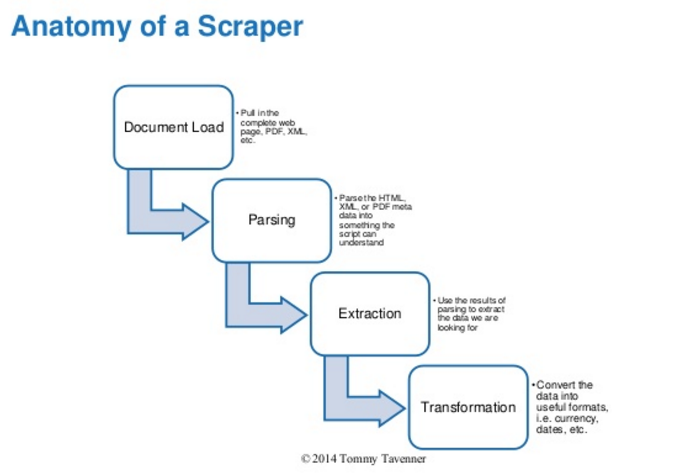
### 3.1.1. SCREEN SCRAPING

Sredinom 80-ih godina, pružatelji financijskih podataka kao što su Quotron, Telerate i Reuters prikazali su podatke u formatu 24 × 80 prilagođen čovjeku. Korisnici tih podataka, posebice banke za investicije, napisali su aplikacije za snimanje i pretvaranje takvih podataka u numeričke kako bi se iskoristili za izračune odluka o trgovanju. Screen scraping obično je povezan sa programskim skupljanjem vizualnih podataka iz nekakvog izvora, umjesto raščlanjivanja podataka kao kod web scrapinga. Izvorno, screen scraping odnosi se na praksu čitanja tekstualnih podataka sa zaslona računala. To je uglavnom učinjeno čitanjem memorije terminala kroz pomoćni priključak ili spajanjem izlaznog izlaza terminala jednog računalnog sustava s ulaznim priključkom na drugom. Izraz "struganje"(eng. scraping) zaslona također se obično koristi za dvosmjernu razmjenu podataka. To bi mogli biti jednostavni slučajevi u kojima se kontrolni program kretao kroz korisničko sučelje ili složenije scenarije gdje kontrolni program unosi podatke u sučelje koje je namijenjeno ljudima.   
Kao konkretan primjer klasičnog screen scraping alata, ​​može se navesti računalo s korisničkim sučeljima iz razdoblja 80-tih.

Često korišteno samo kroz tekstualne terminale koji nisu bili mnogo više od virtualnih printera. Želja za sučeljem takvog sustava u modernijim sustavima je česta. Kvalitetno rješenje često zahtijeva više stvari koje nisu na raspolaganju, poput izvornog koda, dokumentacije sustava, API-ja ili programera s iskustvom u računalnom sustavu od 50 godina. U takvim slučajevima, jedino moguće rješenje može biti napisati screen scraper. Screen scraper može se povezati s sustavom putem Telnet-a, oponašati tipke potrebne za navigaciju starim korisničkim sučeljem, obraditi rezultirajući izlazni prikaz, izvući željene podatke i proslijediti ga modernom sustavu. Moderne tehnike screen scraping-a obuhvaćaju snimanje bitmap podataka s ekrana i prikazivanje kroz OCR motor ili usklađivanje bitmap podataka s nekakvim očekivanim rezultatima. Putem aplikacija sa grafičkim sučeljem i s kontrolnim upitima, niz ekrana se automatski bilježi i pretvara u bazu podataka.

### 3.1.2. WEB SCRAPING

Web stranice izrađuju se pomoću tekstualnih jezika (HTML i XHTML), a često sadrže bogatstvo korisnih podataka u obliku teksta. Međutim, većina web stranica namijenjena je ljudskim krajnjim korisnicima, dok strojevi teško iskorištavaju takve podatke za automatsku upotrebu. Zbog toga su izrađeni alati koji mogu učitati sadržaj iz web stranica i pretvoriti ih u pogodan oblik. Web scraper je API ili alat za učitavanje podataka s web stranica. Web scraping koristi tehniku učitvanja velikih količina podataka s internetskih stranica nakon kojih se podatci sažimaju i spremaju u lokalnu datoteku na računalu ili u bazu podataka s tablicama. Podatci koje sadrže razne internetske stranice mogu se prikazivati samo putem internetskog preglednika. Internetski preglednici ne nude funkcionalnosti kao što je spremanje kopije podataka za kasniju upotrebu. Jedina mogućnost je kopirati i zalijepiti podatke. Stoga, web scraping omogućuje automatsku realizaciju procesa spremanja kopije podataka u vrlo kratkom vremenu. Web scraping je vrlo sličan indeksiranju, procesu kojim pretraživači indeksiraju internetski sadržaj. Razlika je u tome što internetski pretraživači imaju pravila postavljena u *robot.txt* datoteci i moraju ih poštovati, dok web scraperi ne.



**Slika 4.** Anatomija Web Scrapera

### 3.1.3. REPORT MINING

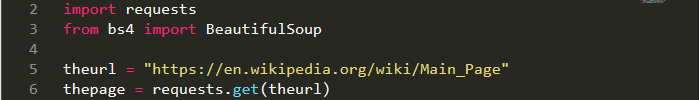
Report mining je prikupljanje podataka iz izvješća koja su čitljiva i prilagođena tako da ih ljudi mogu čitati. Prikupljanje podataka zahtjeva vezu sa sustavom koji sadrži izvorni program, korištenje odgovarajućih standarda ili programskih sučelja i složene upite. Korištenjem standardnih opcija koje su dostupne pri izradi izvješća i usmjeravanjem na određenu datoteku umjesto na pisač, može se generirati statično izvješće prikladno za analizu bez pristupa internetskoj vezi. Ovaj pristup može izbjeći prekomjerenu upotrebu glavnih dijelova računala tijekom radnog vremena, može smanjiti cijenu određenih sustava za sakupljanje podataka te ponuditi vrlo brzo i prilagođeno riješenje za izradu izvješća. Razlika između report mining proces i drugih tehnika prikupljanja podataka je ta da report mining uključuje izdvajanje podataka iz datoteka u ljudima čitljivom obliku, kao što su HTML, PDF ili tekst. Ovaj proces može pružiti brz i jednostavan način za prikupljanje podataka bez potrebe za programiranjem i izradom vlastitih alata.

## 3.2. ALATI ZA UČITAVANJE NESTRUKTURIRANIH PODATAKA

### 3.2.1. BIBLIOTEKE

#### 3.2.1.1. REQUESTS

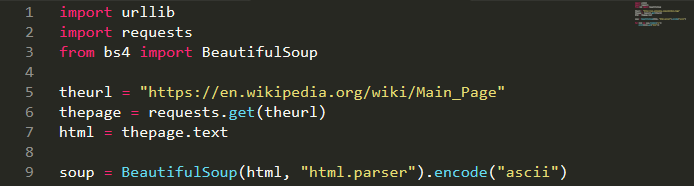
Requests je Apache2 licencirana HTTP biblioteka, napisana u Pythonu. Namijenjena je ljudima da ih koriste za interakciju s Python programskim jezikom. Requests omogućuje slanje HTTP zahtjeva koristeći Python programski jezik. Kod korištenja ove biblioteke nema potrebe ručnog dodavanja nizova upita URL-ovima ili oblikovanja kodiranih POST podataka.



**Slika 4.** Primjer korištenja Requests biblioteke

#### 3.2.1.2. **BEAUTIFUL** SOUP

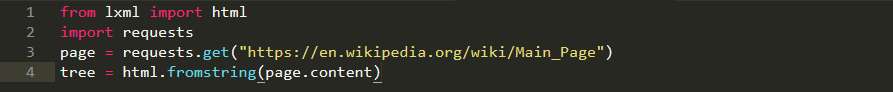
Beutiful Soup je Python biblioteka za izvlačenje podataka iz HTML i XML datoteka. Ova biblioteka pruža nekoliko jednostavnih metoda i idioma za navigaciju, pretraživanje i izmjenu dobivenih podataka. Predstavlja jednostavan alat za analizu dokumenta i vađenje podataka koji su nam potrebni. Beautfiul Soup automatski pretvara dolazne dokumente u Unicode i odlazne dokumente u UTF-8.



**Slika 5.** Primjer korištenja Beautiful Soup biblioteke

#### 3.2.1.3. LXML

LXML je visokokvalitetna HTML i XML biblioteka napisana u Pythonu. Jedinstvena je po tome što kombinira brzinu i cjelovitost XML značajki sa jednostavnim izvornim Python API-jem koji je uglavnom kompatibilan, ali i superioran od drugih API-ja. LXML metoda učitava cijeli dokument i gradi stablo u memoriji. Stablo koje je učitano sadržava informacije o kontekstu čvora, uključujući i reference na ono što je prije čvora.

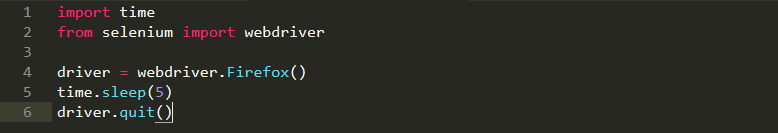


**Slika 6.** Primjer korištenja LXML bibilioteke

#### 

#### 3.2.1.4. SELENIUM

Selenium je paket biblioteka koji služi za automatizaciju internetskog preglednika, također je poznat i kao web driver.

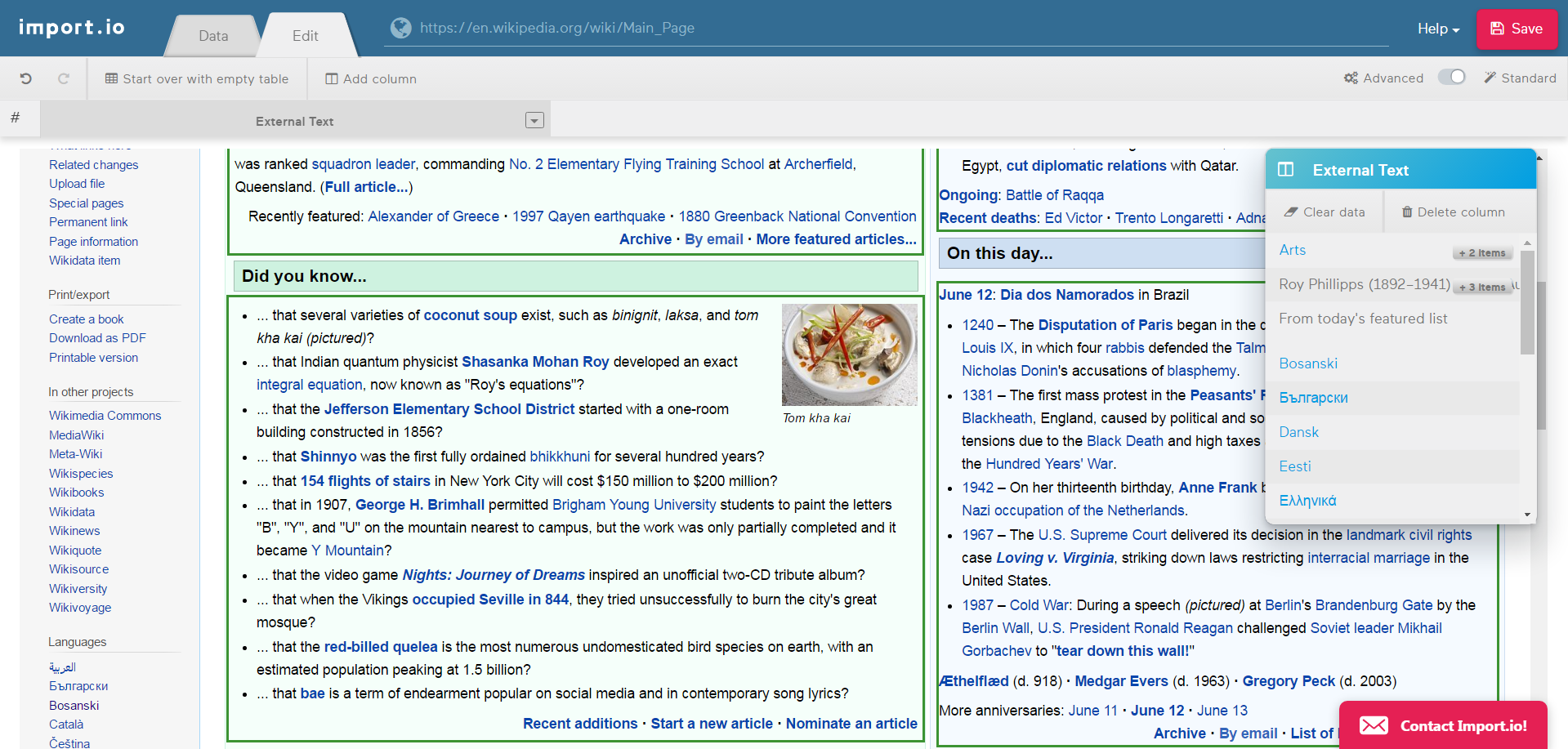


**Slika 7.** Primjer korištenja Selenium Web driver paketa

### 3.2.2. ALATI

#### 3.2.2.1. IMPORT.IO

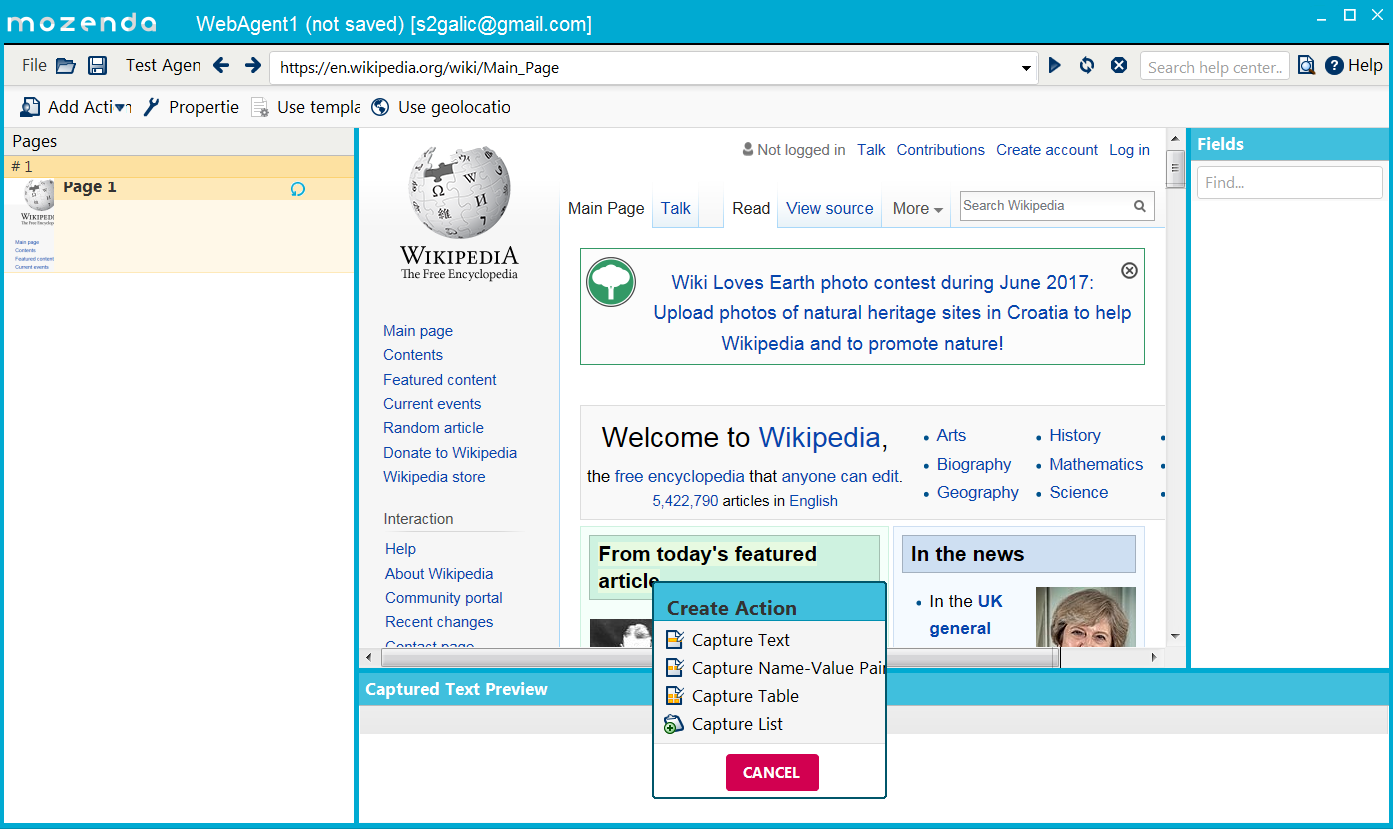
Import.io je web-based platforma za izdvajanje podataka s web stranica. Alat koji omogućuje pretvaranje nestrukturiranih web podataka u strukturirani format za upotrebu u strojnom učenju, umjetnoj inteligenciji, praćenju maloprodajnih cijena, lokacijskim lokatorima, kao i akademskim i drugim istraživanjima. Korisnici unesu URL i aplikacija pokuša automatski izvući podatke za koje misli da su korisniku potrebni, ako automatsko izdvajanje ne daje točno ono što vam je potrebno, sučelje omogućuje da se filtrira pretraga i količina podataka svede na konkretne. Podatci koje prikupljaju korisnici pohranjuju se na poslužiteljima oblaka Import.io i mogu se preuzeti kao CSV, Excel, Google tablice, JSON ili pristupiti putem API-ja. Korisnici mogu jednostavno integrirati internetske podatke uživo u vlastite aplikacije ili softver za analizu i vizualizaciju. Više izvora podataka može se izdvojiti istodobno.



**Slika 8. Primjer korištenja Import.io programskog alata**

#### 3.2.2.2. Mozenda

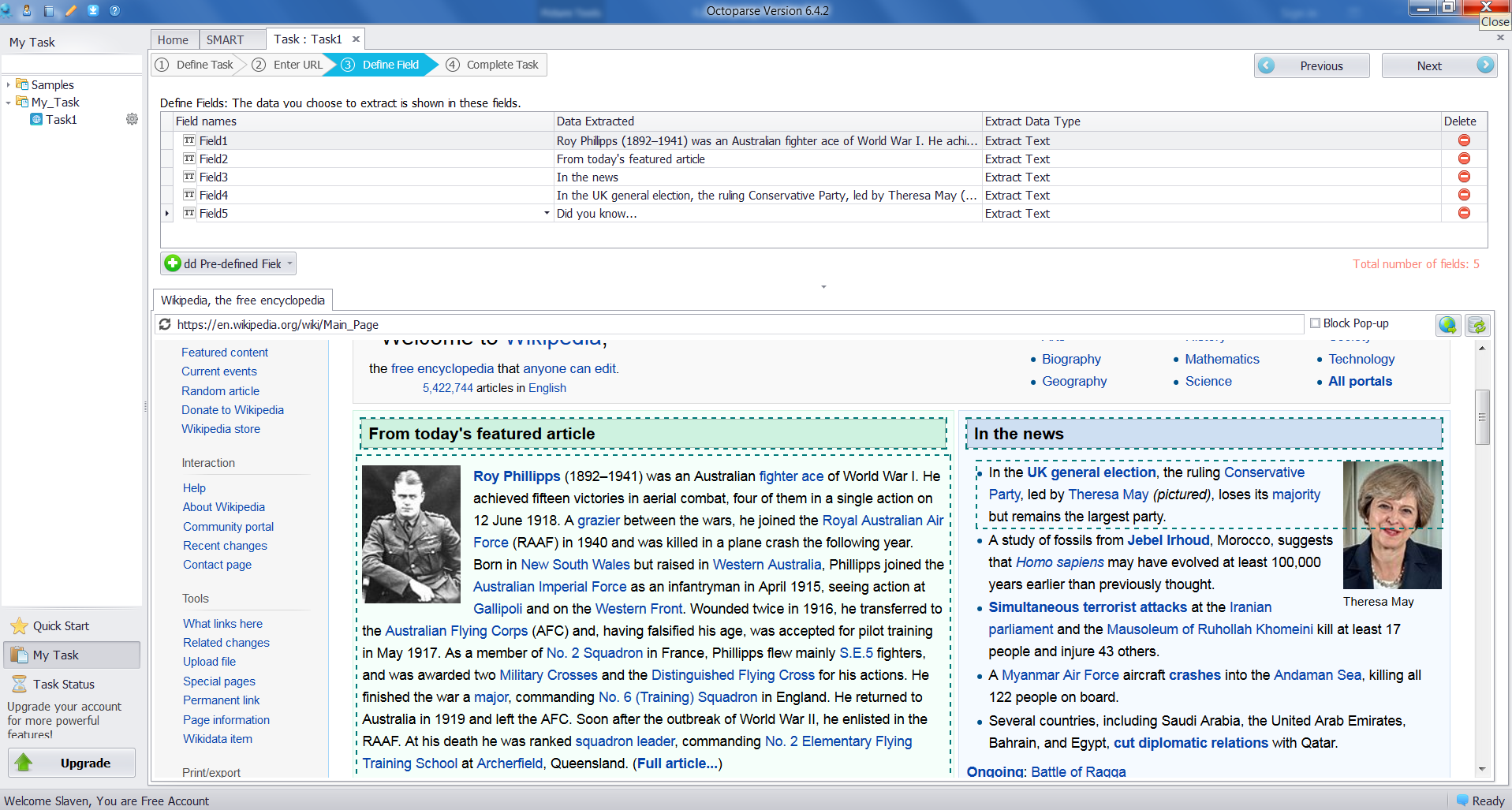
Mozenda je aplikacijski okvir za indeksiranje web stranica i vađenje strukturiranih podataka koji se mogu koristiti za širok raspon korisnih aplikacija, poput rudarenja podataka, obrade podataka ili povijesnog arhiviranja. Iako je Mozenda alat izvorno dizajniran za skidanje podataka sa web stranica, može se koristiti i za ekstrakciju podataka pomoću API-ja (kao što su Amazon Associates Web Services) ili kao web pretraživač opće namjene. Mozenda koristi opciju Agent Buildera. Agent Builder podržava stvaranje agenata koji prikupljaju određene podatke s web stranica. Agent ima opcije koje se mogu mjenjati kako bi se postigla što konkretnija pretraga i dobio kvalitetniji rezultat.



**Slika 9. Primjer korištenja Mozenda pogramskog alata**

### 3.2.2.3. OCTOPARSE

Octoparse je besplatan, jednostavan za korištenje, ali moćan alat za učitavanje podataka sa klijentske strane koji može biti od velike pomoći u pojednostavljenju procesa učitavanja internetskih stranica, povećavajući učinkovitost i optimiziranjem performansi.

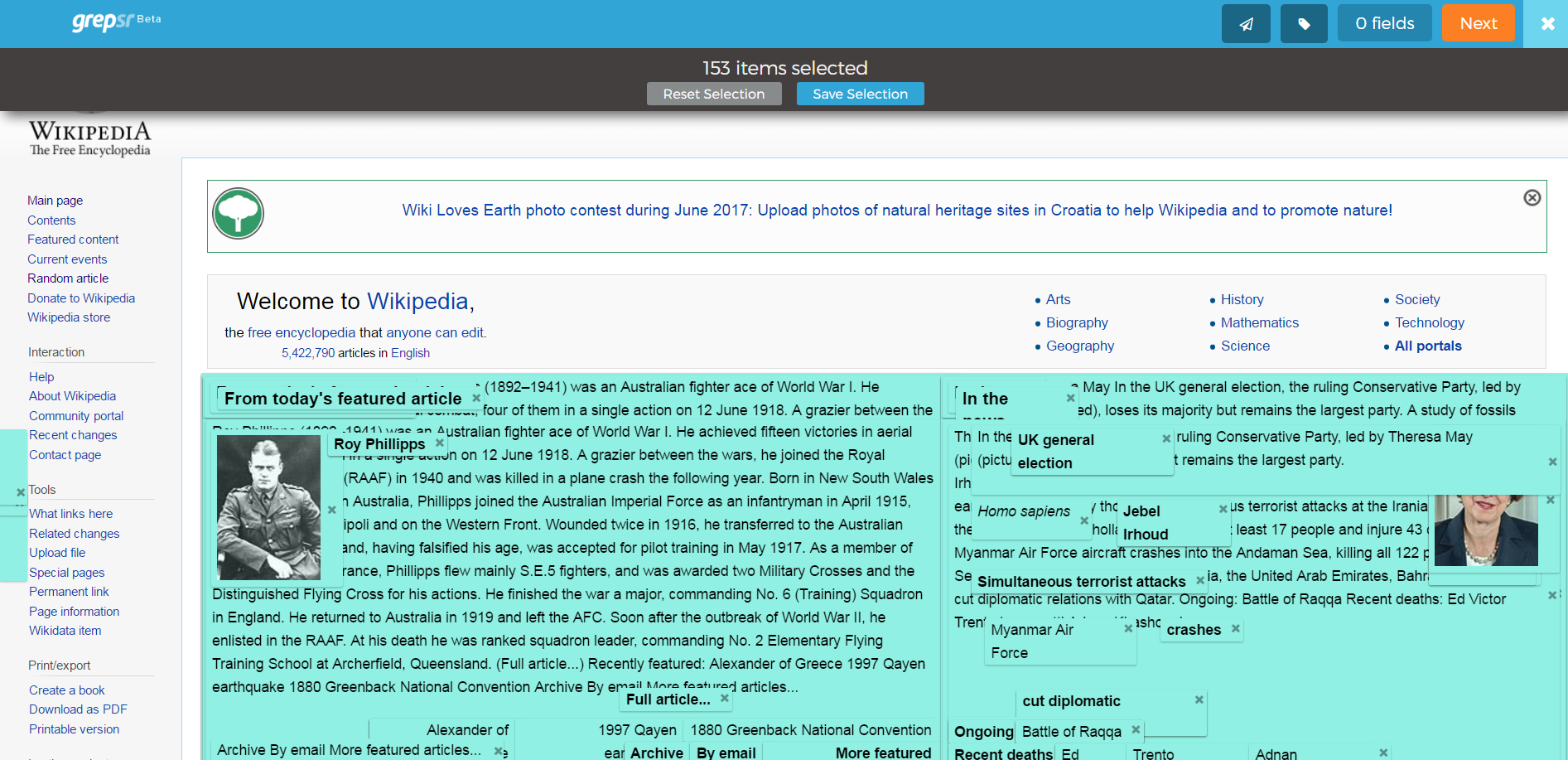


**Slika 10. Primjer korištenja Octoparse alata**

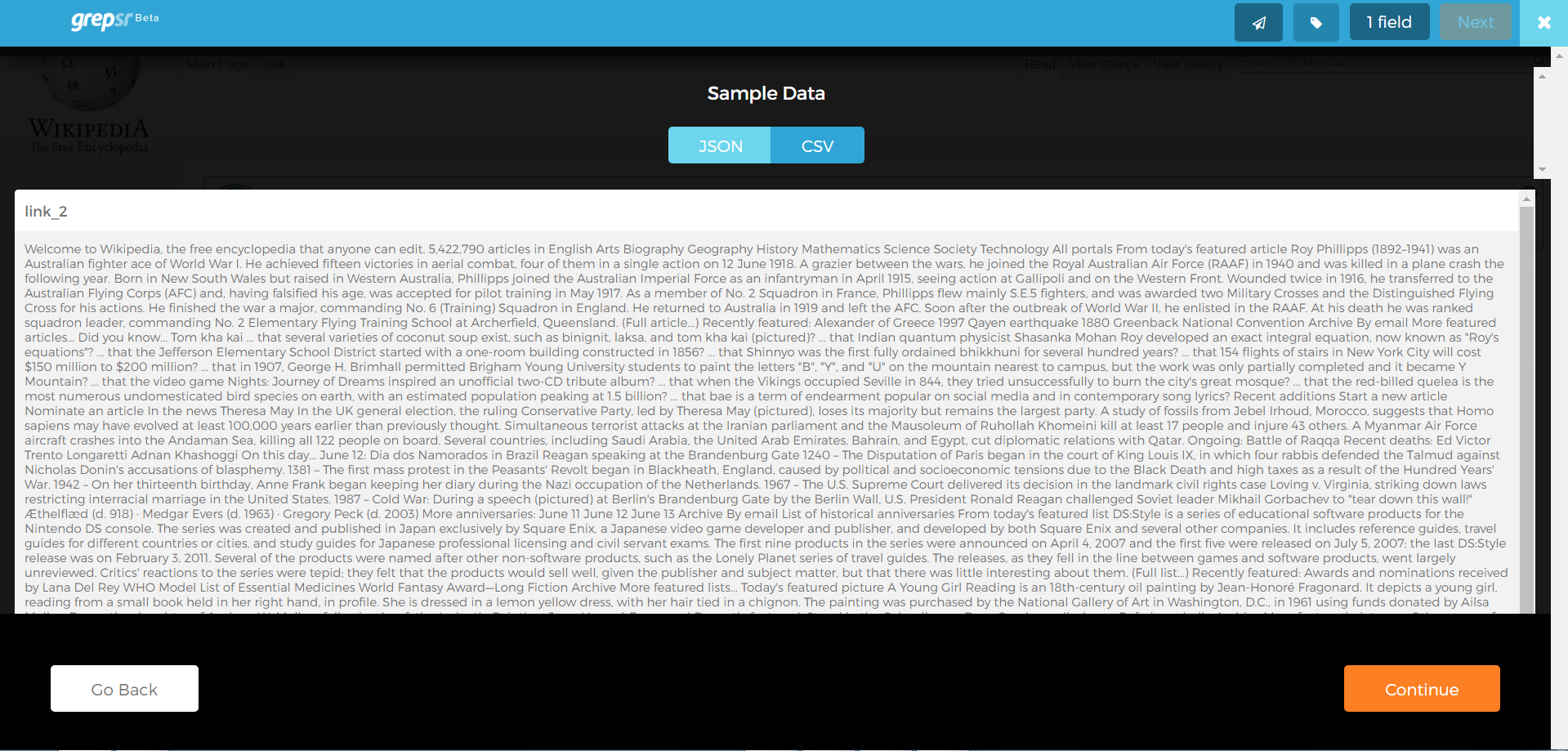
### 3.2.3. DODATCI ZA PREGLEDNIKE

#### 3.2.3.1. GREPSR

Grepsr je besplatni dodatak za Google Chrome preglednik koji vam omogućuje jednostavno izdvajanje podataka bilo koje web stranice pomoću intuitivnog point and click načina i pretvaranje tih podataka u proračunsku tablicu.



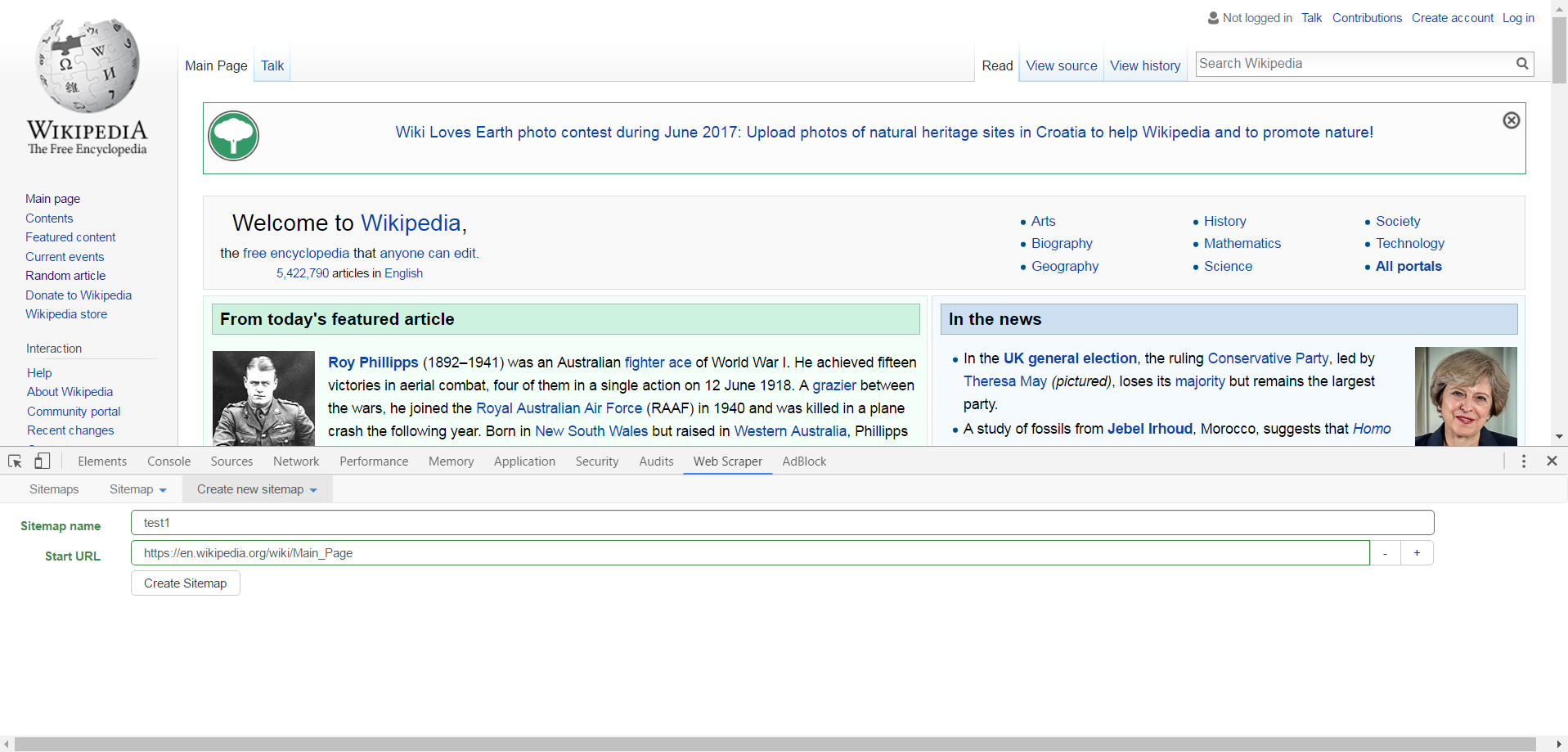
**Slika 11.** Primjer korištenja Grepsr dodatka



**Slika 12.** Spremanje podataka u polustrukturiranom obliku

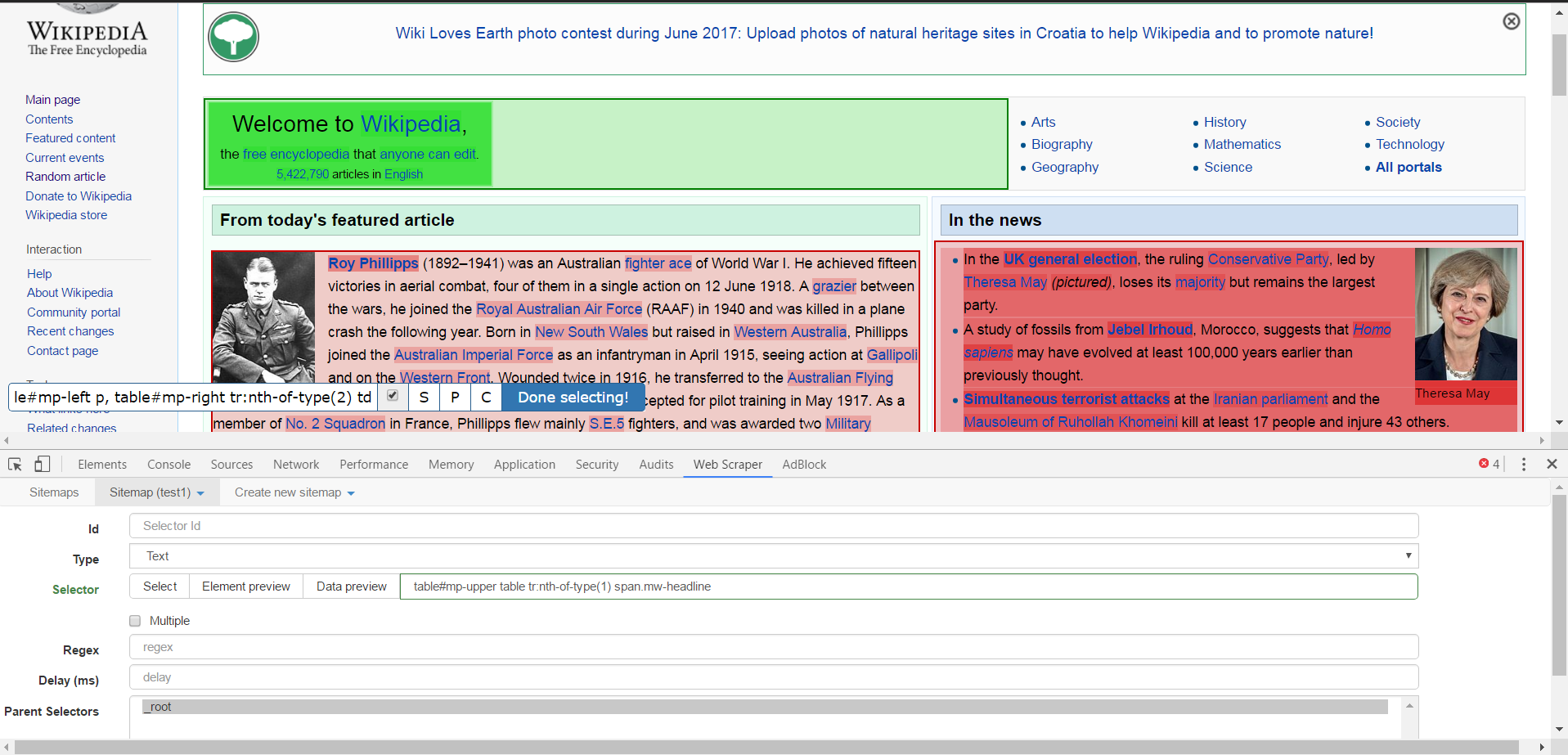
#### 3.2.3.2. WEB SCRAPER CHROME DODATAK

Web Scraper je proširenje za Google Chrome preglednik izgrađeno za ekstrakciju podataka s web stranica. Pomoću ovog proširenja možete izraditi plan (eng. sitemap) kako bi trebalo proći web stranicu i što bi trebalo izvući. Pomoću ovih planova Web Scraper prelazi kroz internetsku stranicu prema planu i izvlači sve podatke. Kasnije kopirani podaci mogu se izvesti u .CSV formatu.



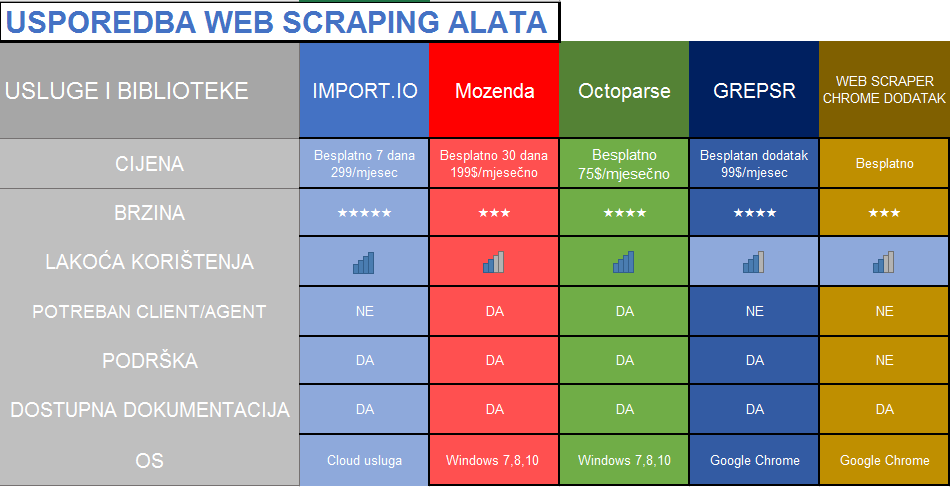
**Slika 13.** Primjer korištenja Web Scraper dodatka za Google Chrome preglednik

Prvo što se mora napraviti je izraditi plan(eng. Sitemap), gdje pri kreiranju stavljamo naziv i početnu poveznicu. Nakon toga može definirati detalje koji će se koristiti u planu, odnosno postaviti *selector* po kojima će alat pretražiti internetsku stranicu. Dodatak nudi opciju point and click, tako da se određeni dijelovi samo odaberu klikom miša.



**Slika 14.** Odabir *selectora* u Web Scraper dodatku

### 3.2.4. USPOREDBA USLUGA I ALATA



**Slika 15.** Usporedba alata za učitavanje podataka

# 4. PRAKTIČNI DIO

Praktični dio rada, alat za učitavanje podataka, implementiran je kao skripta koristeći programski jezik Python. Odabrane su dvije biblioteke navedene u ovom radu, Beautiful Soup i urllib.

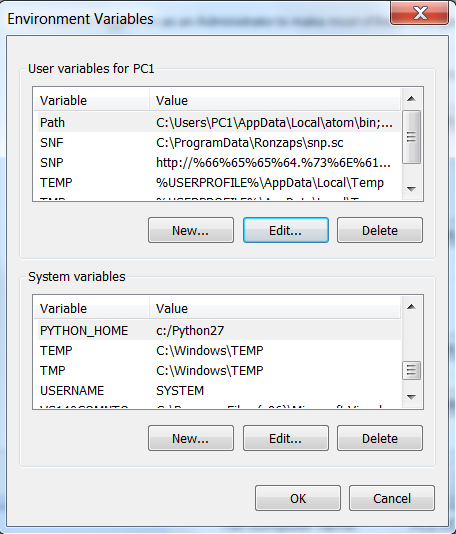
### 4.1. **RAZVOJNO OKRUŽENJE**

### Prije početka izrade svakog programskog alata potrebno je pripremiti razvojno okruženje. Ovaj projekt je napravljen u Windows opercijskom sustavu, a za njegovu izradu bilo je potrebno instalirati:

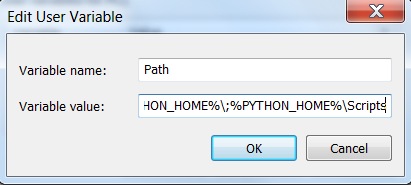
1. Pyhton 2.7.13
2. pip paketni upravljač
3. Python biblioteku Beautiful Soup
4. Python biblioteku urllib

**Python instalacija**

Sa službene stranice može se preuzeti instalacijski paket (.exe) za Windows operacijski sustav. Nakon što je instalacija završena, potrebno je omogućiti pokretanje Python-a iz komandne linije u Windows operativnom sustavu. Odabirom na početnu tikpu „Start“ u Windows operativnom sustavu, možemo pomoću trake za pretraživanje pronaći opciju „Environment Variables“ gdje se unosi putanja(eng. Path) do lokalnog Python direktorija.

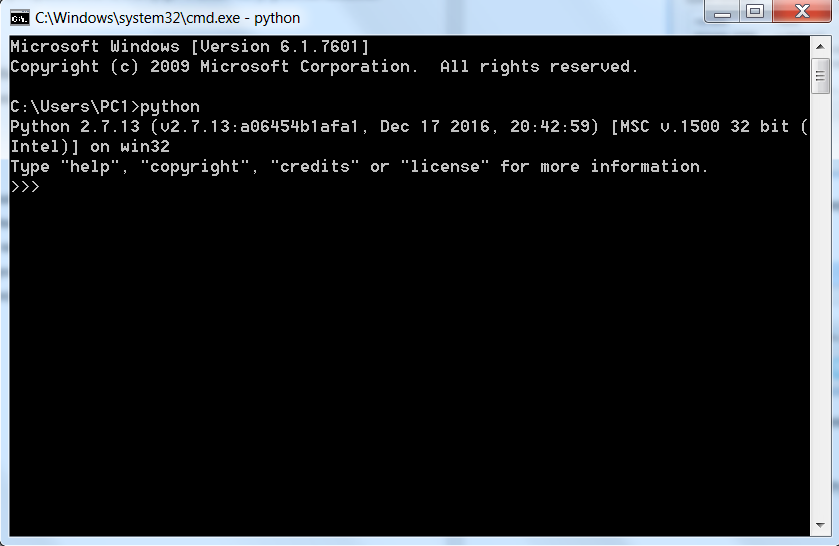


**Slika 16.** Postavljanje putanje do Python direktorija



**Slika 17.** Odabrani direktorij u kojem se nalaze skripte za pokretanje

Nakon što smo odabrali i unijeli ispravnu putanju, Python se može pokrenuti kroz komandnu liniju u Windows operativnom sustavu.



**Slika 18. Pokretanje Python programskog sučelja iz komandne linije**

Nakon pokretanja Python programskog sučelja možemo odmah započeti sa izvođenjem jednostavnih programa unosom željenih varijabli i operacija. Za potrebe korištenja prije spomenutih biblioteka potrebna je instalacija *pip* paketnog upravljača. Prvo moramo skinuti get-pyp.py datoteku sa Python službene stranice te ju pokrenuti.



**Slika 19.** Instalacija pip paketnog upravljača

Instalacijom *pip* paketnog upravljača dobije se mogućnost instalacije besplatnih biblioteka potrebnih za izradu vlastitog alata za učitavanje podataka. Za početak je potrebno instalirati biblioteku BeautifulSoup. Jednostavnim pokretanjem naredbe u programskom sučelju instaliramo besplatnu biblioteku.



**Slika 20.** Instalacija besplatne biblioteke BeautifulSoup

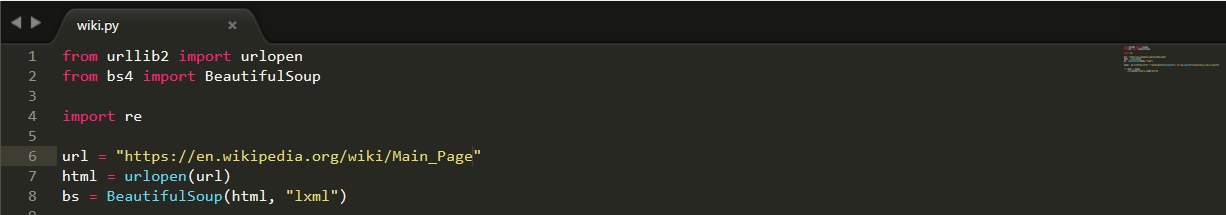
Za ispravan rad alata za učitavanje podatak potrebna nam je i biblioteka Requests koja nam omogućuje slanje HTTP zahtjeva. Nju također instaliramo putem programskog sučelja pokretanjem naredbe.



**Slika 21.** Instalacija besplatne biblioteke Requests

### 4.2. IZRADA ALATA

Nakon što smo instalirali sve potrebne biblioteke možemo započeti sa izradom alata za učitavanje podataka. Pomoću biblioteke urllib odabiremo koju stranicu želimo učitati tako što definiramo putanju na tu stranicu. Također učitavamo i biblioteku BeautifulSoupkoja će nam poslužiti kako bi se učitani podatci prikazali u normalnom obliku.



**Slika 22.** Pozivanje biblioteka i učitavanje poveznice s koje želimo učitati podatke

Kako bi dobili podatke u ispravnom obliku potrebno je pozvati i biblioteku *re.* Biblioteka *re* označava regularni izraz tj. specificira niz znakova koji se moraju podudarati. Ona omogućuje provjeru je li određeni niz znakova odgovarajući datom regularnom izrazu.

### 4.3. MOGUĆE NADOGRADNJE SKRIPTE

# 5. ZAKLJUČAK

Iako se alati za učitavanje podataka mogu nositi s jednostavnim ili umjerenim zahtjevima za vađenje podataka, oni nisu preporučeno rješenje ako ste poslovni korisnik koji pokušava steći podatke za istraživanje tržišta ili dobivanje prednosti nad konkurencijom. Kada je zahtjev velik i/ili kompliciran, alati za učitavanje podataka ne ispunjavaju očekivanja. Pomoću gotovih alata mogućnosti prilagodbe su ograničene i automatizacija je gotovo nepostojeća. Alati također dolaze s nedostatkom održavanja, što može biti zastrašujući zadatak. Samostalna izrada je bolja opcija i može biti pravi izbor ako su zahtjevi za podatke ograničeni, a web lokacije koje korisnik želi učitati nisu komplicirane. Ako učitavanje podataka zahtijeva prilagođeni skup postavki, samostalno izrađeni alat to ne može učitati. Korisnik podatke koji sadrže komercijalne informacije mora prvo dobiti dozvolu od vlasnika podataka ukoliko ne želi kršiti autorska prava.

# 6. PRILOZI

# 7. LITERATURA

# Popis slika i tablica

1. http://www.enciklopedija.hr/natuknica.aspx?id=48887 [↑](#footnote-ref-1)
2. http://www.enciklopedija.hr/natuknica.aspx?id=27405 [↑](#footnote-ref-2)
3. http://schema.org/ [↑](#footnote-ref-3)