

Web Services

Louise Moser
Firat Kart
Winter 2007



SOAP

- SOAP stands for **Simple Object Access Protocol**
- SOAP is a **communication protocol**
- SOAP is for communication **between applications**
- SOAP is a format for **sending messages**
- SOAP is designed to communicate over the **Internet**
- SOAP is **platform independent**
- SOAP is **language independent**
- SOAP is **based on XML**
- SOAP is **simple and extensible**
- SOAP allows you to **get around firewalls**



The logo for Yahoo! Developer Network, featuring the word "YAHOO!" in red and "DEVELOPER NETWORK" in black.

- Yahoo! Search Web Services - Local Search
 - <http://developer.yahoo.net/search/local/V3/localSearch.html>
 - Search the Internet for businesses near a specified location
 - Returns both the latitude and longitude and Yahoo! user ratings of the establishment, as well as search by business categories

Request Formatting

- Request URL
 - <http://api.local.yahoo.com/LocalSearchService/V3/localSearch?>
- All search request URLs start with the hostname and resemble the following sample
 - <http://api.search.yahoo.com>
- After the hostname are the service name and version number:
 - [LocalSearchService/V3](#)
- Next is the method followed by a question mark
 - [localSearch?](#)
- Those components form the base URL
 - <http://api.local.yahoo.com/LocalSearchService/V3/localSearch?>

Request Formatting (cont)

- The method is followed by the actual query parameters, which take the form **argument=value**, where the arguments and values are **url encoded**
- Multiple parameters are separated by an ampersand (&)
- The following example searches the local database for locations related to "Pharmacy"
- `http://api.search.yahoo.com/WebSearchService/V1/webSearch?appid=W06ECE155B&query=Pharmacy&sort=distance`
 - You will need your own *appid*, apply for one
 - Register an application ID from
 - http://api.search.yahoo.com/webservices/register_application

Parameters You Can Send

- **appid** (String, required)
 - The application ID
- **query** (String)
 - The query issued
- **results** (integer)
 - Number of results to return
- **start** (integer)
 - Starting index of results
- **sort** (string, default relevance – title, distance or rating)
 - Sort the result set according to the chosen criteria
- **radius** (float)
 - How far (in miles) from the specified location
- **street** (String)
- **city** (String)
- **state** (String)
- **zip** (integer or integer-integer)
- More available at web site
 - <http://developer.yahoo.net/search/local/V3/localSearch.html>

Response Message

```
<Result id="21169080">
  <Title>US Post Office</Title>
  <Address>200 Pacific Oaks Rd</Address>
  <City>Goleta</City>
  <State>CA</State>
  <Phone/>
  <Latitude>34.429387</Latitude>
  <Longitude>-119.877504</Longitude>
  <Rating>
    <AverageRating>NaN</AverageRating>
    <TotalRatings>0</TotalRatings>
    <TotalReviews>0</TotalReviews>
  </Rating>
  <Distance>5.22</Distance>
</Result>
```

- Parse for this info (bold items)
 - Use result ID as primary key

How Do You Send HTTP Requests?

```
String BaseAddress =
    "http://api.local.yahoo.com/LocalSearchService/V3/localSearch";
```

```
// Submit request
```

```
URL url = new URL(BaseAddress+
    "?appid=W06ECE155B&query="+
    URLEncoder.encode("Pharmacy") + // In case you need to
    "&zip=93117&sort=distance");
URLConnection connection = url.openConnection();
```

- You may need to decode request parameters, use **URLEncoder.encode** method where you need to (in case you have spaces, or special characters)

Reading response

```
// Read response
```

```
BufferedReader br = new BufferedReader(new  
    InputStreamReader(connection.getInputStream()));
```

```
String xmlResponse = "", line;  
while ((line = br.readLine ()) != null)  
    xmlResponse += line;
```

```
br.close();
```

