

RSSence Technical Reference Manual

The Palmetto Grizzlies

Gabriel Fair and Matthew Stratton

Table Of Contents:

External Web Service	2
• getCatalog()	2
• rssence()	2
• getFeedURL()	2
Android App RSSence	3
• MainMenuActivity	3
• FeedActivity	4
• ShowWeb	5
• DataSelectActivity	5
• ShowMap	6
• DatabaseManager	7
• Rss Classes	7
• XMLParser	7

External web service: webservice.php

The primary function of the web service is to parse and direct datasets to the end user's mobile device. The two most important methods are `getCatalog()` and `rssence()`.

getCatalog():

This returns the list of RSS feeds and datasets available in our database that the user can select and view. The first thing the app does when it starts, besides load a splash screen, is request the catalog from the server. The server then returns an RSS (xml) file with the contents of the Catalog table. Currently this is done in real-time; in the future this would ideally be done as soon as the catalog is updated to minimize response time.

rssence():

This is the essence of our app, at least server side. Its job is to parse a specified table from the MySQL database into an xml file, in a format our app can understand. First it constructs a sqlite command the app can use to create the table and puts it in the "description" field of the xml file. Then it creates a separate `<item>` for each entry in the table, populating it with the appropriate data. With the data parsed and formatted, the server returns the xml file dataset to the mobile device. Again, this should ideally be done as soon as the data arrives and is put into the database.

getFeedURL():

This returns the URL of a specified feed to the mobile device. The end user will select a feed name he/she is interested in from a list, the app then requests the URL of the feed with that name.

Android Application: RSSence

From the user's point of view the app is split into a number of activities. The activities themselves are mostly listener functions to allow the app to respond to the users input (what happens when a button is clicked, how to populate drop down boxes, etc.). There are also several Java support classes used by these activities.

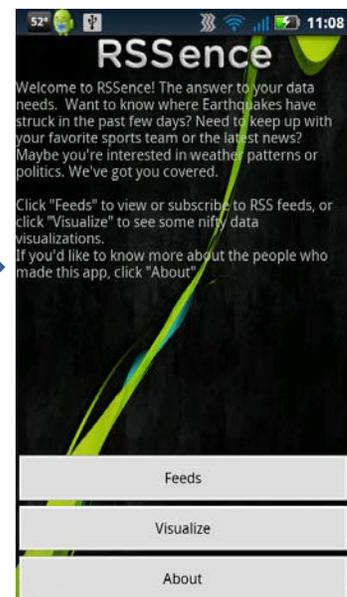
MainMenuActivity:

This is the main menu, the first thing the user sees. When the app starts this is the activity that is launched.



First the activity greets the user with a splash screen while it checks to see if the catalog is out of date and if so downloads the new catalog from our web service (see DatabaseManager). Once the catalog is verified or downloaded, the actual main menu is shown.

From here the user can select an action, either Feeds, Visualize or About which launch the FeedActivity, DataSelectActivity or AboutActivity, respectively.



FeedActivity:

This activity allows a user to select an RSS feed from a list and view the feed contents, generally news articles. New RSS feed links can also be added with the 'Add New Subscription' menu button.

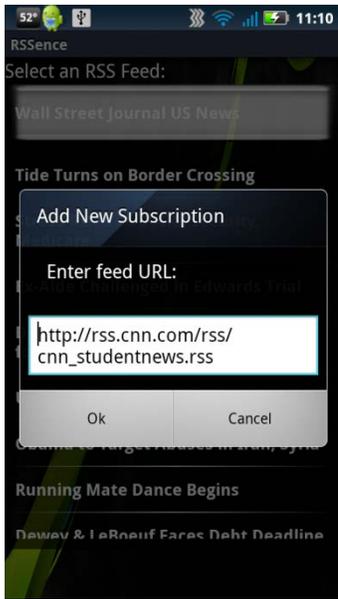


At the top are 2 spinners, or dropdown boxes. One allows the user to select categories, the other a specific feed from that category. The category spinner is populated when the activity is loaded, the feed spinner is populated when a category is selected.

When the user selects a feed from the feed spinner, the feed is fetched and parsed. A listbox is populated with items from RSS feed, generally these are different articles the user may want to read

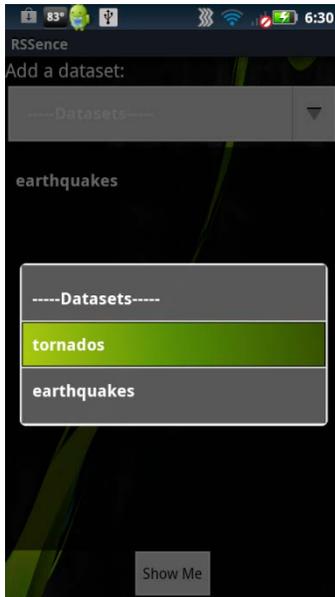
When the user presses the menu button, the 'Add New Subscription' option appears. The user can now enter a personal RSS Feed URL they wish to view that is not a part of our database

This dialog box is displayed when the user selects 'Add New Subscription' When OK is selected the string is checked to ensure not only that it is a valid URL, but that it is actually an RSS Feed.



DataSelectActivity:

This activity allows a user to select one or more datasets from a list to be visualized.

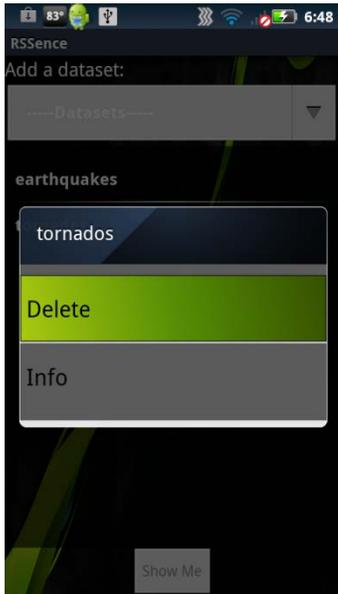


Like Feed Activity, the user selects datasets to be viewed from the spinner. When a dataset is selected it is added to the listbox. As soon as the dataset is selected, it is downloaded in the background from our server



The Show Me button will send the selected datasets to ShowMap to be visualized

Context Menu. When the user long presses a dataset in the list, a context menu appears with options to delete or view information on the dataset.



ShowWeb:

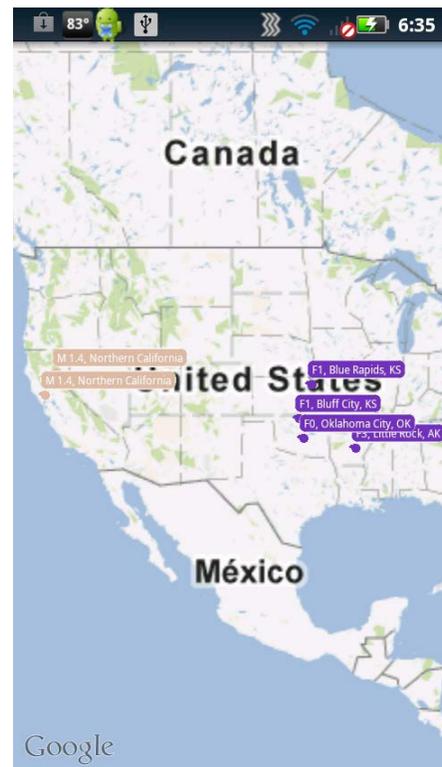
When a feed item (i.e. news article) is selected in the FeedActivity, the URL is fetched from the external server, and that URL is sent to the ShowWeb class.



The ShowWeb class essentially creates a web browser inside our app. It removes the URL bar so the user cannot navigate too freely, although they can still follow links.

ShowMap:

GeoData is plotted on a Google map, multiple datasets are shown with different colored dots. The points are pulled from the database with the getMapData() function in DatabaseManager



DatabaseManager:

The database manager has many important functions. It handles all transactions involving either the internal database or external server. Anything from fetching the catalog to getting lists of information from the internal database goes through DatabaseManager.

Void `getFromCatalog(String type)` – returns a subset of the Catalog, all entries of a given type. Used to populate the FeedActivity and DataSelectActivity spinners

Void `FetchCatalog()` – retrieves the catalog from the external server and stores it in the internal database, called while showing the splash screen

`ArrayList<String> getSubscriptions()` – returns a list of the users local RSS feed subscriptions

`String getFeedURL(String feed_name)` – retrieves a feed URL for the specified feed from the external server

`ArrayList<MapData> getMapData(String dataset)` – returns a list of MapData from a requested dataset, used in ShowMap. MapData is a very simple (like 8 lines of code) container used to store the GeoPoint and relative information about a point like name and description.

There are a few more functions in DatabaseManager, but they are mostly just support methods for the methods described above.

RSS Classes:

RssFeed, RssHandler, RssItem, RssReader.

These are data structures, classes and methods used to parse RSS Feeds. RssReader has a single function that accepts a URL and returns an RssFeed object. To parse a given feed, simply pass the URL to the `RssReader.read()` function. The RssFeed object is mostly getter functions to get various information from the feed: title, description, link, RssItems, etc. RssItem contains information about a particular item in the feed, so a news article will be 1 item in the RSS feed.

XMLParser:

The XMLParser is used to parse datasets received from the external server. The xml file contains the command needed to create the internal sqlite table associated with the data, as well as information about data types contained in the data (integer, float, text, etc). The parser reads through the xml, creates a table for the data in our internal database, and populates that table with the data contained in the file.