How to Submit a Web Form

WE'LL COVER THE FOLLOWING

- Wrapping Up

In this section, we will compare how to submit a web form with requests versus urllib. Let's start by learning how to submit a web form. We will be doing a web search with **duckduckgo.com** searching on the term *python* and saving the result as an HTML file. We'll start with an example that uses urllib:

```
import urllib.request
import urllib.parse
import webbrowser

data = urllib.parse.urlencode({'q': 'Python'})
url = 'http://duckduckgo.com/html/'
full_url = url + '?' + data
response = urllib.request.urlopen(full_url)
with open("results.html", "wb") as f:
    f.write(response.read())

webbrowser.open("results.html")
```

The first thing you have to do when you want to submit a web form is figure out what the form is called and what the url is that you will be posting to. If you go to duckduckgo's website and view the source, you'll notice that its action is pointing to a relative link, "/html". So our url is "http://duckduckgo.com/html". The input field is named "q", so to pass duckduckgo a search term, we have to concatenate the url to the "q" field. The results are read and written to disk. Now let's find out how this process differs when using the requests package.

The requests package does form submissions a little bit more elegantly. Let's

The requests package does form submissions a little bit more elegantly. Let's take a look:

```
import requests

url = 'https://duckduckgo.com/html/'
payload = {'q':'python'}
r = requests.get(url, params=payload)
with open("requests_results.html", "wb") as f:
    f.write(r.content)
```

With requests, you just need to create a dictionary with the field name as the key and the search term as the value. Then you use **requests.get** to do the search. Finally you use the resulting requests object, "r", and access its content property which you save to disk.

Wrapping Up

Now you know the basics of the **requests** package. I would recommend reading the package's online documentation as it has many additional examples that you may find useful. I personally think this module is more intuitive to use than the standard library's equivalent.