

Umsetzung und Erweiterung **Web Scraper** **Handbuch**

Monica De Blasiis

April 2023

Inhaltsverzeichnis

[Inhaltsverzeichnis](#)

[1. Einleitung](#)

[2. Hinzufügen einer neuen Sprache](#)

[3. Navigieren auf einer Website mit Selenium](#)

[4. Verwendung der CSV-Datei in WordPress](#)

1. Einleitung

Willkommen Leser! Dies ist ein Handbuch für mein Web-Scraper-Projekt, an dem ich im Zeitraum von Ende März bis Anfang April 2023 gearbeitet habe.

Für den Kontext wäre es besser, zuerst meine Dokumentation durchzulesen, da sie einen guten Einblick in die Funktionsweise der von mir entwickelten Anwendung gibt. Alles, was Sie hier finden, ist eine Erklärung, wie Sie die Funktionalität der Anwendung erweitern können und wie Sie die gesammelten Daten auf einer Wordpress-Seite anzeigen können.

2. Hinzufügen einer neuen Sprache

Beginnen wir mit der einfachsten Erweiterung: dem Hinzufügen einer neuen Sprache. Als Erstes müssen Sie nur eines der bereits vorhandenen Sprachpakete duplizieren und es umbenennen, damit es dem Format der anderen entspricht. Für dieses Beispiel nehmen wir an, dass wir Italienisch, die fehlende Schweizer Landessprache, als neue Sprache hinzufügen wollen. Die Datei sollte dann in "lp_it" umbenannt werden.

Sie werden feststellen, dass das Sprachpaket nichts weiter als eine Liste von String-Variablen ist. Der nächste Schritt besteht einfach darin, den zugewiesenen Wert jeder Variablen zu ändern, indem wir sie in die Sprache übersetzen, die wir hinzufügen möchten.

```
# Pacchetto lingua italiana

selection = "Lingua selezionata: Italiano"

selector = "Scegliere una lingua"
```

```
location_question = "Di quale paese/citt  volete una lista di hotel?"

error_input_empty = "Dovete inserire qualcosa prima di poter continuare!"

error_input_contains_numbers = "I numeri non sono ammessi come input!"

error_input_contains_special_characters = "I caratteri speciali non sono ammessi come input!"

next = "Avanti"

back = "Indietro"

url_booking =
"https://www.booking.com/index.it.html?gclid=CjwKCAjw586hBhBrEiwAQYEnHffqs
FY8_mbTNkFHplsIeJ6HpldxUN1UowOjzxp6Ha1F1f1j9hwKgBoCXdgQAvD_BwE%3Bws%3D%3Ba
id%3D376363%3Blabel%3Dbooking-name-L*Xf2U1sq4*GEkIwcLOALQS540988882221%3Ap
l%3Aata%3Ap1%3Ap22%2C563%2C000%3Aac%3Aap%3Aneg%3Afi%3Atikwd-65526620%3Alp10
02859%3Ali%3Adec%3Adm%3Appccp%3DUMFuZG9tSVYkc2R1Iyh9YfqndqqG8nt1XFzPnqOODw
s&lang=it&soz=1&lang_changed=1&selected_currency=EUR"

start_scraping = "Avvia lo Scraper"

scraping_in_process = "Lo Scraper sta attualmente cercando informazioni.
Abbiate un po' di pazienza."

order_question = "Volete riordinare le informazioni raccolte in maniera
specifica?"

sort_alphabet = "Riordina in ordine alfabetico"

sort_price = "Riordina per prezzo (ascendente)"

no_sort = "Non riordinare"

storage_question = "Come volete salvare i dati raccolti?"

generate_file = "Generare un CSV"

generate_script = "Generare un SQL"

generate_all = "Generare entrambi i file"
```

```

db_upload = "Caricare in una banca dati"

db_upload_question = "Inserire le credenziali corrette dentro ogni campo di input."

db_host_input_label = "Nome dell'host"

db_user_input_label = "Nome dell'utente"

db_password_input_label = "Password"

db_name_input_label = "Nome della banca dati"

send_data = "Inviare i dati"

process_done = "Il processo di scraping è giunto al termine. Se volete, potete far ripartire l'applicazione premendo il sottostante bottone."

restart = "Ricomincia"

```

Denken Sie daran, die Datei nach der Änderung zu speichern. Der nächste Schritt besteht darin, das Sprachpaket als Option in das Optionsmenü der Datei scraper_gui.py einzufügen. Dazu ist es zunächst erforderlich, das Sprachpaket selbst in die Datei zu importieren.

```
import language_packages.lp_it as lp_it
```

Im nächsten Schritt fügen Sie eine neue Optionsschaltfläche hinzu, scrollen ganz nach unten, wo alle Elemente der Startseite deklariert sind, und fügen Code im folgenden Format hinzu.

```

R4 = Radiobutton(root, text="Italiano", value=4,

                  command= lambda: select_language("IT"))

R4.pack()

```

Jetzt muss nur noch eine neue elif-Regel in die select_language(lp)-Funktion eingefügt werden. Gehen Sie zu der Funktion und scrollen Sie nach unten, bis Sie sehen, wo sich alle anderen elif-Regeln befinden. Fügen Sie direkt nach der letzten Regel einen Code in diesem Format ein:

```

elif lp == "IT":

    language_pack = lp_it

```

So einfach ist das. Wenn es um die GUI allein geht, sind wir hier fertig!

Je nachdem, welche Websites Sie scrapen wollen und wie, müssen Sie möglicherweise auch einige Ergänzungen in der Datei scraper.py vornehmen.

Im Fall von Booking.com beispielsweise habe ich den Scraper so entwickelt, dass er von einer Ergebnisseite zur nächsten navigieren kann, indem er auf eine Schaltfläche mit der Aufschrift "Nächste Seite" in der Sprache der Wahl des Benutzers verweist.

Daher sind zwei weitere Schritte erforderlich, um die Einführung der italienischen Sprache abzuschließen.

Zunächst muss, genau wie in der vorherigen Datei, das Sprachpaket selbst importiert werden.

Dann müssen Sie ganz nach unten scrollen, bis Sie die letzten Codezeilen innerhalb der Funktion scrape_booking(url, input) sehen.

Es wird zwei Sätze von elif-Regeln geben. Fügen Sie unter der letzten elif-Regel des ersten Satzes Code in diesem Format ein:

```
elif url == lp_it.url_booking:

    street = "Non disponibile"
```

Fügen Sie unter der letzten elif-Regel des zweiten Satzes einen Code in diesem Format ein:

```
elif url == lp_it.url_booking:

    next_page = "pagina successiva"
```

Denken Sie daran, immer darauf zu achten, wie Sie Elemente referenzieren, mit denen die Anwendung interagieren soll, da einige Elemente ihren Text oder sogar ihre Klasse ändern können, wenn dieselbe Website in einer anderen Sprache geladen wird.

3. Navigieren auf einer Website mit Selenium

Unter der Voraussetzung, dass Sie meine Dokumentation gelesen haben, zumindest alle erforderlichen Selenium-Bibliotheken korrekt installiert haben und die Grundlagen zur Deklaration und Einrichtung eines WebDrivers kennen, ist es eigentlich sehr einfach, eine Python-Anwendung dazu zu bringen, auf einer Website zu navigieren (vorausgesetzt, die betreffende Website verfügt nicht über sehr aggressive Anti-Bot-Maßnahmen).

Normalerweise haben Buchungswbseiten, mit sehr wenigen Ausnahmen, sehr aggressive Anti-Bot-Maßnahmen, die Selenium daran hindern, richtig mit der Webseite zu interagieren. Der

Einfachheit halber werde ich anstelle einer Buchungswebsite zeigen, wie man mit einem einfachen Wiki interagiert und dort Daten sammelt.

Nehmen wir als Zielseite das Fandom-Wiki für das beliebte chinesische Open-World-Spiel Genshin Impact.

Der Einfachheit halber nehmen wir an, dass unser Scraper die folgenden Schritte ausführen soll:

- die Startseite des Wikis öffnen
- das Sucheingabefeld öffnen
- den Namen einer Figur eingeben
- die Eingabe abschicken
- zu der Seite gehen, zu der das erste passende Ergebnis führt
- zur Registerkarte "Synchronsprecher" navigieren
- die Namen der englischen und japanischen Synchronsprecher sammeln und in der Konsole ausgeben

Wie können wir also vorgehen? Als Erstes müssen wir der `get()`-Methode des `WebDrivers` die richtige URL geben. Dies kann entweder durch Kopieren und Einfügen der URL direkt in die Methode geschehen, wobei sie in Anführungszeichen gesetzt wird, um sie in einen String zu verwandeln, oder wir erstellen eine Variable, die einen String mit der URL als Wert enthält. Ich habe mich für Letzteres entschieden.

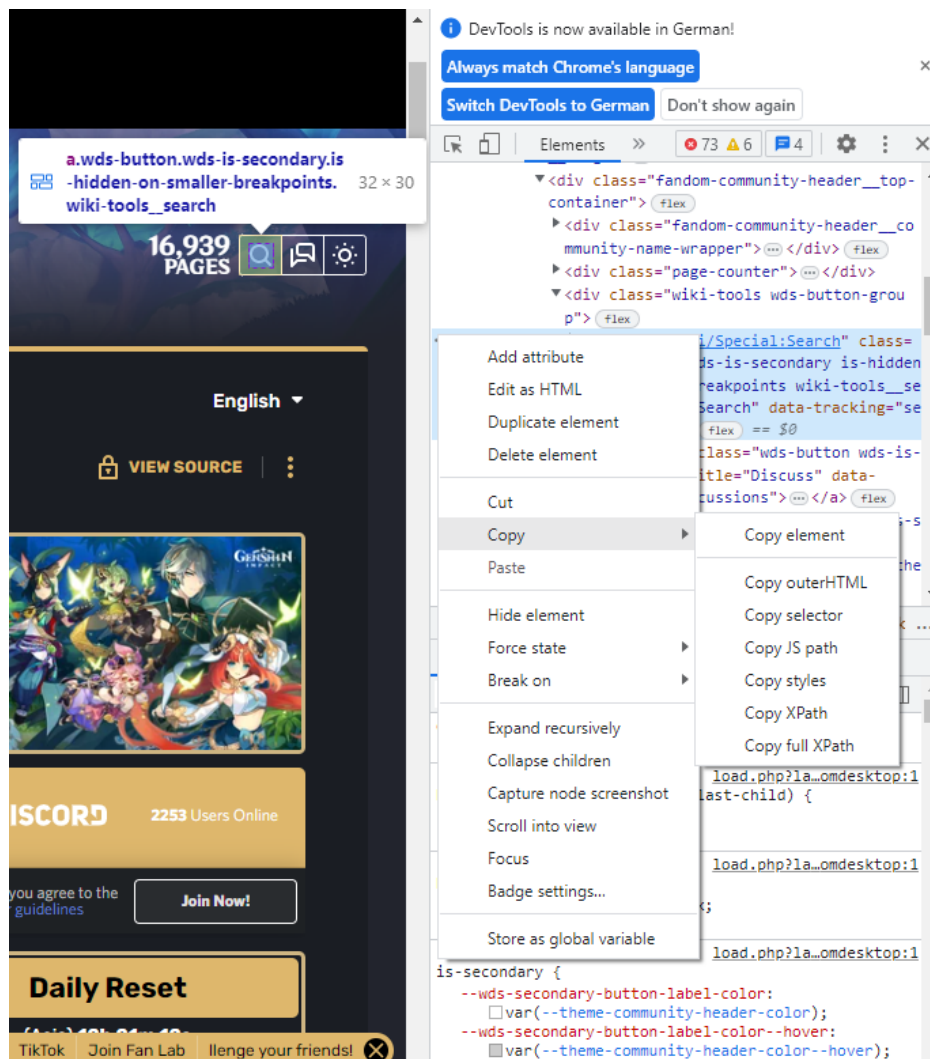
```
url = 'https://genshin-impact.fandom.com/wiki/Genshin_Impact_Wiki'
driver.get(url)
```

Es ist wichtig zu erwähnen, dass einige Seiten, wie z. B. Fandom-Wikis, ein Popup-Fenster für Cookies anzeigen, sobald Sie die Website öffnen, was den Scraper stören kann. Um dies zu verhindern, kann die folgende experimentelle Option zu den Optionen des `WebDrivers` hinzugefügt werden:

```
options.add_experimental_option("prefs",
{"profile.default_content_setting_values.cookies": 2})
```

Bevor wir nun mit der Codierung fortfahren können, müssen wir die eigentliche Seite besuchen. Warum? Um den CSS-Selektor zu erhalten, natürlich! Ich habe ihn bereits ausführlich in meiner Dokumentation erklärt, aber um kurz zu erklären, was das ist: Es ist ein Muster von Elementen, durch das Selenium das Element finden kann, mit dem es interagieren soll. Aber wie können wir diesen CSS-Selektor finden?

Das ist ganz einfach: Sie müssen nur mit der rechten Maustaste auf das betreffende Element klicken und dann auf "Prüfen". Sie müssen dies sogar zweimal tun. Wenn Sie es das erste Mal tun, werden die Google DevTools in Ihrem Browser auf der rechten Seite geöffnet. Beim zweiten Mal werden Sie tatsächlich dorthin geführt, wo sich das betreffende Element befindet. Dort können Sie entweder mit der rechten Maustaste auf das markierte Element klicken oder auf das Symbol mit den drei grauen Punkten links neben dem markierten Element. Es sollte dann ein Menü erscheinen, in dem eine der Optionen "Kopieren" lautet. Wenn Sie mit dem Mauszeiger über das Menü fahren, werden Sie feststellen, dass eine Reihe von Optionen angezeigt werden. Diejenige, die Sie für dieses Projekt anklicken möchten, ist "Selektor kopieren". Dadurch wird der CSS-Selektor in Ihre Zwischenablage kopiert.



Jetzt müssen wir nur noch dafür sorgen, dass die Anwendung mit dem Element interagiert. In diesem Fall wollen wir auf das Element klicken. Dazu nehmen wir die driver-Variable und verketteten die Methoden `find_element(by=, value=)` und `click()`. Als Wert für das `by`-Attribut können wir die Klasse `By` aufrufen und dann deren Variable `CSS_SELECTOR` aufrufen. Als Wert für das `value`-Attribut können wir einfach den CSS-Selektor einfügen, den wir zuvor kopiert haben, und ihn in Anführungszeichen einschließen, um ihn zu einem String zu machen. Der resultierende Code sollte am Ende etwa so aussehen.

```
driver.find_element(by=By.CSS_SELECTOR, value="body > div.main-container >
div.resizable-container > div.community-header-wrapper > header > div >
div.wiki-tools.wds-button-group >
a.wds-button.wds-is-secondary.is-hidden-on-smaller-breakpoints.wiki-tools_
_search").click()
```

Wir wissen jetzt, wie wir die App dazu bringen, auf etwas zu klicken, aber wie bringen wir sie dazu, etwas in das Eingabefeld einzugeben? Alles, was Sie tun müssen, ist, den obigen Code zu kopieren und ein paar kleine Änderungen vorzunehmen: Erstens müssen Sie den CSS-Selektor durch den des Eingabefeldes ersetzen, und zweitens werden wir statt der Methode `click()` die Methode `send_keys("")` verwenden. Der Parameter für `send_keys` kann entweder hart kodiert werden oder Sie können ihn flexibler gestalten, indem Sie eine String-Variable einfügen (was ich für mein Projekt getan habe). Um dieses Beispiel nicht zu komplex zu machen, werden wir einfach einen festen String als Parameter angeben. Ich beschloss, nach Informationen über die spielbare Figur namens "Albedo" zu suchen.

```
driver.find_element(by=By.CSS_SELECTOR, value="body > div.search-modal >
div > form > input").send_keys("Albedo")
```

Die nächsten Schritte fügen nichts Neues hinzu, wenn es um Wissen geht, da wir einfach den Code aus der ersten Elementinteraktion nehmen und einfach die CSS-Selektorzeichenfolge ändern können. Insbesondere lassen wir die Anwendung zuerst auf die Suchschaltfläche und dann auf das erste Ergebnis auf der Suchergebnisseite klicken.

```
driver.find_element(by=By.CSS_SELECTOR, value="body > div.search-modal >
div > form > button.wds-button.wds-is-square").click()

driver.find_element(by=By.CSS_SELECTOR, value="#mw-content-text > section
> div > div.unified-search__layout__main > ul > li:nth-child(1) > article
> h3 > a").click()
```

Nun sollte die Anwendung auf der Seite der betreffenden spielbaren Figur angekommen sein. Wir können den Schritt, den wir gerade getan haben, noch einmal wiederholen und die Anwendung dazu bringen, auf die Registerkarte Synchronsprecher im Infofeld der Figur zu klicken.

```
driver.find_element(by=By.CSS_SELECTOR, value="#mw-content-text >
div.mw-parser-output > aside > section:nth-child(5) >
div.wds-tabs__wrapper > ul > li:nth-child(3) > div").click()
```

Dort können wir endlich auf die gesuchten Informationen zugreifen. Wieder einmal werden wir dieselbe Codezeile verwenden, die wir bisher so oft kopiert und eingefügt haben, aber etwas anders. Zuerst müssen wir die click()-Funktion daraus entfernen, und dann müssen wir diese ganze Codezeile einer Variablen zuweisen.

```
english_va = driver.find_element(by=By.CSS_SELECTOR,
value="#mw-content-text > div.mw-parser-output > aside >
section:nth-child(5) > div.wds-tab__content.wds-is-current >
div:nth-child(1) > div > a")

japanese_va = driver.find_element(by=By.CSS_SELECTOR,
value="#mw-content-text > div.mw-parser-output > aside >
section:nth-child(5) > div.wds-tab__content.wds-is-current >
div:nth-child(3) > div > a")
```

Wir können dann die Texteigenschaft der Variablen drucken und ... voila! Aufgabe abgeschlossen! Die Daten wurden gesammelt!

```
print("English VA: " + english_va.text)

print("Japanese VA: " + japanese_va.text)
```

Output:

```
English VA: Khoi Dao
Japanese VA: Nojima Kenji (野島 健児)
```

4. Verwendung der CSV-Datei in WordPress

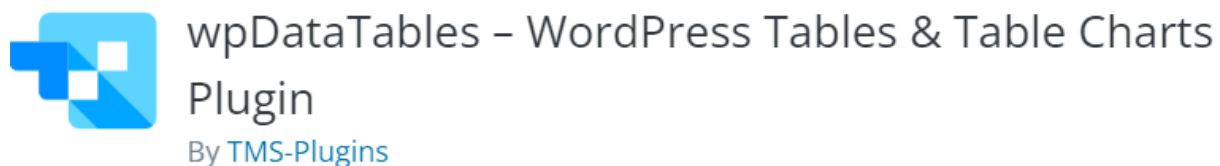
Ursprünglich hatte ich vor, den Web Scraper selbst auf WordPress nutzbar zu machen, indem ich die Anwendung vielleicht zum Beispiel in eine php-Seite des Admin-Dashboards einbette. Zum Zeitpunkt der Erstellung dieses Handbuchs habe ich es jedoch versäumt, rechtzeitig eine Lösung zu finden. Trotzdem wollte ich diesen Punkt nicht ganz unangetastet lassen, also habe ich eine alternative Lösung gefunden, die zwar nicht optimal ist, aber die von der App generierte CSV-Datei verwendet und ihre Daten mithilfe eines Drittanbieter-Plugins in einer WordPress-Seite anzeigt.

Das betreffende Plugin heißt wpDataTables. wpDataTables ist ein beliebtes WordPress-Tabellen-Plugin zum schnellen Erstellen von Tabellen und Tabellendiagrammen aus Excel, CSV, XML, JSON, PHP und anderen Datenquellen. Dieses WP-Tabellen-Plugin kann verwendet werden, um große Mengen komplizierter Daten auf prägnante, benutzerfreundliche Weise mithilfe von Tabellen oder Diagrammen darzustellen.

Die Hauptvorteile, die wpDataTables seinen Benutzern bietet, sind Zeitersparnis (da es die Anzeige von Daten auf der Website automatisiert und dies ermöglicht, ohne Zeit in die Codierung zu investieren) und keine Notwendigkeit, Codierung zu lernen (das Erstellen einer WordPress-Tabelle wie in wpDataTable würde eine Entwicklung ohne sie erfordern).¹

Also, wie gehen wir vor? Nun, zuerst muss die CSV-Datei mit allen vom Scraper gesammelten Daten generiert und einsatzbereit sein.

Zweitens müssen wir natürlich das Plugin installieren. Melden Sie sich dazu einfach bei Ihrem Admin-Dashboard an, gehen Sie zu Plugins und klicken Sie auf Neu hinzufügen. Dann müssen Sie nur noch den Namen des Plugins in das Suchfeld eingeben. So sehen Symbol und Name zum Zeitpunkt der Erstellung dieses Dokuments aus.



¹ wpDataTables – WordPress Tables & Table Charts Plugin, TMS-Plugins, <https://wordpress.org/plugins/wpdatatables/>



Nachdem Sie die CSV-Datei mit Ihren Daten gespeichert und das Plugin installiert haben, gehen Sie im nächsten Schritt zu Ihrem WordPress-Admin und erstellen eine neue wpDataTable aus der CSV-Datenquelle. Öffnen Sie Ihren WordPress-Adminbereich und:

1. Gehen Sie zu wpDataTables -> Tabelle erstellen und wählen Sie die Option Tabelle erstellen, die mit einer vorhandenen Datenquelle verknüpft ist.
2. Geben Sie in der Eingabe „Tabellentitel“ einen Namen für die neue wpDataTable ein, damit Sie sie leichter identifizieren können.
3. Wählen Sie „CSV-Datei“ als Eingabedatenquellentyp aus.
4. Die WordPressMedia-Bibliothek wird automatisch als Dateispeicherort ausgewählt und verwendet, wenn Sie eine Tabelle erstellen, indem Sie eine CSV-Datei hochladen.
5. Klicken Sie auf die Schaltfläche Durchsuchen, um die WordPress-Medienbibliothek zu öffnen.
6. Laden Sie die vorbereitete CSV-Datei hoch (oder wählen Sie eine der CSV-Dateien aus, die Sie zuvor in Ihre Medienbibliothek hochgeladen haben); Klicken Sie dann auf Ausgewählte Datei verwenden.
7. Klicken Sie auf Änderungen speichern, damit wpDataTables die von Ihnen hochgeladene CSV-Datei liest und die Metadaten der Spalten initialisiert.

Wenn die Konfiguration Ihrer CSV-basierten wpDataTable abgeschlossen ist, müssen Sie sie einfach in Ihren Beitrag oder Ihre Seite einfügen.

Öffnen Sie einen WordPress-Bertrag oder eine Seite, platzieren Sie den Cursor an der Stelle, an der Sie Ihre Tabelle einfügen möchten, klicken Sie im MCE-Editor-Panel auf die Schaltfläche „Eine wpDataTable einfügen“ und wählen Sie die CSV-basierte Tabelle aus, die Sie vorbereitet haben.

Oder, wenn Sie es vorziehen, die Dinge manuell zu erledigen, können Sie einfach den wpDataTable-Shortcode kopieren und einfügen (Sie finden ihn auf der wpDataTables-Durchsuchen-Seite oder auf der wpDataTable-Bearbeitungsseite).²

² How to create a table from CSV data, wpDataTables Team,
<https://wpdatatables.com/documentation/creating-wpdatatables/creating-wpdatatables-from-csv/>