Requests: Pythonic HTTP

Stephen Haywood AverageSecurityGuy

You can find me on:

Twitter - @averagesecguy

GitHub - averagesecurityguy

Web - averagesecurityguy.info

The Requests Library

The Requests library was written by Kenneth Reitz to provide a more Pythonic HTTP library. You can use httplib or urllib but Requests is much cleaner.

Simple Stuff Is Simple

Urllib2

```
import urllib2

url = 'http://server/page'
req = urllib2.Request(url)
r = urllib2.urlopen(req)
print r.read()
```

Requests

```
import requests

url = 'http://server/page'
resp = requests.get(url)
print resp.content
```

Complex Stuff is Simple

```
import requests

data = {"name": "stephen", "age": "34"}

resp = requests.put('http://server/page', data=data)
print resp.content
```

Complex Stuff is Simple

```
import urllib
import urllib2

data = {"name": "stephen", "age": "34"}
data = urllib.urlencode(data)

o = urllib2.build_opener(urllib2.HTTPHandler)
r = urllib2.Request('http://server/page', data=data)

# Must override the get_method to use PUT
r.get_method = lambda: 'PUT'

url = o.open(request)
print url.read()
```

Complex Stuff is Simple

```
import httplib

conn = httplib.HTTPConnection('server:80')
data = {"name": "stephen", "age": "34"}

conn.request('PUT', 'page', data)
resp = conn.getresponse()
print resp.read()
```

Use All The Verbs

```
# Requests supports all the verbs natively
r = requests.get('http://server/page')
r = requests.post('http://server/page')
r = requests.put('http://server/page')
r = requests.delete('http://server/page')
r = requests.head('http://server/page')
r = requests.options('http://server/page')

# Urllib2 only supports GET and POST natively

# The httplib library supports all the verbs but must
# build requests manually
```

Send Post Data or Query Parameters

```
params = {"id": "1", "resource": "contact"}
data = {"name": "stephen", "age": "34"}

# Send a dictionary as query parameters
r = requests.get('http://server/page',
                 params=data)

# Send a dictionary as POST data
r = requests.post('http://server/page',
                  data=data)

# Send both
r = requests.post('http://server/page',
                  data=data, params=params)
```

Send Files

```
# Sending multipart data is easy
files = {'file': open('report.xls', 'rb')}

r = requests.post('http://server/page',
                  files=files)

# Both urllib and httplib require building a custom
# method for encoding multipart data
```

Custom Headers

```
headers = {'content-type': 'application/json'}

r = requests.post('http://server/page',
                  headers=headers)
```

Authentication

```
# Basic Authentication
requests.get('https://server/test',
             auth=('user', 'pass'))

# Digest Authentication
from requests.auth import HTTPDigestAuth

url = 'http://server/page'
requests.get(url, auth=HTTPDigestAuth('user','pass'))
```

Custom Authentication

```
from requests.auth import AuthBase

class MyAuth(AuthBase):
    def __init__(self, username):
        # setup any auth-related data here
        self.username = username

    def __call__(self, r):
        # modify and return the request
        r.headers['X-Pizza'] = self.username
        return r

requests.get('http://server/page',
             auth=MyAuth('user'))
```

Sessions

```
s = requests.Session()

# These headers will be use throughout the session.
headers = {'content-type': 'application/json'}
s.headers.update(headers)

s.get('http://server/page')
```

Response Content

```
resp = requests.get('http://server/page')

# Print the response with the guessed encoding type.
print resp.text
print resp.encoding

# Set a specific encoding type
resp.encoding = 'utf-8'

# Print the raw content
print resp.content

# Print the json content, which is a Python dictionary
print resp.json()
```

Response Data

```
resp = requests.get('http://server/page')

# Print the response headers and status code
print resp.headers
print resp.status_code

# Print the request headers
resp.request.headers
```

Let's Look At Some Code

More Info

```
# Install requests with easy_install or pip
easy_install requests
pip install requests

# Visit the web site
http://docs.python-requests.org/en/latest/
```

Requests: Pythonic HTTP