Beyond HTTP POST

HTTP isn’t limited to `GET` and `POST`. Those are certainly the most common types of requests, especially in web browsers. But web service APIs can go beyond `GET` and `POST`, and `httplib2` is ready.

```
# continued from the previous example
from xml.etree import ElementTree as etree
tree = etree.fromstring(content)                                          #①
status_id = tree.findtext('id')                                           #②
status_id
#'5131472'

url = 'https://identi.ca/api/statuses/destroy/{0}.xml'.format(status_id)  #③
resp, deleted_content = h.request(url, 'DELETE')                          #④
```

① The server returned XML, right? You know how to parse XML.

② The `findtext()` method finds the first instance of the given expression and extracts its text content. In this case, we’re just looking for an `<id>` element.

③ Based on the text content of the `<id>` element, we can construct a URL to delete the status message we just published.

④ To delete a message, you simply issue an HTTP `DELETE` request to that URL.

This is what goes over the wire:

```
send: b'DELETE /api/statuses/destroy/5131472.xml HTTP/1.1      #①
Host: identi.ca
Accept-Encoding: identity
user-agent: Python-httplib2/$Rev: 259 $

'
reply: 'HTTP/1.1 401 Unauthorized'                             #②
send: b'DELETE /api/statuses/destroy/5131472.xml HTTP/1.1      #③
Host: identi.ca
Accept-Encoding: identity
authorization: Basic SECRET_HASH_CONSTRUCTED_BY_HTTPLIB2       #④
user-agent: Python-httplib2/$Rev: 259 $
```

```
'
reply: 'HTTP/1.1 200 OK'                                      #⑤


resp.status
#200
```

① “Delete this status message.”

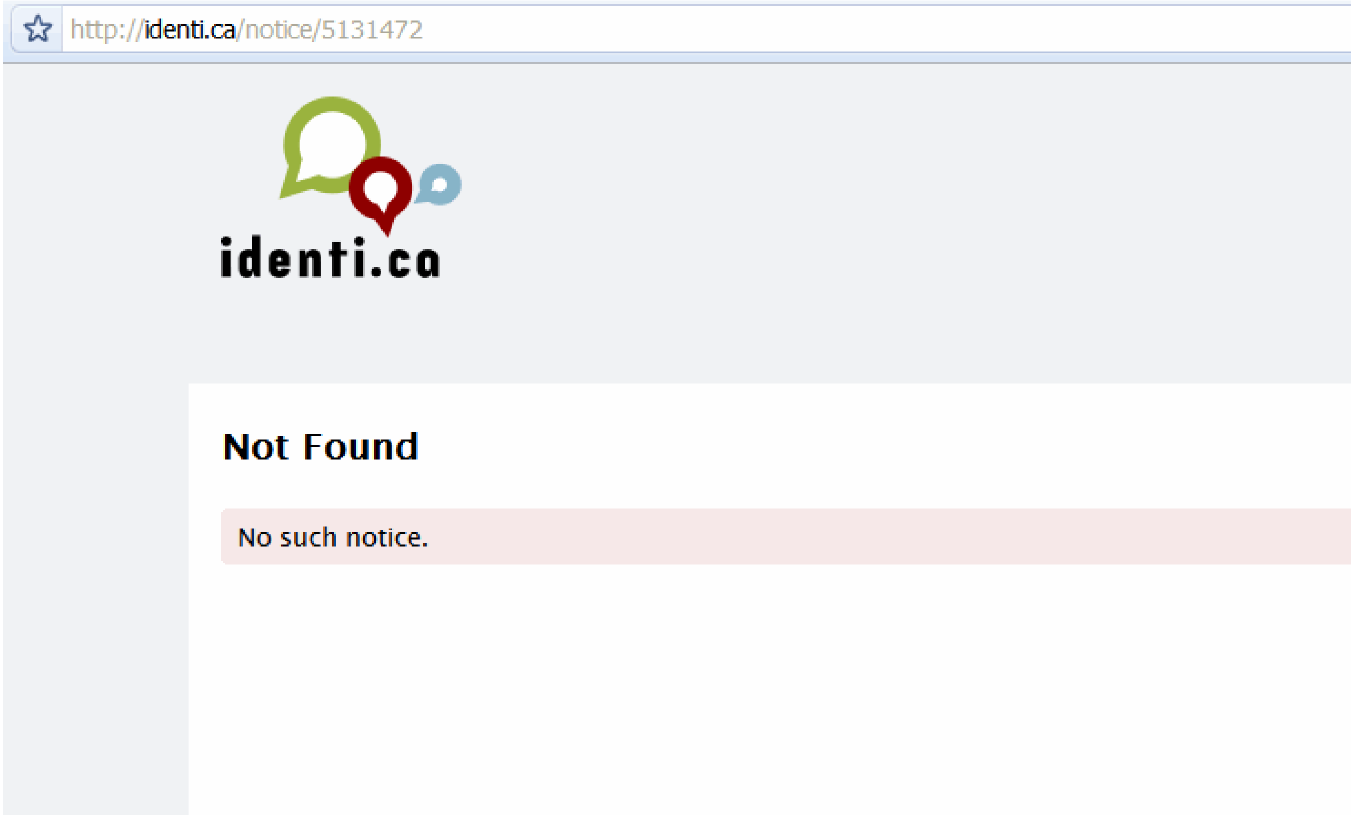
② “I’m sorry, Dave, I’m afraid I can’t do that.”

③ “Unauthorized‽ Hmmph. Delete this status message, please…

④ …and here’s my username and password.”

⑤ “Consider it done!”

And just like that, poof, it’s gone.



http://identi.ca/notice/5131472

Not Found

No such notice.