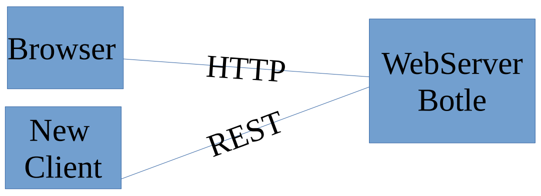
MEEC - ASInt
Laboratory 6
9/15 November 2016

Change the project from the last laboratory so that a command line clinet can also interact with the book database:



The new Client is a python application (Just like the one done in the first laboratories that interacts with the library using a REST webservice

REST Web Service

Using the basic book library as a base, implement a REST service that exports all possible data (books and authors).
Defines the possible URI for each resource:

- List of books (e.g. /data/books)
- List of authors
- Single author
- Single books
- List of books of an author

Using the regular python datatypes define the representation of each resource, take into consideration the datatypes returned by the library class

If the functions return python data types, bottle automatically transforms them into jsin and sends it to the client.
When accessing a certain resource verify if it exists and return the correct HTTP error code:
http://bottlepy.org/docs/dev/tutorial.html#http-errors-and-redirects

REST client

In order to call the REST webservices from python the **urllib/json** libraries can be used:
GET

```
import json,urllib2
data = urllib2.urlopen(URI).read()
d = json.loads(data)
print (d)
```

POST

```
import json,urllib2
data = {'ids': [12, 3, 4, 5, 6] }
req = urllib2.Request(URI)
req.add_header('Content-Type', 'application/json')
response = urllib2.urlopen(req, json.dumps(data))
```

HTTP error codes are caught with the exceptions **urllib2.HTTPError** :
https://docs.python.org/2/library/urllib2.html#urllib2.HTTPError
You can also install a more user-friendly library: **requests** (http://docs.python-requests.org)

More information in:
https://realpython.com/blog/python/api-integration-in-python/
http://rest.elkstein.org/2008/02/using-rest-in-python.html
https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop