I know its been said already, but I'd highly recommend the Requests python package:<http://docs.python-requests.org/en/latest/index.html>

If you've used languages other than python, you're probably thinking urllib and urllib2 are easy to use, not much code, and highly capable, that's how I used to think. But the Requests package is so unbelievably useful and short that everyone should be using it.

First, it supports a fully restful API, and is as easy as:

import requests

...

resp = requests.get('http://www.mywebsite.com/user')

resp = requests.post('http://www.mywebsite.com/user')

resp = requests.put('http://www.mywebsite.com/user/put')

resp = requests.delete('http://www.mywebsite.com/user/delete')

Regardless of whether GET/POST you never have to encode parameters again, it simply takes a dictionary as an argument and is good to go.

userdata = {"firstname": "John", "lastname": "Doe", "password": "jdoe123"}

resp = requests.post('http://www.mywebsite.com/user', params=userdata)

Plus it even has a built in json decoder (again, i know json.loads() isn't a lot more to write, but this sure is convenient):

resp.json()

Or if your response data is just text, use:

resp.text

This is just the tip of the iceberg. This is the list of features from the requests site:

* International Domains and URLs
* Keep-Alive & Connection Pooling
* Sessions with Cookie Persistence
* Browser-style SSL Verification
* Basic/Digest Authentication
* Elegant Key/Value Cookies
* Automatic Decompression
* Unicode Response Bodies
* Multipart File Uploads
* Connection Timeouts
* .netrc support
* List item
* Python 2.6—3.4
* Thread-safe.

<http://stackoverflow.com/questions/2018026/what-are-the-differences-between-the-urllib-urllib2-and-requests-module>

Using Telnet

GET /articles/http-basics HTTP/1.1

Host: www.articles.com

Connection: keep-alive

Cache-Control: no-cache

Pragma: no-cache

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8