A Glimpse of BOSS

- BOSS: Build your Own Search Service: Yahoo's open search web services platform: http://developer.yahoo.com/search/boss/

- BOSS API: http://developer.yahoo.com/search/boss/boss_guide/

- Requires a Yahoo BOSS APPID to identify your project.

- Example Query:
http://boss.yahooapis.com/ysearch/web/v1/Python+New+Hampshire?appid=a...z&filter=-porn,-hate&type=html,text,pdf

- Parameters:
  start: Where to start seeing results. First position is 0 (for paging)
  count: Total number of results to return. Maximum is 50.
  lang: Defaults to en.
  region: Defaults to us
  format: xml or json. Defaults to json.
  callback. JSON callback function.
  sites: Restrict results to these sites.

- Filters (human-language specific):
  -porn
  -hate

- Document types to return:
  html
  text
  pdf
  xl (xls)
  msword
  ppt
  msoffice (xl, msword, ppt)
  nonhtml (text, pdf, msword, ppt)

- Returns list of: count (# results per page), start (first result on page), totalhits, deephits (may have duplicates), abstract, title, url, clickurl, dispurl, size, date.

So where's the Python?

BOSS Mashup Framework:
http://developer.yahoo.com/search/boss/mashup.html

"...experimental Python library that provides developers with SQL-like constructs for mashing up the BOSS API with third-party data sources.

FEATURES:
(From the Yahoo website)
Current features of the Mashup Framework include:
- Support for SQL-like functions such as select, group (reduce), sort, union, inner join and user defined functions (map)
- Text normalization and duplicate removal
- Auto-transformation of resource-oriented API results into tables without specifying any parsing logic (unifies XML and JSON responses based on inferred data format)
- All-in-memory storage and retrieval operations
- Ability to join lists of tables via an arbitrary predicate function
- Mashup output available in XML or JSON
- Includes UI templates to allow developers to easily render mashed up search results on a customizable search results page template

Let's look at Yahoo examples.
(all examples by Vik Singh of Yahoo, Inc.)

ex3.py
Search 'aptana' on yahoo news, summize, and digg
Join results based on titles having an overlap of 3 terms or more
Group duplicates based on yahoo news title
In the group by, sum by diggs, save as field rank
Then sort by rank and print to stdout

ex4.py
Four way of join of 'google android' on yahoo news, summize, youtube, and digg
Combine results based on titles having an overlap of 3 terms or more
Group results based on yahoo news title (remove duplicates)
Redefined the group by equality operator to use text.norm to do near duplicate text removal
In the group sum the digg and youtube favorite counts as the rank for each joined result
Sort by rank, print to stdout

Presented to New Hampshire PySIG by
Ray Cote, rgacote@AppropriateSolutions.com
28 August, 2008