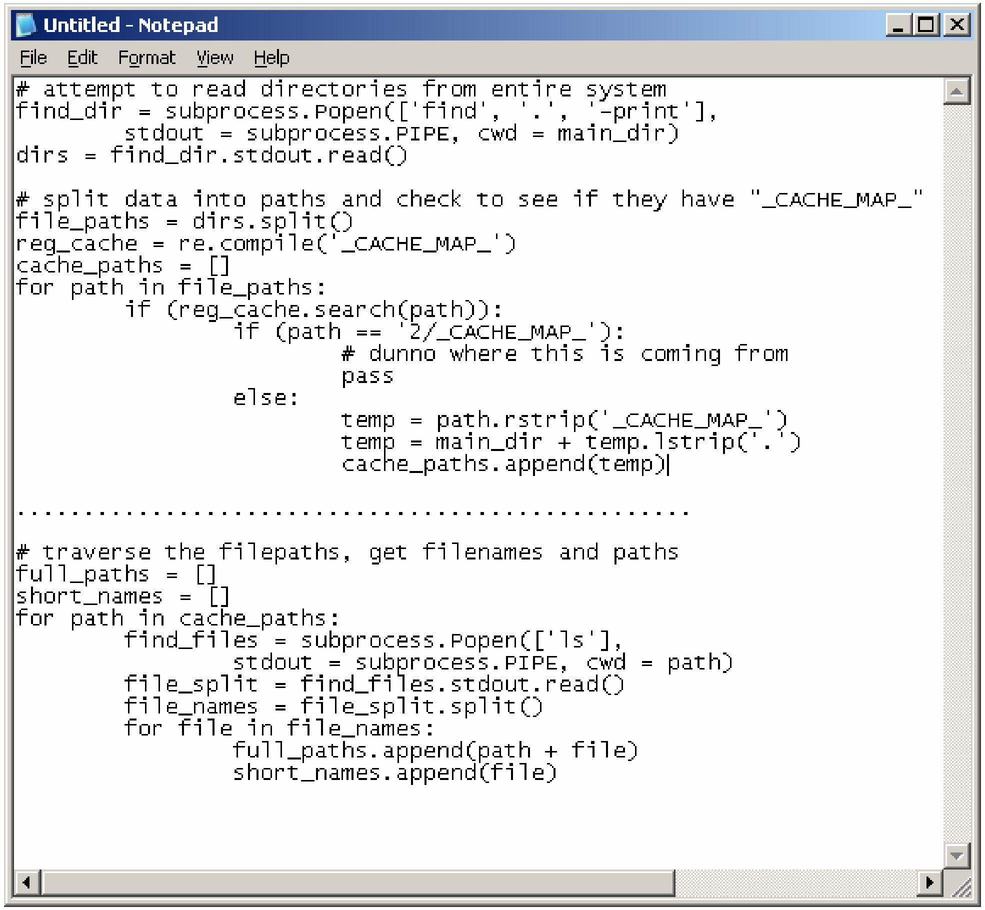
Storage Jars / Cache Stash Web Service and Surf Spy Client-Server

Or,
“What I did on My ~~Summer Vacation~~ Final Project”

By Will MacLean

Storage Jars / Cache Stash: Code Lowlights

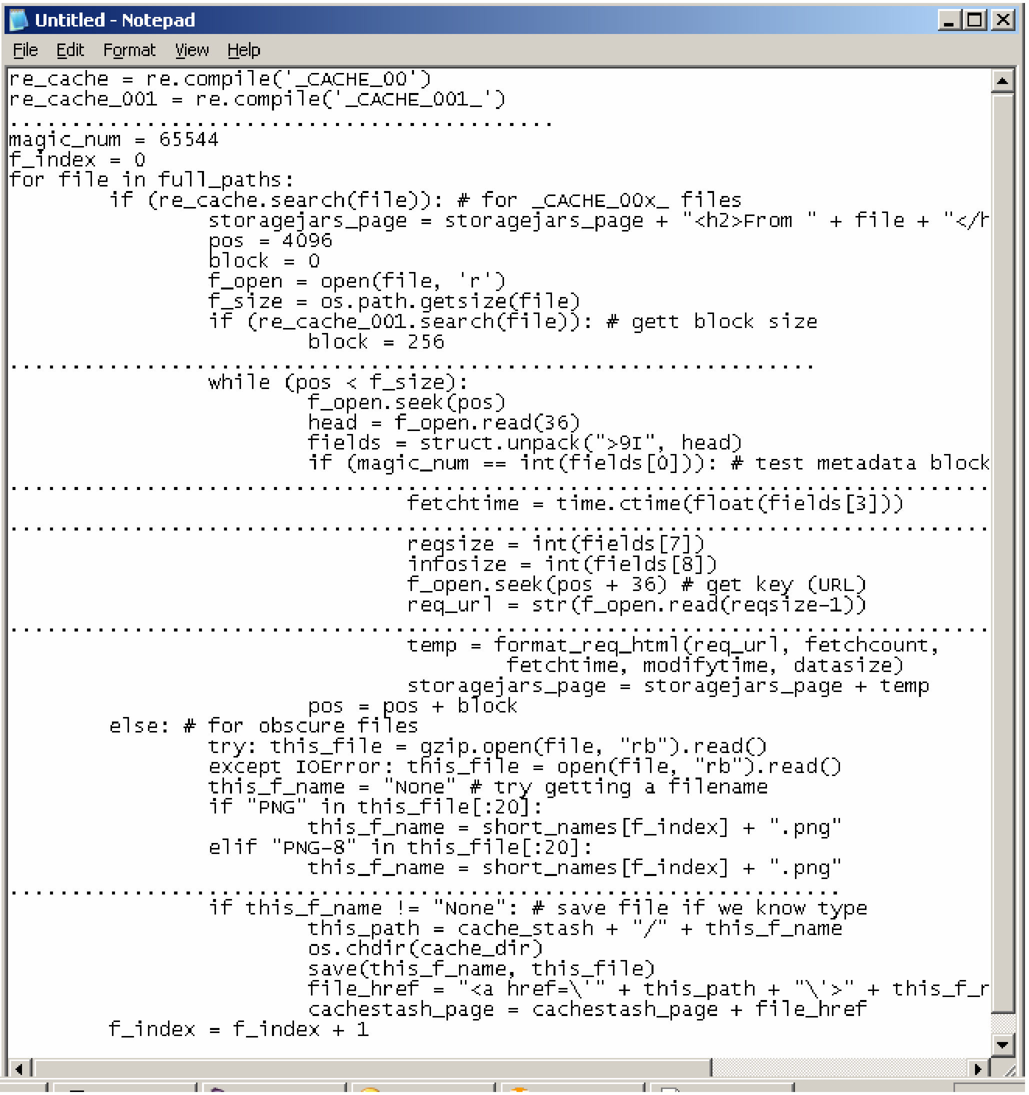
- Uses Popen with “ls” to get directories and files.
- Bad: longhand, looks static (like C)
- Bad: it creates a bad first result with dirs.
- Lesson learned: wait ‘til Prof gives helpful hints (os)



```
# attempt to read directories from entire system
find_dir = subprocess.Popen(['find', '.', '-print'],
                stdout = subprocess.PIPE, cwd = main_dir)
dirs = find_dir.stdout.read()

# split data into paths and check to see if they have "_CACHE_MAP_"
file_paths = dirs.split()
reg_cache = re.compile('_CACHE_MAP_')
cache_paths = []
for path in file_paths:
                if (reg_cache.search(path)):
                                if (path == '2/_CACHE_MAP_'):
                                                # dunno where this is coming from
                                                pass
                                else:
                                                temp = path.rstrip('_CACHE_MAP_')
                                                temp = main_dir + temp.lstrip('.')
                                                cache_paths.append(temp)

..................................................

# traverse the filepaths, get filenames and paths
full_paths = []
short_names = []
for path in cache_paths:
                find_files = subprocess.Popen(['ls'],
                                stdout = subprocess.PIPE, cwd = path)
                file_split = find_files.stdout.read()
                file_names = file_split.split()
                for file in file_names:
                                full_paths.append(path + file)
                                short_names.append(file)
```

Storage Jars / Cache Stash: Code Midlights

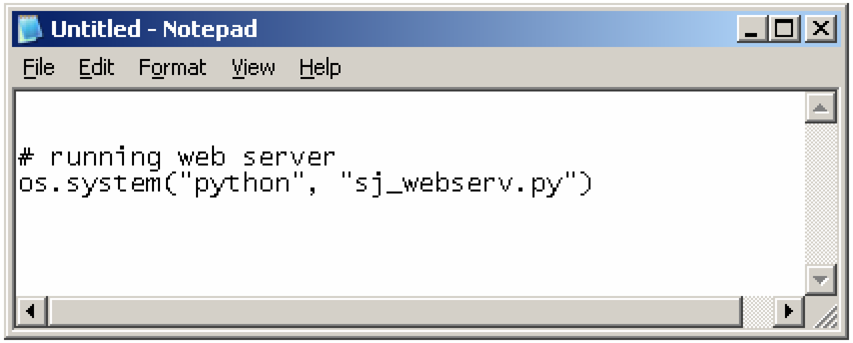
- I guess this is probably much how everyone did it
- Unless you tried something based on the Prof’s generator samples in lecture 9
- But, generators confused me



```
re_cache = re.compile('_CACHE_00')
re_cache_001 = re.compile('_CACHE_001_')
............................................
magic_num = 65544
f_index = 0
for file in full_paths:
        if (re_cache.search(file)): # for _CACHE_00x_ files
                storagejars_page = storagejars_page + "<h2>From " + file + "</h
                pos = 4096
                block = 0
                f_open = open(file, 'r')
                f_size = os.path.getsize(file)
                if (re_cache_001.search(file)): # gett block size
                        block = 256
.................................................................
                while (pos < f_size):
                        f_open.seek(pos)
                        head = f_open.read(36)
                        fields = struct.unpack(">9I", head)
                        if (magic_num == int(fields[0])): # test metadata block
...............................................................................
                                fetchtime = time.ctime(float(fields[3]))
...............................................................................
                                reqsize = int(fields[7])
                                infosize = int(fields[8])
                                f_open.seek(pos + 36) # get key (URL)
                                req_url = str(f_open.read(reqsize-1))
...............................................................................
                                temp = format_req_html(req_url, fetchcount,
                                        fetchtime, modifytime, datasize)
                                storagejars_page = storagejars_page + temp
                        pos = pos + block
        else: # for obscure files
                try: this_file = gzip.open(file, "rb").read()
                except IOError: this_file = open(file, "rb").read()
                this_f_name = "None" # try getting a filename
                if "PNG" in this_file[:20]:
                        this_f_name = short_names[f_index] + ".png"
                elif "PNG-8" in this_file[:20]:
                        this_f_name = short_names[f_index] + ".png"
...................................................................
                if this_f_name != "None": # save file if we know type
                        this_path = cache_stash + "/" + this_f_name
                        os.chdir(cache_dir)
                        save(this_f_name, this_file)
                        file_href = "<a href=\'" + this_path + "\'>" + this_f_r
                        cachestash_page = cachestash_page + file_href
        f_index = f_index + 1
```

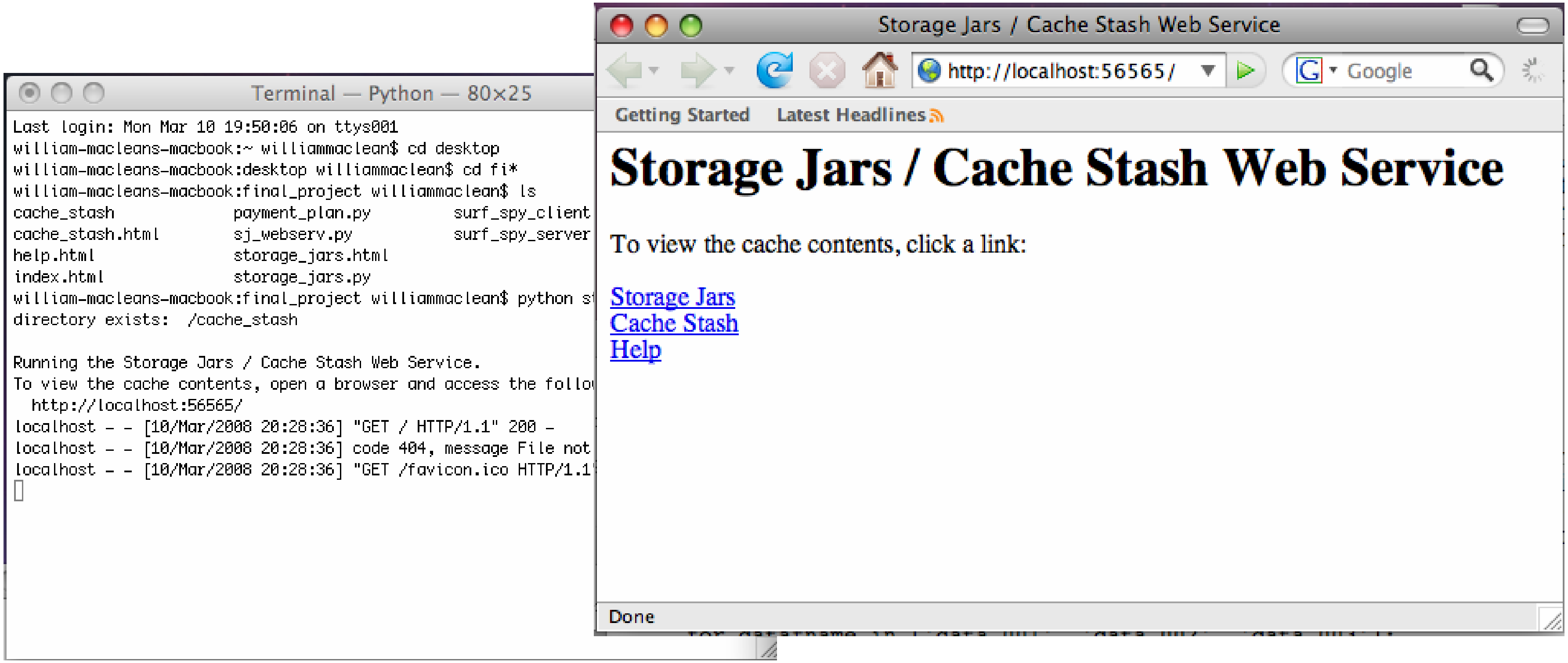
Storage Jars / Cache Stash: Code Highlights

- Autorun of web server
- Simple
- Neat
- Used os module after all
- Looks more like real Python code



```
# running web server
os.system("python", "sj_webserv.py")
```

Running the Web Service



```
Last login: Mon Mar 10 19:50:06 on ttys001
william-macleans-macbook:~ williammaclean$ cd desktop
william-macleans-macbook:desktop williammaclean$ cd fi*
william-macleans-macbook:final_project williammaclean$ ls
cache_stash             payment_plan.py         surf_spy_client
cache_stash.html        sj_webserv.py           surf_spy_server
help.html               storage_jars.html
index.html              storage_jars.py
william-macleans-macbook:final_project williammaclean$ python s
directory exists:  /cache_stash

Running the Storage Jars / Cache Stash Web Service.
To view the cache contents, open a browser and access the follo
  http://localhost:56565/
localhost - - [10/Mar/2008 20:28:36] "GET / HTTP/1.1" 200 -
localhost - - [10/Mar/2008 20:28:36] code 404, message File not
localhost - - [10/Mar/2008 20:28:36] "GET /favicon.ico HTTP/1.1
```

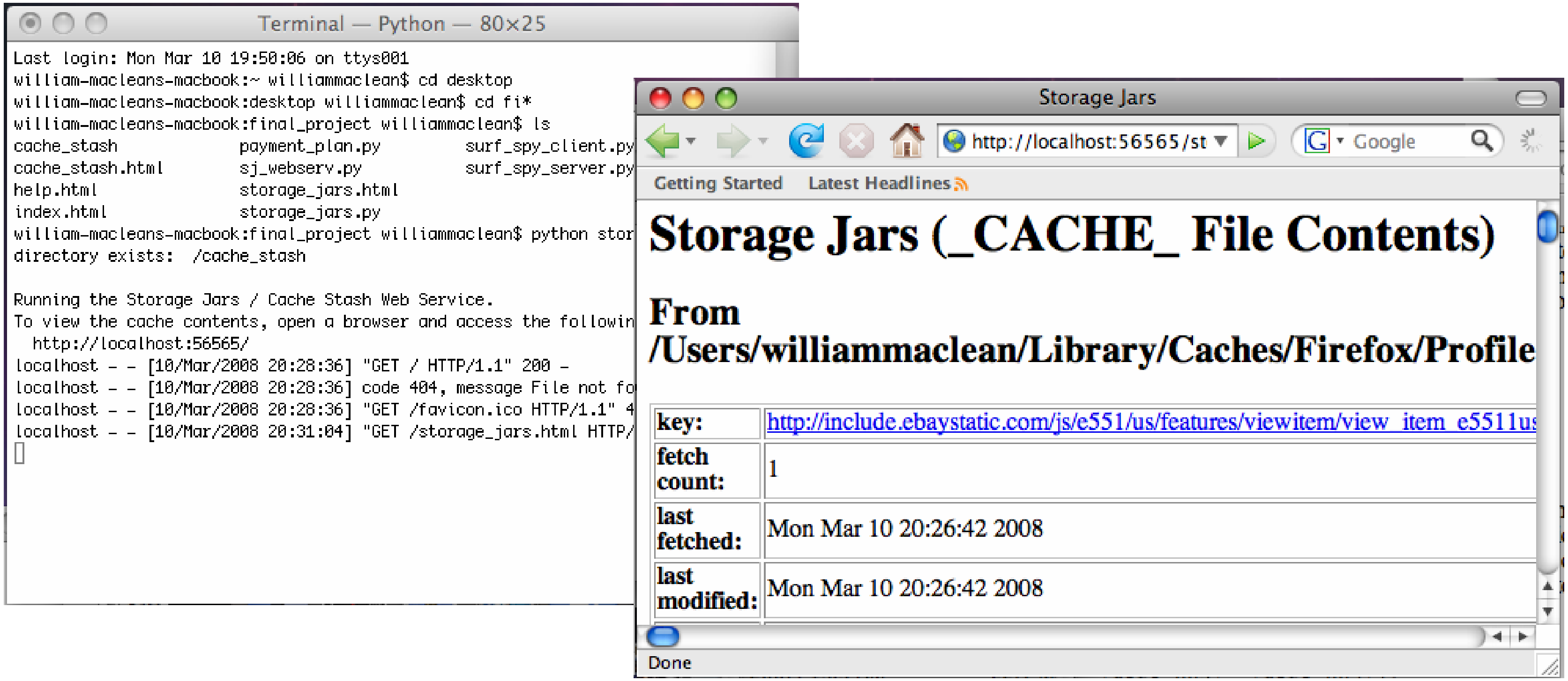
Storage Jars / Cache Stash Web Service

To view the cache contents, click a link:

Storage Jars
Cache Stash
Help

- Code scans for all _CACHE_ dirs
- Generates html
- Runs web server

Storage Jars



```
Last login: Mon Mar 10 19:50:06 on ttys001
william-macleans-macbook:~ williammaclean$ cd desktop
william-macleans-macbook:desktop williammaclean$ cd fi*
william-macleans-macbook:final_project williammaclean$ ls
cache_stash            payment_plan.py        surf_spy_client.py
cache_stash.html       sj_webserv.py          surf_spy_server.py
help.html              storage_jars.html
index.html             storage_jars.py
william-macleans-macbook:final_project williammaclean$ python stor
directory exists:  /cache_stash

Running the Storage Jars / Cache Stash Web Service.
To view the cache contents, open a browser and access the followin
  http://localhost:56565/
localhost - - [10/Mar/2008 20:28:36] "GET / HTTP/1.1" 200 -
localhost - - [10/Mar/2008 20:28:36] code 404, message File not fo
localhost - - [10/Mar/2008 20:28:36] "GET /favicon.ico HTTP/1.1" 4
localhost - - [10/Mar/2008 20:31:04] "GET /storage_jars.html HTTP/
```

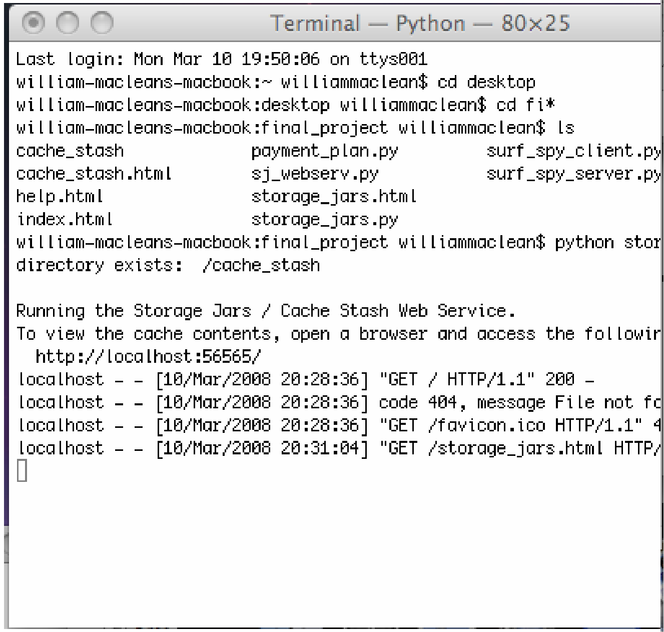
Storage Jars (_CACHE_ File Contents)

From /Users/williammaclean/Library/Caches/Firefox/Profile

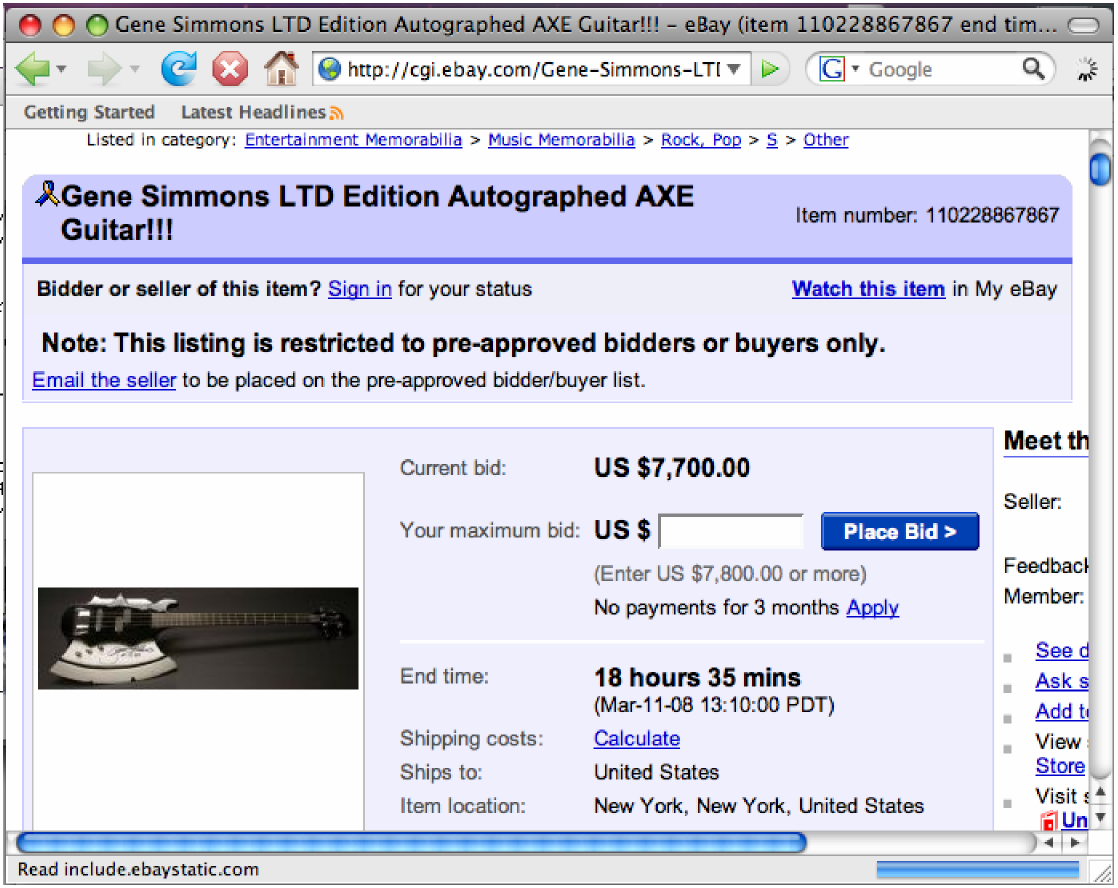
key:	http://include.ebaystatic.com/js/e551/us/features/viewitem/view_item_e5511us
fetch count:	1
last fetched:	Mon Mar 10 20:26:42 2008
last modified:	Mon Mar 10 20:26:42 2008

- Displays most information available in the Firefox cache
- Provides a link to web source

Storage Jars



```
Last login: Mon Mar 10 19:50:06 on ttys001
william-macleans-macbook:~ williammaclean$ cd desktop
william-macleans-macbook:desktop williammaclean$ cd fi*
william-macleans-macbook:final_project williammaclean$ ls
cache_stash             payment_plan.py         surf_spy_client.py
cache_stash.html        sj_webserv.py           surf_spy_server.py
help.html               storage_jars.html
index.html              storage_jars.py
william-macleans-macbook:final_project williammaclean$ python stor
directory exists:  /cache_stash

Running the Storage Jars / Cache Stash Web Service.
To view the cache contents, open a browser and access the followir
  http://localhost:56565/
localhost - - [10/Mar/2008 20:28:36] "GET / HTTP/1.1" 200 -
localhost - - [10/Mar/2008 20:28:36] code 404, message File not fc
localhost - - [10/Mar/2008 20:28:36] "GET /favicon.ico HTTP/1.1" 4
localhost - - [10/Mar/2008 20:31:04] "GET /storage_jars.html HTTP/
▯
```


Gene Simmons LTD Edition Autographed AXE Guitar!!! – eBay (item 110228867867 end tim...
http://cgi.ebay.com/Gene-Simmons-LTI
Google
Getting Started
Latest Headlines
Place Bid >
Read include.ebaystatic.com

Listed in category: Entertainment Memorabilia > Music Memorabilia > Rock, Pop > S > Other

Gene Simmons LTD Edition Autographed AXE Guitar!!!

Item number: 110228867867

Bidder or seller of this item? Sign in for your status

Watch this item in My eBay

Note: This listing is restricted to pre-approved bidders or buyers only.

Email the seller to be placed on the pre-approved bidder/buyer list.

Current bid: **US $7,700.00**

Your maximum bid: **US $** ▭

(Enter US $7,800.00 or more)

No payments for 3 months Apply

End time: **18 hours 35 mins**
(Mar-11-08 13:10:00 PDT)

Shipping costs: Calculate

Ships to: United States

Item location: New York, New York, United States

Meet th

Seller:

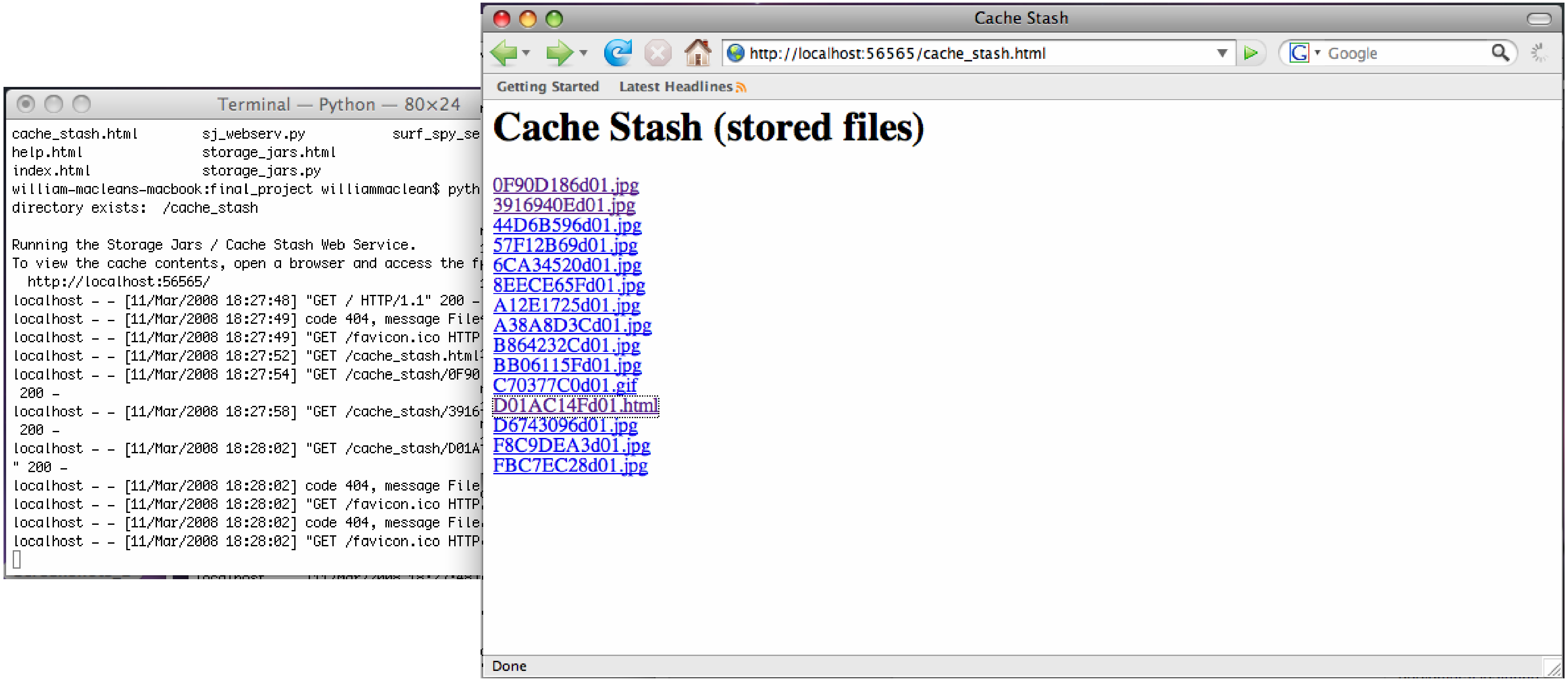
Feedback

Member:

- See d
- Ask s
- Add t
- View Store
- Visit Un

- Custom bass from Gene Simmons of KISS (cheap, too).

Cache Stash



Terminal — Python — 80×24
Cache Stash
http://localhost:56565/cache_stash.html
Google
Getting Started
Latest Headlines
Done

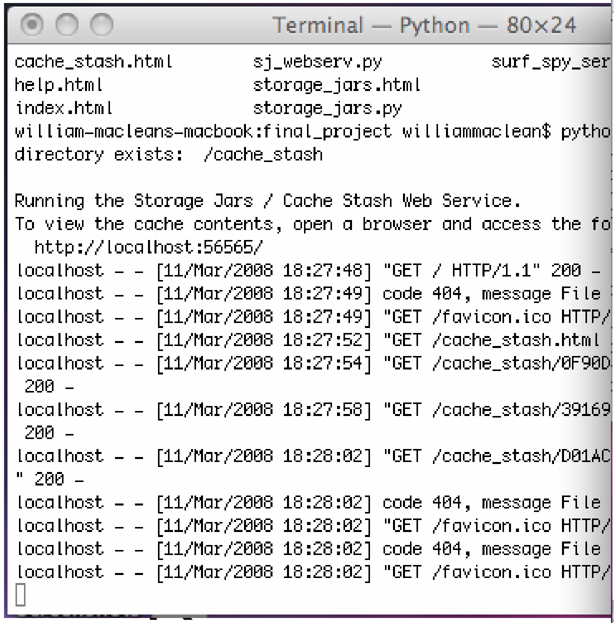
```
cache_stash.html        sj_webserv.py           surf_spy_se
help.html               storage_jars.html
index.html              storage_jars.py
william-macleans-macbook:final_project williammaclean$ pyth
directory exists:  /cache_stash

Running the Storage Jars / Cache Stash Web Service.
To view the cache contents, open a browser and access the f
  http://localhost:56565/
localhost - - [11/Mar/2008 18:27:48] "GET / HTTP/1.1" 200 -
localhost - - [11/Mar/2008 18:27:49] code 404, message File
localhost - - [11/Mar/2008 18:27:49] "GET /favicon.ico HTTP
localhost - - [11/Mar/2008 18:27:52] "GET /cache_stash.html
localhost - - [11/Mar/2008 18:27:54] "GET /cache_stash/0F90
 200 -
localhost - - [11/Mar/2008 18:27:58] "GET /cache_stash/3916
 200 -
localhost - - [11/Mar/2008 18:28:02] "GET /cache_stash/D01A
" 200 -
localhost - - [11/Mar/2008 18:28:02] code 404, message File
localhost - - [11/Mar/2008 18:28:02] "GET /favicon.ico HTTP
localhost - - [11/Mar/2008 18:28:02] code 404, message File
localhost - - [11/Mar/2008 18:28:02] "GET /favicon.ico HTTP
```

Cache Stash (stored files)

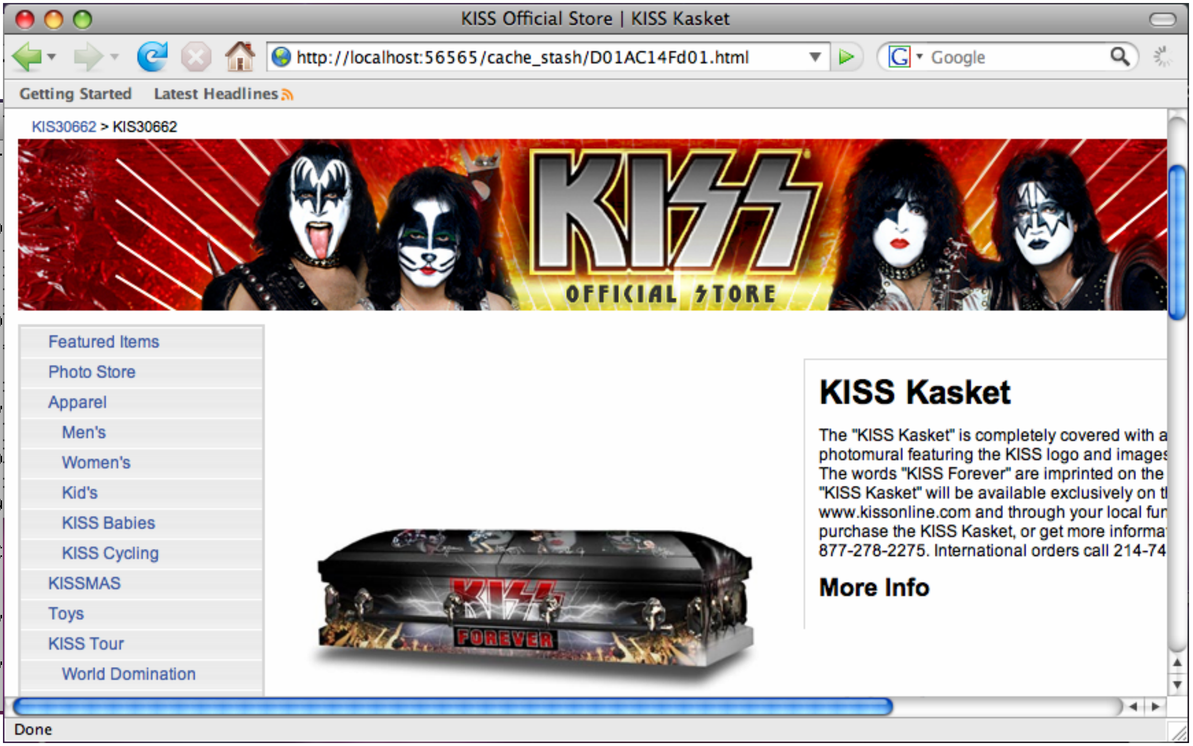
0F90D186d01.jpg
3916940Ed01.jpg
44D6B596d01.jpg
57F12B69d01.jpg
6CA34520d01.jpg
8EECE65Fd01.jpg
A12E1725d01.jpg
A38A8D3Cd01.jpg
B864232Cd01.jpg
BB06115Fd01.jpg
C70377C0d01.gif
D01AC14Fd01.html
D6743096d01.jpg
F8C9DEA3d01.jpg
FBC7EC28d01.jpg

- Provides links to copies of locally-stored files in cache

Viewing Stashed Caches



```
cache_stash.html          sj_webserv.py             surf_spy_ser
help.html                 storage_jars.html
index.html                storage_jars.py
william-macleans-macbook:final_project williammaclean$ pytho
directory exists:  /cache_stash

Running the Storage Jars / Cache Stash Web Service.
To view the cache contents, open a browser and access the fo
  http://localhost:56565/
localhost - - [11/Mar/2008 18:27:48] "GET / HTTP/1.1" 200 -
localhost - - [11/Mar/2008 18:27:49] code 404, message File
localhost - - [11/Mar/2008 18:27:49] "GET /favicon.ico HTTP/
localhost - - [11/Mar/2008 18:27:52] "GET /cache_stash.html
localhost - - [11/Mar/2008 18:27:54] "GET /cache_stash/0F90D
 200 -
localhost - - [11/Mar/2008 18:27:58] "GET /cache_stash/39169
 200 -
localhost - - [11/Mar/2008 18:28:02] "GET /cache_stash/D01AC
" 200 -
localhost - - [11/Mar/2008 18:28:02] code 404, message File
localhost - - [11/Mar/2008 18:28:02] "GET /favicon.ico HTTP/
localhost - - [11/Mar/2008 18:28:02] code 404, message File
localhost - - [11/Mar/2008 18:28:02] "GET /favicon.ico HTTP/
```



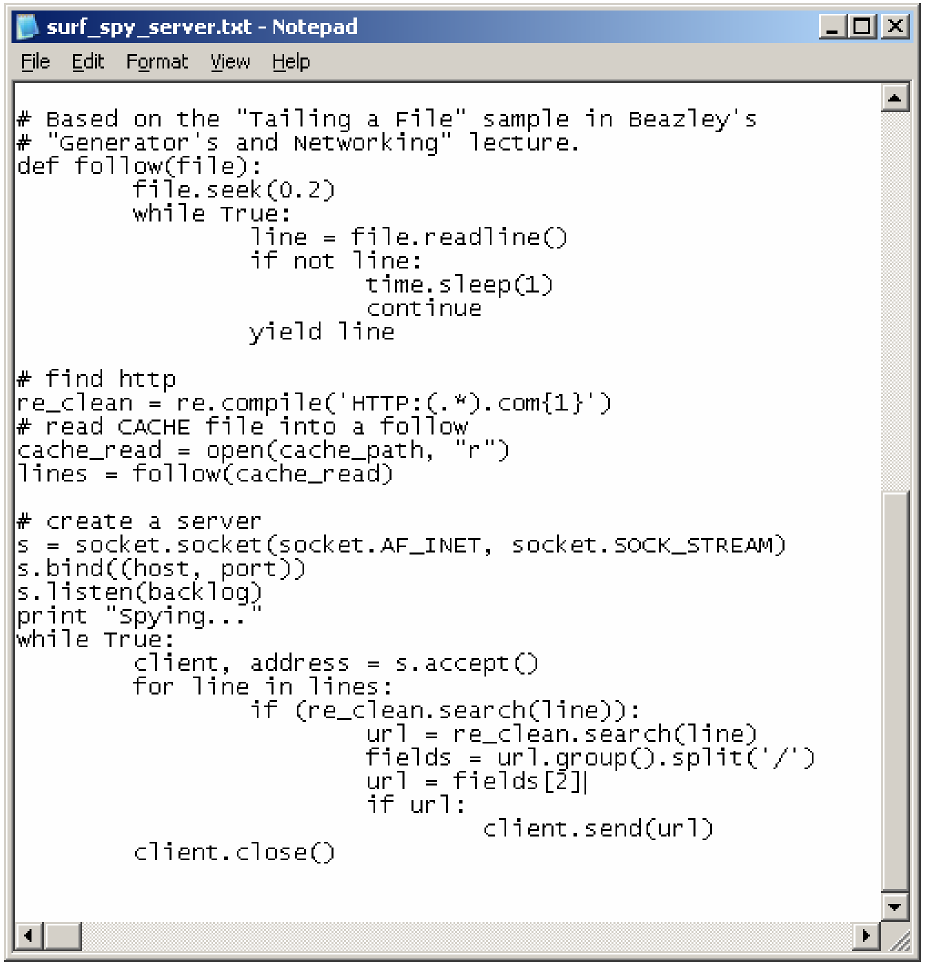
KISS Kasket

The "KISS Kasket" is completely covered with a photomural featuring the KISS logo and images The words "KISS Forever" are imprinted on the "KISS Kasket" will be available exclusively on t www.kissonline.com and through your local fur purchase the KISS Kasket, or get more informa 877-278-2275. International orders call 214-74

- KISS Kasket (retail - $5,000)
- Lesson learned: Gene Simmons of KISS is a greedy merchandising opportunist

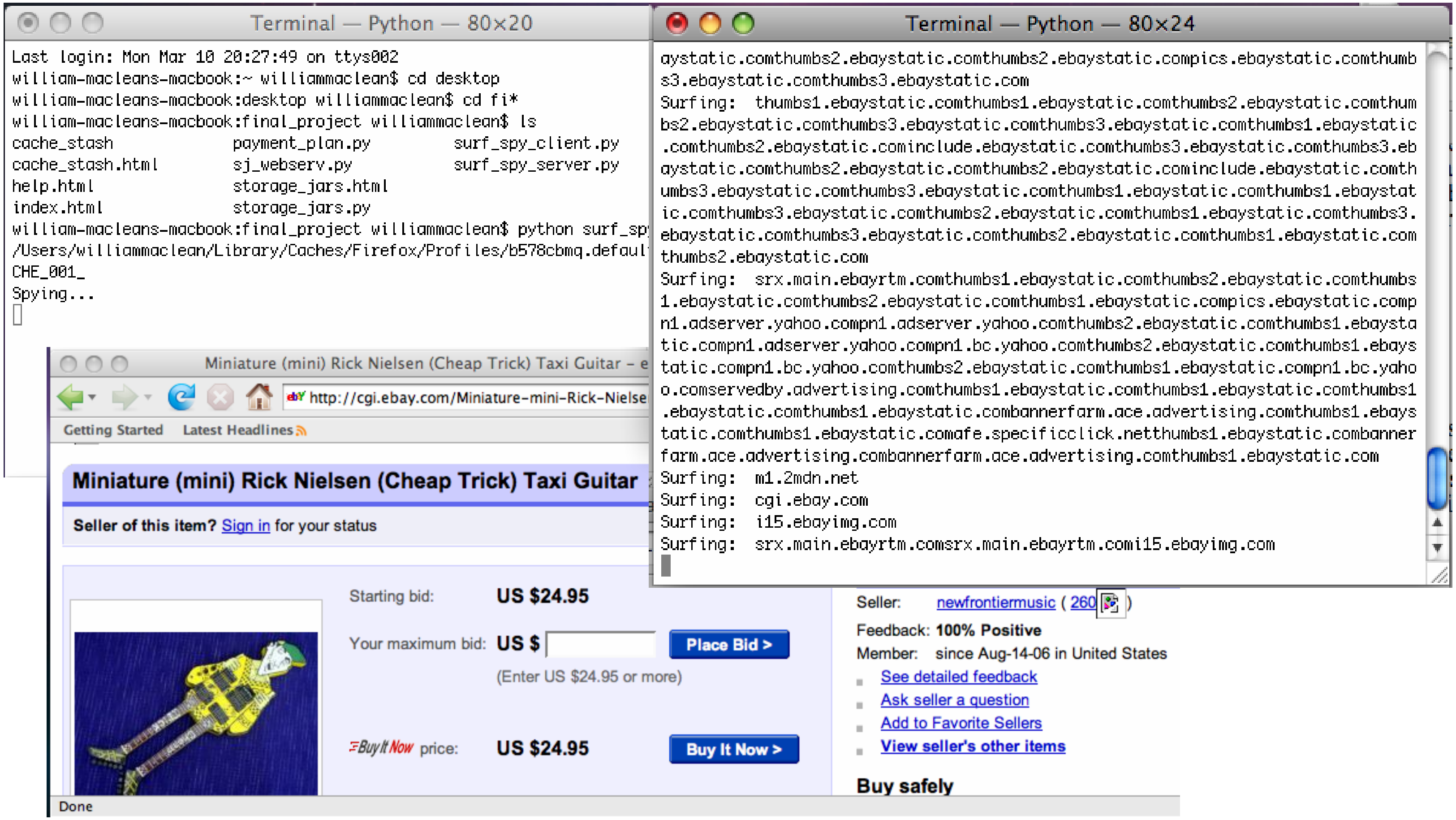
Surf Spy Client-Server: Code High/Low-lights

- Highlight:
  I used a generator, after all
- Lowlight:
  I was lazy, and read the _CACHE_ file as a string, instead of seeking and unpacking



```
# Based on the "Tailing a File" sample in Beazley's
# "Generator's and Networking" lecture.
def follow(file):
        file.seek(0.2)
        while True:
                line = file.readline()
                if not line:
                        time.sleep(1)
                        continue
                yield line

# find http
re_clean = re.compile('HTTP:(.*).com{1}')
# read CACHE file into a follow
cache_read = open(cache_path, "r")
lines = follow(cache_read)

# create a server
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((host, port))
s.listen(backlog)
print "Spying..."
while True:
        client, address = s.accept()
        for line in lines:
                if (re_clean.search(line)):
                        url = re_clean.search(line)
                        fields = url.group().split('/')
                        url = fields[2]
                        if url:
                                client.send(url)
        client.close()
```

Surf Spy In Action



```
Last login: Mon Mar 10 20:27:49 on ttys002
william-macleans-macbook:~ williammaclean$ cd desktop
william-macleans-macbook:desktop williammaclean$ cd fi*
william-macleans-macbook:final_project williammaclean$ ls
cache_stash             payment_plan.py         surf_spy_client.py
cache_stash.html        sj_webserv.py           surf_spy_server.py
help.html               storage_jars.html
index.html              storage_jars.py
william-macleans-macbook:final_project williammaclean$ python surf_sp
/Users/williammaclean/Library/Caches/Firefox/Profiles/b578cbmq.defaul
CHE_001_
Spying...
```

Miniature (mini) Rick Nielsen (Cheap Trick) Taxi Guitar – e
http://cgi.ebay.com/Miniature-mini-Rick-Nielse
Getting Started
Latest Headlines

Miniature (mini) Rick Nielsen (Cheap Trick) Taxi Guitar

Seller of this item? Sign in for your status

Starting bid: US $24.95

Your maximum bid: US $

(Enter US $24.95 or more)

Buy It Now price: US $24.95

Seller: newfrontiermusic (260)

Feedback: 100% Positive

Member: since Aug-14-06 in United States

- See detailed feedback
- Ask seller a question
- Add to Favorite Sellers
- View seller's other items

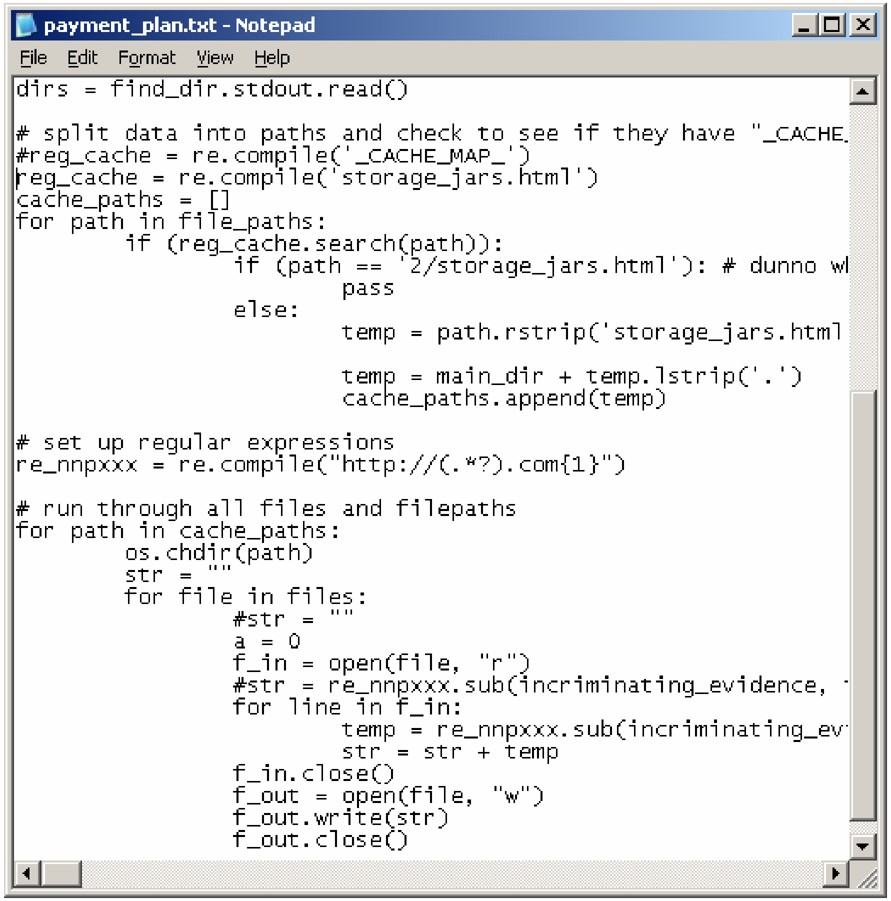
Buy safely

```
aystatic.comthumbs2.ebaystatic.comthumbs2.ebaystatic.compics.ebaystatic.comthumb
s3.ebaystatic.comthumbs3.ebaystatic.com
Surfing:  thumbs1.ebaystatic.comthumbs1.ebaystatic.comthumbs2.ebaystatic.comthum
bs2.ebaystatic.comthumbs3.ebaystatic.comthumbs3.ebaystatic.comthumbs1.ebaystatic
.comthumbs2.ebaystatic.cominclude.ebaystatic.comthumbs3.ebaystatic.comthumbs3.eb
aystatic.comthumbs2.ebaystatic.comthumbs2.ebaystatic.cominclude.ebaystatic.comth
umbs3.ebaystatic.comthumbs3.ebaystatic.comthumbs1.ebaystatic.comthumbs1.ebaystat
ic.comthumbs3.ebaystatic.comthumbs2.ebaystatic.comthumbs1.ebaystatic.comthumbs3.
ebaystatic.comthumbs3.ebaystatic.comthumbs2.ebaystatic.comthumbs1.ebaystatic.com
thumbs2.ebaystatic.com
Surfing:  srx.main.ebayrtm.comthumbs1.ebaystatic.comthumbs2.ebaystatic.comthumbs
1.ebaystatic.comthumbs2.ebaystatic.comthumbs1.ebaystatic.compics.ebaystatic.comp
n1.adserver.yahoo.compn1.adserver.yahoo.comthumbs2.ebaystatic.comthumbs1.ebaysta
tic.compn1.adserver.yahoo.compn1.bc.yahoo.comthumbs2.ebaystatic.comthumbs1.ebays
tatic.compn1.bc.yahoo.comthumbs2.ebaystatic.comthumbs1.ebaystatic.compn1.bc.yaho
o.comservedby.advertising.comthumbs1.ebaystatic.comthumbs1.ebaystatic.comthumbs1
.ebaystatic.comthumbs1.ebaystatic.combannerfarm.ace.advertising.comthumbs1.ebays
tatic.comthumbs1.ebaystatic.comafe.specificclick.netthumbs1.ebaystatic.combanner
farm.ace.advertising.combannerfarm.ace.advertising.comthumbs1.ebaystatic.com
Surfing:  m1.2mdn.net
Surfing:  cgi.ebay.com
Surfing:  i15.ebayimg.com
Surfing:  srx.main.ebayrtm.comsrx.main.ebayrtm.comi15.ebayimg.com
```

- Now here is a cool guitar

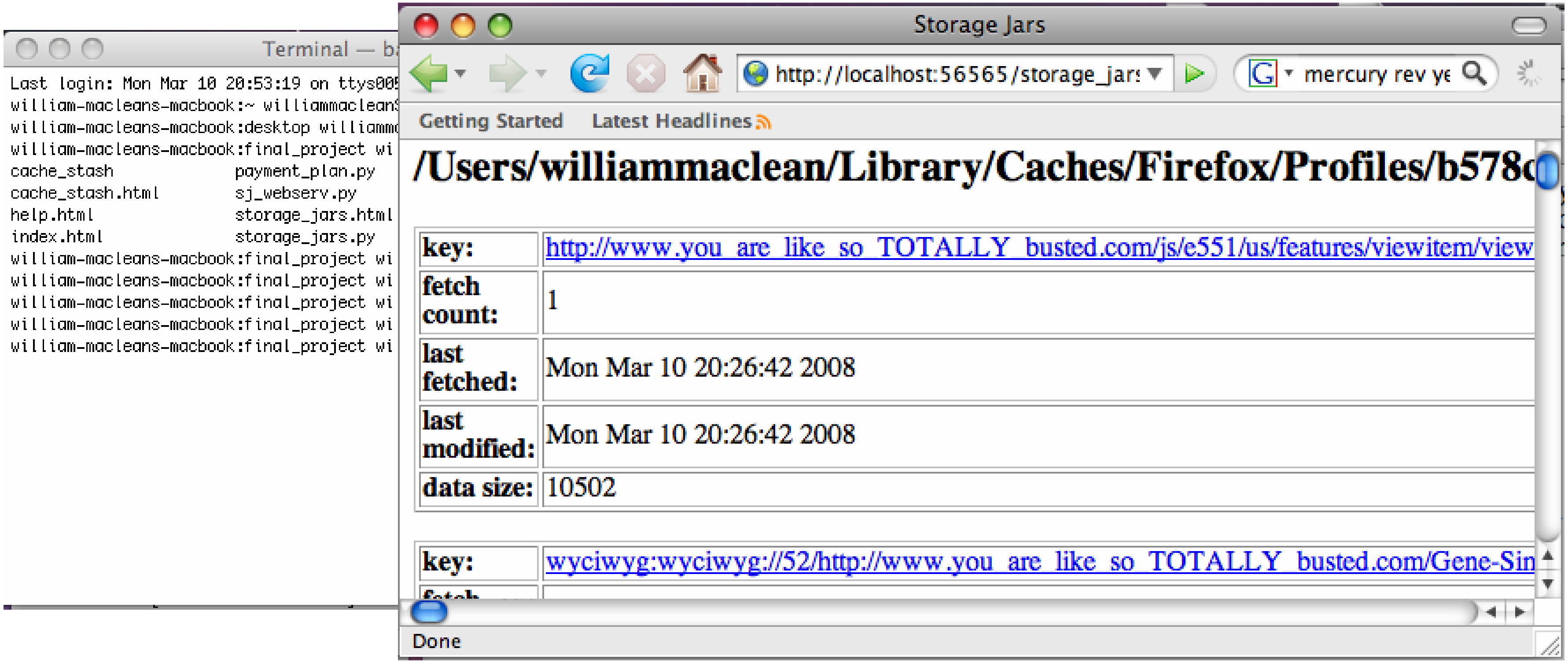
“Payment Plan” – Going for Evil: Lowlights

- The whole thing
- Dirty and devious
- Underhanded and unethical
- Sleazy and slimey
- Blackmail… pure and simple
- On top of that, I could not figure out how to mod _CACHE_ file while adjusting reqsize
- Boring code, too



```
dirs = find_dir.stdout.read()

# split data into paths and check to see if they have "_CACHE_
#reg_cache = re.compile('_CACHE_MAP_')
reg_cache = re.compile('storage_jars.html')
cache_paths = []
for path in file_paths:
        if (reg_cache.search(path)):
                if (path == '2/storage_jars.html'): # dunno w
                        pass
                else:
                        temp = path.rstrip('storage_jars.html

                        temp = main_dir + temp.lstrip('.')
                        cache_paths.append(temp)

# set up regular expressions
re_nnpxxx = re.compile("http://(.*?).com{1}")

# run through all files and filepaths
for path in cache_paths:
        os.chdir(path)
        str = ""
        for file in files:
                #str = ""
                a = 0
                f_in = open(file, "r")
                #str = re_nnpxxx.sub(incriminating_evidence,
                for line in f_in:
                        temp = re_nnpxxx.sub(incriminating_ev
                        str = str + temp
                f_in.close()
                f_out = open(file, "w")
                f_out.write(str)
                f_out.close()
```

“Payment Plan”



```
Last login: Mon Mar 10 20:53:19 on ttys005
william-macleans-macbook:~ williammaclean$
william-macleans-macbook:desktop williamm
william-macleans-macbook:final_project wi
cache_stash               payment_plan.py
cache_stash.html          sj_webserv.py
help.html                 storage_jars.html
index.html                storage_jars.py
william-macleans-macbook:final_project wi
william-macleans-macbook:final_project wi
william-macleans-macbook:final_project wi
william-macleans-macbook:final_project wi
william-macleans-macbook:final_project wi
```

/Users/williammaclean/Library/Caches/Firefox/Profiles/b578c

key:	http://www.you_are_like_so_TOTALLY_busted.com/js/e551/us/features/viewitem/view
fetch count:	1
last fetched:	Mon Mar 10 20:26:42 2008
last modified:	Mon Mar 10 20:26:42 2008
data size:	10502

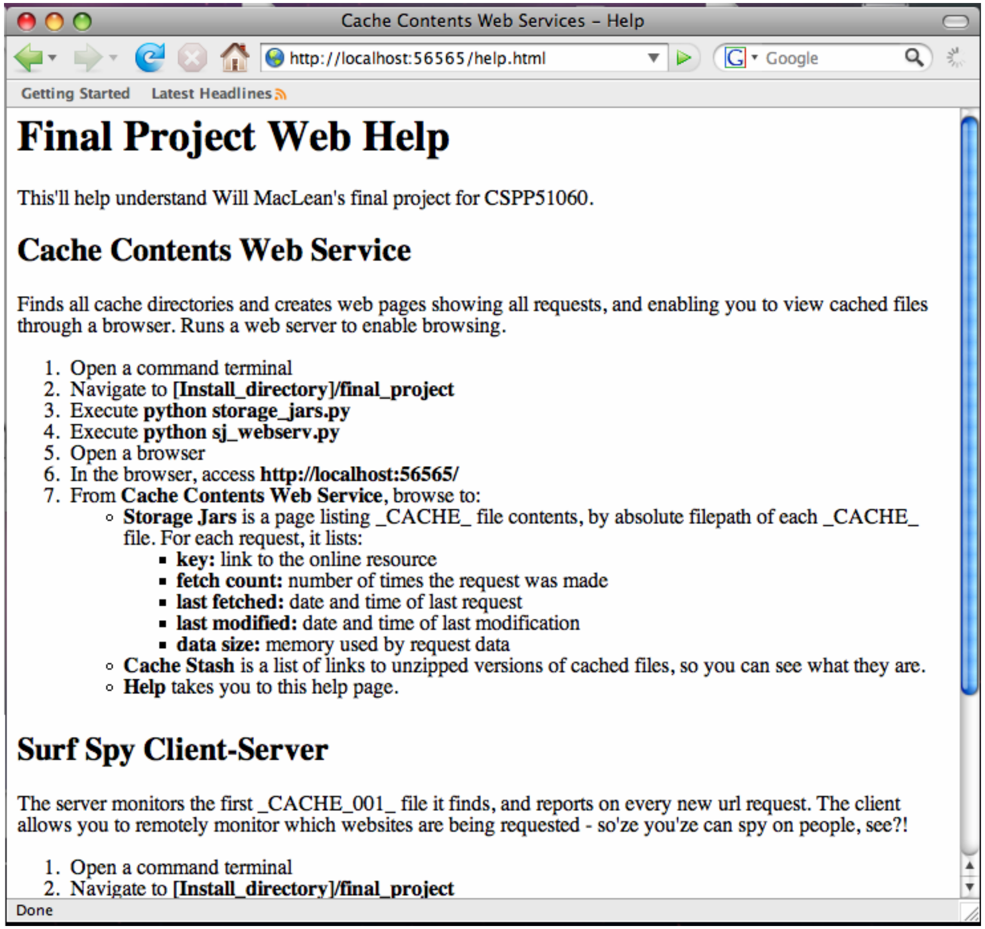
key:	wyciwyg:wyciwyg://52/http://www.you_are_like_so_TOTALLY_busted.com/Gene-Sin
fetch	

- “’ello ‘ello – whot ‘ave we ‘ere?!”

-Anonymous Police Constable

Help

- This is how I make my living
- So, this was the dull part


Cache Contents Web Services – Help
http://localhost:56565/help.html
Google
Getting Started
Latest Headlines
Done

Final Project Web Help

This'll help understand Will MacLean's final project for CSPP51060.

Cache Contents Web Service

Finds all cache directories and creates web pages showing all requests, and enabling you to view cached files through a browser. Runs a web server to enable browsing.

1. Open a command terminal
2. Navigate to **[Install_directory]/final_project**
3. Execute **python storage_jars.py**
4. Execute **python sj_webserv.py**
5. Open a browser
6. In the browser, access **http://localhost:56565/**
7. From **Cache Contents Web Service**, browse to:
   - **Storage Jars** is a page listing _CACHE_ file contents, by absolute filepath of each _CACHE_ file. For each request, it lists:
     - **key:** link to the online resource
     - **fetch count:** number of times the request was made
     - **last fetched:** date and time of last request
     - **last modified:** date and time of last modification
     - **data size:** memory used by request data
   - **Cache Stash** is a list of links to unzipped versions of cached files, so you can see what they are.
   - **Help** takes you to this help page.

Surf Spy Client-Server

The server monitors the first _CACHE_001_ file it finds, and reports on every new url request. The client allows you to remotely monitor which websites are being requested - so'ze you'ze can spy on people, see?!

1. Open a command terminal
2. Navigate to **[Install_directory]/final_project**