

# What is a 'requests' package?

## WE'LL COVER THE FOLLOWING ^

- Using requests

The **requests** package is a more Pythonic replacement for Python's own **urllib**. You will find that requests package's API is quite a bit simpler to work with. You can install the requests library by using pip or easy\_install or from source.

## Using requests #

Let's take a look at a few examples of how to use the requests package. We will use a series of small code snippets to help explain how to use this library.

```
r = requests.get("http://www.google.com")
```



This example returns a **Response** object. You can use the Response object's methods to learn a lot about how you can use requests. Let's use Python's **dir** function to find out what methods we have available:

```
print(dir(r))
#[ '__attrs__', '__bool__', '__class__', '__delattr__', '__dict__',
# '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattr__',
# '__getstate__', '__gt__', '__hash__', '__init__', '__iter__', '__le__', '__lt__',
# '__module__', '__ne__', '__new__', '__nonzero__', '__reduce__', '__reduce_ex__',
# '__repr__', '__setattr__', '__setstate__', '__sizeof__', '__str__', '__subclasshook__',
# '__weakref__', '_content', '_content_consumed', 'apparent_encoding', 'close',
# 'connection', 'content', 'cookies', 'elapsed', 'encoding', 'headers', 'history',
# 'iter_content', 'iter_lines', 'json', 'links', 'ok', 'raise_for_status', 'raw',
# 'reason', 'request', 'status_code', 'text', 'url']
```



If you run the following method, you can see the web page's source code:

```
r.content()
```



The output from this command is way too long to include in the book, so be sure to try it out yourself. If you'd like to take a look at the web pages headers, you can run the following:

```
r.headers
```



Note that the **headers** attribute returns a dict-like object and isn't a function call. We're not showing the output as web page headers tend to be too wide to show correctly in a book. There are a bunch of other great functions and attributes in the Response object. For example, you can get the cookies, the links in the page, and the `status_code` that the page returned.

The requests package supports the following HTTP request types: POST, GET, PUT, DELETE, HEAD and OPTIONS. If the page returns json, you can access it by calling the Response object's **json** method. Let's take a look at a practical example.