

Introduction

What is a RSS?

RSS ("Really Simple Syndication") - is a family of Web feed formats used to publish frequently updated content such as blog entries, news headlines, and podcasts in a standardized format. An RSS document (which is called a "feed," "web feed," or "channel") contains either a summary of content from an associated web site or the full text. RSS makes it possible for people to keep up with web sites in an automated manner that can be piped into special programs or filtered displays.

Discussion

Our mini-project's purpose is to allow "SAYA" to be more informative than now, she will be able to know by simply "reading" information from the web about a variety of issues (such as: "news", "sports", "new technology" etc.).

As for the moment "SAYA" can response only to a few questions which she can provide an answer to and this addition will allow her to be much more informative than at this present time.

Our main discussion circled around the need for such a feature for "SAYA". Why does she need to able to supply such information as a "receptionist".

In our opinion, by implementing this very extendable feature, we will allow "SAYA" to be more informative and useful as a full featured interface to the internet. By allowing her to connect to RSS feeds, and the fact that almost every respected source of information on the internet has an RSS feed, this will basically allow "SAYA" to function as a source of dynamic information to the Computer Science department's visitors.

In a very likely scenario, a visitor to the department's lobby would come up to "SAYA" and ask "news sports" and get the recent winners of gold medals in the current Olympic games. (Probably just Michael Phelps..)

Our conclusion was that this feature will improve "SAYA"s' ability to interact with the people around her and provide her with a greater variety of replies to questions asked and to able to converse with the surrounding people.

Implementation

Our mini-project allows the "SAYA" robot to connect to the internet and download RSS feeds upon recognizing pre-defined key words (i.e. "news", "sports", finance" etc.).

Upon recognition of one of the key words she will connect to one of the pre-defined RSS feeds corresponding to that word.

The connection to the proper RSS feed is done with a HTTP Client and then downloading the RSS feed which is in XML format. Our program supports both standards RSS 1.0, RSS 2.0 and Atom.

Once we have the XML file in hand, we are required to fetch only the relevant information for "SAYA" to say, our goal is to supply to the person requesting the information the output that will be the most relevant and informative. We chose to supply the title and description of the N (default N = 5) most recent items from that feed.

The fetching of said information requires parsing the XML, to allow "SAYA" to interpret the message. We decided to program our own parsing mechanism, since various models and packages we found online provided too much functionality. The parsing phase requires "stripping" down the XML and leaving the title and description as the text returned to "SAYA".

Once we crop the relevant information we are still left with an incorrect output since RSS feeds allow the insertion of HTML into title and description fields of the feed. During our work on this mini project we gained some experience with using Java's support of regular expressions in the String class to pass the output through another stage of filtering out HTML tags, using String's "replaceAll" method we remove the HTML and CDATA meta tag that indicates HTML is present.

Another stage in the parsing is to check whether or not the text contains HTML format as the text of the XML element and if so then to strip the HTML as well (using a regular expression and Strings' method of "replaceAll").

Now the information is held in an instance of a class named RssItem, changing this class's toString method can allow future projects to change the way "SAYA" outputs the information, currently each item is spoken as "Item: [item text], description: [description text]", while this isn't the best way to say things, it's outputted this way just to show that the output corresponds to the toString method.

The final phase of the process is the returning the constructed output string out of the N chosen last items, this N is set as a final int in the main class RssReader, and can be supplied in the constructor.

"SAYA" will then speak this string to the person requesting the information.

The Use Of Dapper™:

The URL of one of the RSS feeds is a "Dapp". We used a feature provided by Dapper (an Israeli based web 2.0 startup) to supply "SAYA" with information from a website that doesn't have an RSS feed.

Basically Dapper's feature allows users to create personal feeds off any site by choosing fields off that site's HTML and Dapper constructs the RSS feed and updates it whenever the site updates.

More information on dapper can be found at <http://www.dapper.net> and a video demo of the feature is located at <http://www.dapper.net/dapperDemo/> .

We used dapper to show that "SAYA"'s new found functionality can be further extended to support information retrieval from the internet without requiring each site to have an RSS feed, though each "Dapp" still

has to be created manually using Dapper's website.

The URL supplied by dapper for the dapp constructed for the Bloomberg currency page was:

[http://www.dapper.net/transform.php?dappName=WorldCurrencies&transformer=RSS&extraArg_title=FromTo&extraArg_description\[\]=Value&applyToUrl=http%3A%2F%2Fwww.bloomberg.com%2Fmarkets%2Fcurrencies%2Feurafr_currencies.html](http://www.dapper.net/transform.php?dappName=WorldCurrencies&transformer=RSS&extraArg_title=FromTo&extraArg_description[]=Value&applyToUrl=http%3A%2F%2Fwww.bloomberg.com%2Fmarkets%2Fcurrencies%2Feurafr_currencies.html)

and it also defined in the submitted input map file (rssMap.txt)

Future Add-on possibilities

1. One could construct an online xml that would contain the names of the "Alon building" employees and their corresponding office numbers, then "SAYA" would fetch that xml and use XQuery to extract the relevant data corresponding to the request of the person in front of her.
2. future projects can extend our feature by having her recognize free speech, search for a given word in Google, use the "I'm feeling lucky" feature to find a website, and parse that website's html for the corresponding RSS feed, then giving the person the feature to add that feed to her feed map.

This will make her more dynamic and interactive than today.

User Guide and API

Format of input file:

The constructor to the RssReader class should receive the name of an input file, to construct the map from.

The input file consists of lines, each line has two elements, the first being the key string then a '|' character (without a space after the keyword, to allow more than one word to be the key string.) then the URL of an RSS feed corresponding to that key string.

For example:

news|http://www.ynet.co.il/Integration/StoryRss3254.xml

gadgets|http://www.engadget.com/rss.xml

iphone|http://topiphonenews.com/feed/

sports|http://sports.yahoo.com/top/rss.xml

API Usage:

- Constructors:

RssReader(String filename, int n) – the parameter filename is the name of the map file, the parameter n – states the number of items to display.

RssReader(String filename) – same as the above constructor with a default value n=5.

- Methods:

public String getRss(String key) – this method retrieves the RSS feed from the designated location and returns a parsed String to "SAYA" which she should say as a reply to the person who requested the key.

public boolean keywordExists(String key) – once "SAYA" is spoken to, she should be able to know if the person wants information from a RSS feed that corresponds to the spoken words. This method should be called to know whether or not to direct a person's request to our Object or to continue with the main loop.

Execution Examples

Input: "dollar"

Output:

Title: EUR-USD
Description: 1.4706.
Title: GBP-USD
Description: 1.8629.
Title: USD-CHF
Description: 1.0956.
Title: USD-SEK
Description: 6.3824.
Title: USD-DKK
Description: 5.0720.
.
.
.
.
Description: 7.8262.
Title: USD-MAD
Description: 7.7238.
Title: USD-ZWD
Description: 19.2400.

N Items

Input "news"

Output:

Title: PM: IDF's gloves to come off if Lebanon turns into Hizbullahland
Description: Olmert says Israel avoided using heavy weaponry in Second Lebanon War because it fought against a terror group, not a country..
Title: Encountering Peace: What Olmert has to complete before exiting
Description: Every time I read in the papers that negotiations over Schalit have been frozen, I almost laugh..
Title: A woman's work is never done
Description: Following a rape of a woman by her repairman, TA rape center compiled a list of repair women..
Title: Russia: Israel gave Georgia arms
Description: Russian deputy chief of staff: "Israel supplied weapons, unmanned aerial drones and explosives."
Title: 'PM won't fire Barak despite budget row'
Description: Olmert aides: Barak will remain in his position even if Labor ministers oppose budget proposal..

Bibliography

- Wikipedia - [http://en.wikipedia.org/wiki/RSS_\(file_format\)](http://en.wikipedia.org/wiki/RSS_(file_format)).
- Sun Tutorial – <http://java.sun.com/docs/books/tutorial/essential/regex/>