Buddy App Documentation

Duy Nguyen Feng Ji Moussa Ehsan

July 27, 2012

1 Introduction

Charity Navigator is an independent, non-profit organization that evaluates American charities. Its stated goal is ”to advance a more efficient and responsive philanthropic marketplace by evaluating the financial health of America’s largest charities.”

Given the invaluable information provided by Charity Navigator and its pretty good search API, it’s still hard for the donators to choose a charity organization based on their interests. This is essentially because having a wide options to choose, it’s hard for the users to identify their interest for charity.

Buddy App targets exactly this need for the donators. As Facebook is a widely-used social networking website that somehow records all the users socio-behavioral interactions, it can be greatly used to identify users’ interests.

In this app which is deployed on cloudfoundry extracts the donator’s information from his/her facebook account, uses data mining to extract suitable keywords, which are later used to search in Charity Navigator. The user’s home town, current location, hobbies, liked pages, etc. are all used for extracting the keywords. The best charities in indexed in Charity Navigator is then listed to the user based on the extracted keywords.

2 Architecture Design

Buddy app relies on Spring framework which is an open source application framework for Java platform. Spring framework features its own request-based MVC web application framework which facilitates web developement for Java developers.

As depicted in 1, we have devised two views and controllers: Login and Home. Login view and its corresponding controller is used to communicate with Facebook security APIs to get the authenticate the users and authorize our app to extract his/her information. Home view and its controller’s task is to fetch the user’s information from Facebook, analize it to extract his/her keyword. It thens searches Charity Navigator using its search API, parses the HTML results and displays them to the user. Each part of this process involves some classes and external APIs which we describe in the following subsections.

2.1 FacebookLogin Class

Facebook class (FacebookLogin class) is called by the Login controller to authenticate the user. The class connects to Facebook Graph API and requests a login and authorization page. The returned page is displayed to the user so he/she can authenticate with facebook and authorize the application to access his/her account.

2.2 FBKeywordExtractor

FBKeywordExtractor is composed of two classes: FBDirector and Carrot2Director.

FBDirector is queries Facebook Graph API using a thirdparty package, RestFB. RestFB is a simple and flexible Java Facebook Graph API and Old REST API client. RestFB provides a client interface for Facebook Graph API to query users information. On the other side, Facebook Graph API supports a customized query language (FQL) to query user's information. The following information is fetched from Facebook:

- Music
- Interests
- Hometown location
- Current Location
- Education
- Sports
- Religion

Carrot2Director is a class to call Carrot2 functions to analyze the fetched data from Facebook. We mainly use the Lingo algorithm implemented by Carrot2 to extract good label as keywords. Carrot2 is an open source search results clustering engine. It can automatically cluster small collections of documents, e.g. search results or document abstracts, into thematic categories. The clustering results are then further analyzed to filter out less relavent keywords and increase the accuracy. The following decisions were made based on experiance(!):

- We decided to only accept clusters which their cluster label score was higher than 25/100(confidence more than 25%).
- We decided to only accept clusters in which the number of documents are more than 5% of the whole documents. (Support of 5%)

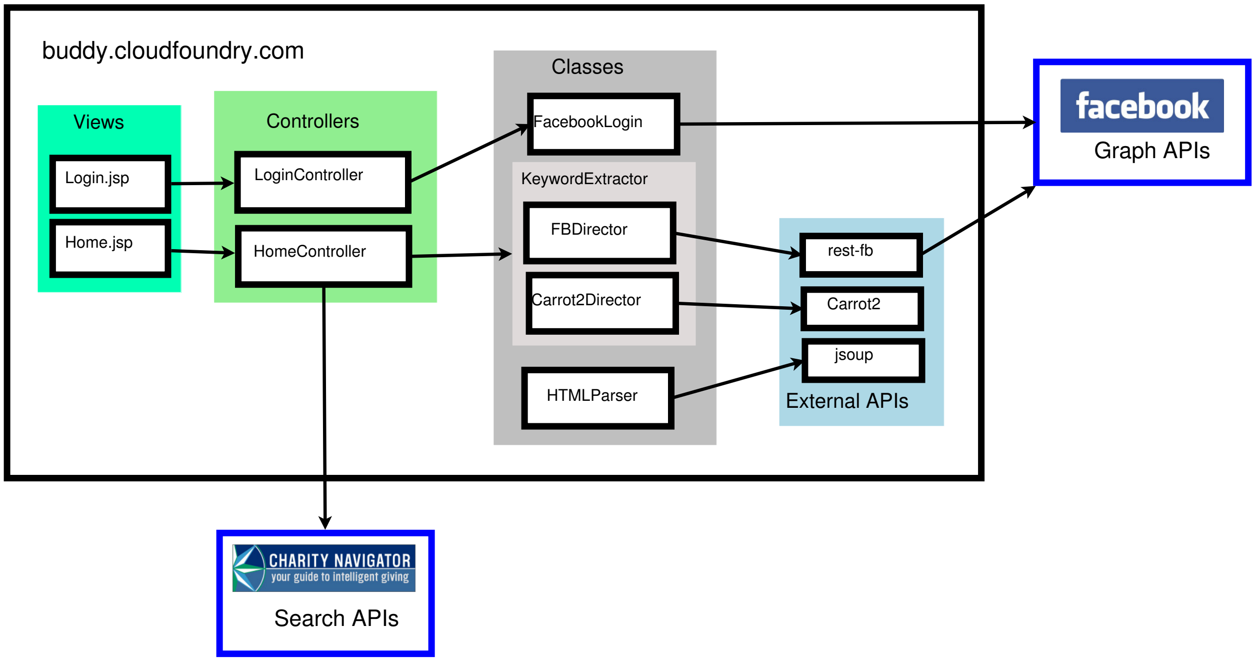

Figure 1: Architecture Overview

2.3 Charity Navigator Search API

To be completed by Feng.

The search result from Charity Navigator is later parsed by HTMLParser class to only extract the sections that the app needs to display. HTMLParser uses a thirdparty API called Jsoul. The open source API parses the given html and returns a DOM which is easily searchable.

3 The App Maintenance

The App is currently being maintained by Duy Nguyen(nguyend@vmware.com), Feng Ji(jif@vmware.com) and Moussa Ehsan(ehsanm@vmware.com). You can contact them for further information or to report any bugs.