urllib2 by example

Building a YouTube downloader

http://github.com/stefanha/youtube

Stefan Hajnoczi <stefanha@gmail.com>

Automating useful web tasks

- What web tasks can we do?
  - Grab 15 latest bookmarks from del.icio.us.
  - Post a tweet using Twitter's REST API.
  - Search Google programmatically.
  - **Download a video from YouTube.**

- Not all websites have APIs, some require *screenscraping*.
- Check for an official API first, it will make life easier.
- Try to be considerate when accessing a service.

- **Use urllib2 to automate web tasks!**

urllib2 - the web client module

- **Simple web requests are easy in Python:**
  - urllib2.urlopen('http://python.org/').read()
- But there is also built-in support for:
  - Redirection
  - Authentication
  - Custom headers
  - Cookies
  - HTTPS
  - Proxies
- ...batteries included!

- This presentation is only a basic example, read the module documentation for the details.

More about urllib2.urlopen()

urllib2.urlopen(**url_or_req**[, **post_data**][, **timeout**])

Returns a **file-like** object that you can .read([size]) the response body from.

url_or_req accepts a URL string. For advanced use, pass a urllib2.Request object that can include HTTP headers.

post_data is an optional urlencoded POST body string. Use urllib.urlencode() to encode arguments.

timeout raises urllib2.URLError if a blocking operation exceeds the optional timeout.

Start by fetching video page HTML

```
#!/usr/bin/env python
import urllib2, json, sys, re

if len(sys.argv) != 2:
    print 'usage: %s <video-url>' % \
          sys.argv[0]
    sys.exit(1)
watch_url = sys.argv[1]

# Fetch the video page HTML
watch_html = urllib2.urlopen(watch_url).read()
```

Extract and decode video file URL

```
# Search the HTML for URL of the video file
match = re.search(r"img.src = '([^']*)';",
                  watch_html)
if match is None:
    print 'unable to find flv url'
    sys.exit(1)

# The video file URL is extracted from
# some Javascript that needs decoding
flv_url = json.loads('"%s"' % \
    match.group(1)).replace(
    'generate_204', 'videoplayback')
print 'flv url:', flv_url
```

Finally download the video file

```
# Download the video file
open('video.flv', 'wb').
    write(urllib2.urlopen(flv_url).read())
```

Try it out...

```
$ ./youtube.py 'http://www.youtube.com/watch?
v=a1Y73sPHKxw'
```

More info on urllib2

Module documentation:
http://docs.python.org/release/2.6.6/library/urllib2.html#module-urllib2

Old but good overview including cookies:
http://www.voidspace.org.uk/python/articles/urllib2.shtml