Final Project Report

CSE 636 Data Integration

Yiming Cheng, Jun Wang - 2013.12.11

Screen For Me

Mashup Website of Movie Data and Social Network

Introduction

Background

Mashup used to be a term of music and now it is widely introduced to the field of web development, which refers to using content from more than one source to create a new result or service. The term implies easy, fast integration, frequently using open application programming interface(API) and datasource to produce enriched results that are needed for various of demands.In the past years, the growing number of Web applications APIs enable software developers to easily retrieve data and call functions instead of building their own.

Meanwhile, the concept of Web 2.0 has been widely accepted by the internet industry.Web 2.0 highlighted the interaction and collaboration between users and service providers. Examples of Web 2.0 include social network sites, blogs, video sharing sites. Users of these sites not only passively browser the content but also produce content. The transformation from consumers to producers has tremendously enriched the ecosystem of the internet, and made it more like the platform of communication.

Motivation

A mashup website perfectly goes with the tide of Web 2.0 and it's technically not complex to implement. However, 1 + 1 > 2 is the ultimate goal of mashup, which makes picking components from thousands online APIs the crucial first step.

After days of searching and thinking, we realize that people need to platform to share their feelings about the movies they've ever viewed while they need advice about lots of upcoming new movies. Most of the existing movie websites (e.g. rottentomatoes, imdb, etc.)are like libraries or databases of movie information. We'd like to add some social features to make our own mashup.

Problems to be addressed

1. There is no guarantee that the service you are implementing will be able to accommodate the traffic your website will get if it grows bigger.
2. APIs is convenient to use but not free enough. Existing limitation include restricted access and non-uniform data format. This is no doubt that further development will be hard.
3. Security is a issue.The most useful mashups will involve Web-based creations that are powered with our personal and business information. There is a lot of work to be done before the average user will trust a mashup with access to their private information. This shortcoming fundamentally limits the real value that mashups have the potential to provide.

Previous related work

Rotten Tomatoes API:

The API gives access to Rotten Tomatoes' wealth of movie information, allowing anyone to build applications and widgets enriched with Rotten Tomatoes data.

Facebook API:

The Facebook platform enables us to make our website more social with plugins that let us easily add social features to the website, and dialogs we can use to let users share content. By using Facebook Login to sign in users with their real identities, calling the Graph API and using Open Graph to tell stories, users will be more engaged and more likely to return.

Youtube Data API:

The Data API allows a program to perform many of the operations available on the YouTube website. It is possible to search for videos, retrieve standard feeds, and see related content. A program can also authenticate as a user to upload videos, modify user playlists, and more.

Technical contribution

Basic concepts and definitions

JSON

Or JavaScript Object Notation, is a lightweight data-interchange format. It is easy for humans to read and write. It is also easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

AJAX

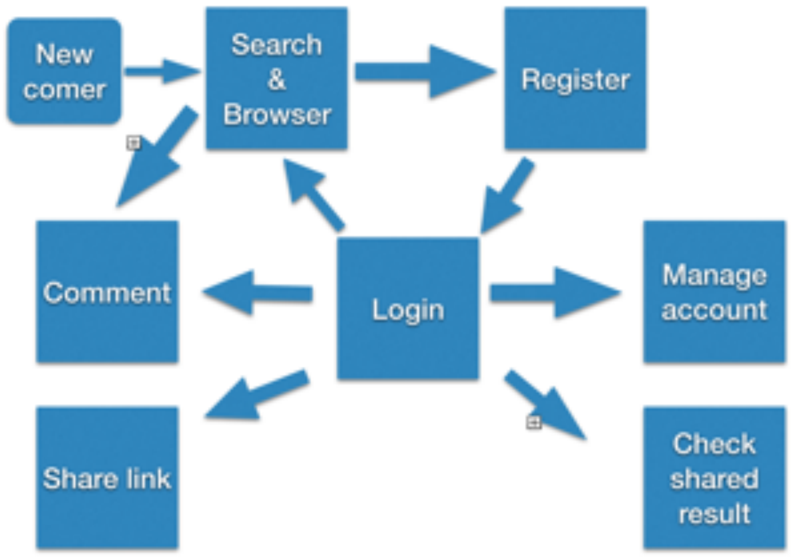
An acronym for Asynchronous JavaScript and XML. Ajax is not a single technology, but a group of technologies. HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display, and allow the user to interact with, the information presented. JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

HTML5

HTML5 will be the new standard for HTML. It aims to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices.HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

Ideas of design

Workflow



This figure shows the overview of workflow that a typical new user is likely to perform.

Dynamic Webpage

Our website will show different content base on the status of a visitor. Visitors stay on the same page during the whole experience without any distraction from popped up windows or new tab pages.

Different User Different Access

We have to decide the identity of a user and grant him certain level of access. From the perspective of a user, there are several scenarios:

1. A new user visit the page.
2. A registered user visit the page.
3. A curious user visit through a shared link.

The web content is different when any of these scenarios occurs. Meanwhile, different tables will be updated in local database as the result of the behaviors of different users.

Complementary Local Database

To implement some of the social features, we need a local database to help store necessary information, which is a complement of datasource.

Justification of the design choice

Single dynamic page

Although it is complicated to implement all the functions on one page, the dynamic feature of website indeed enhanced user experience. There is no popping up window, no frequently reloading, which present less distraction to the visitor. And the website is more instructive to users because the ui is always guiding people to do what they should do.

User access control

This is the fundamental requirement of a website that need users' personal information to provide service.

Local database

Our own database makes it more flexible to store necessary data for implementing new features. We can do more than display the information retrieved from outer sources.

User interface

Search and Result

Search and Result sections are two blank blocks that will be automatically filled up and cached if user triggers the corresponding scripts.

A specific result is an integrated output of different datasources. There will be four areas:

1. Movie information from rotten tomatoes

   Including the title, poster, year, synopsis, directer, cast, studio, ratings and top 3 reviews from a list of critics.

2. Movie clip from youtube

   Once the information of a certain movie is loaded, a video search with the movie title as the keyword will automatically executes. Search result is ordered by relevance and we only keep the most related one in this area.

3. Comments from local tables

   Including global comments and private comments.

4. Comment submit module

3 and 4 are both decided by the current identity of a visitor, the following form explains the mechanism：

FUNCTION REQUIREMENT	LOCAL LOGIN	EXTERNAL(FACEBOOK/TWITTER) LOGIN
BROWSER/SEARCH/COMMENT		
SHARE	NEED	NEED
CHECK COMMENT RECOMMENDATION	NEED	

Local user management system

Users can signed up with either email or username as identifier. Once signed up, the user panel of navigation bar will refresh and the page will jump to 'share history' block, which is loaded already.

Remote Facebook login

This is a function obtained from facebook php api.

Approaches

Dynamic page

With HTML5, we have set several anchors on index.php. A fixed navigation bar wrapped with semantic tag <nav> is also created on the top of the page to make sure people have the control on visiting. While the menu on <nav> is clicked, a href with #anchor will automatically take us to where the anchor is.

The detection of current user status

To achieve this, we have to send message from browser to server.There are two ways the browser client can send information to the web server.

1. The GET Method

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the ? character.

For example,a link

http://screenformecse636.com/?mn=movieid&user=uid

is constructed this way to send two parameters:"mn=movieid" and "user=uid" to server. This is the format of shared link.

2. The POST Method

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING.

The POST method does not have any restriction on data size to be sent.

The POST method can be used to send ASCII as well as binary data.

The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.

PHP provides $_POST associative array to access all the sent information using POST method.

For example:

<form method="post" action="index.php#signin" name="loginform" id="loginform">

The data of form "loginform" is sent to "signin" block,which is a php file by POST.

This the rules we followed in using POST and GET:

Rule 1: Use GET for safe actions and POST for unsafe actions.

Rule 2: Use POST when dealing with sensitive data(user information,comments).

Rule 3: Use POST when dealing with long requests.

Rule 4: Use GET in AJAX environments.

Modularization

The process of website development is module-oriented, which saves a lot time to debug and fix problems. Index.php loads the following modules under different conditions.

Login:	When user click sign in button or share a movie without signing in.
Register:	When user click register new button
Activation:	Once registration is submitted, an activation email will be sent to user's email
Reset password:	When user click forget password link, an password reset email will be sent.
Search:	A list of results will show up.
Movie:	The integration information of a movie will show up.

Comment:	Different comments will be made based on the identity of a user.
Facebook:	The implementation of posting Facebook message when user click share button.
History:	Once users click share button, a share history will be inserted to database for later query. And the history list is accessible to user that have signed in.

Local database

There are four tables maintained in local database.

Schema:

user table （id,username,password,actikey,email,status）

history table (username,movie,shared,positive,negative)

private comment table (friendname,time,username,movie,comment)

global_comment table (username,time,movie,comment)

‘user’ table stores information related to local accounts. It’s the basis to identify a visitor. ‘history’ table stores the tuples of shared movie, including the recommendation results. ‘private comment’ corresponds those comments made by visitors who come through a shared link. ‘global comment’ is the comment made by any visitor.

API Usage

Rotten Tomatoes API:

The Rotten Tomatoes API is RESTful web service that was designed to be easy to explore and use. The base URI to access all resources is http://api.rottentomatoes.com/api/public/v1.0. API accomplished by linking related resources and providing instructions on how to use each representation (link templates) in the response itself.

The structure of Rotten Tomatoes APIs:

- v1.0.json - The "homepage" of our API
- movies.json - Search for movies
- movie.json - info on an individual movie
- reviews.json - the reviews for an individual movie
- cast.json - the full cast for an individual movie
- similar.json - a list of movies similar to the individual movie

- clips.json - a list of clips related to an individual movie
- lists.json - overview of lists available
- movies.json - movies lists available
- box_office.json - a current list of box office movies
- upcoming.json - movies that are upcoming
- in_theaters.json - movies that are in theaters
- opening.json - movies that are opening this week
- dvds.json - dvd lists available
- top_rentals.json
- current_releases.json - dvds that are currently released
- new_releases.json - dvds that are getting released
- upcoming.json - dvds that are getting released

In our project, we use the base json file "v1.0.josn" to retrieve "movies.json" and "lists.json". Given the information of "movie.json", we can get the id and the details of the that movie, such as movie title, ratings, synopsis, posters, the reviews and cast for users. What's more, on the top of the homepage, users can see the movies that are upcoming and in theaters.

Facebook API

facebook.php API is used in the module of facebook login.

The action of posting links to user's wall is executed by javascript API. The link is generated in this format:

http://screenformecse636.com/?mn=movieid&user=uid

Youtube API

Javascript API is used to get JSON data for specific search keyword

Then a swf video player is embed with the url retrieved from JSON data.

The exploitation of JSON

JSON is built on two structures:

1. A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
2. An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In this case, we convert JSON data to javascript object, then output the specific node into the existing DOM(Document Object Model).

Challenges and solutions

The hardest part is the design that makes all the functions works on a single page. There were conflicts at first that makes the page a mess. Modularization helps saving a lot time of debugging and fixing in the develop process.

Another remarkable point is the set of local database. We have to design our own schemas to enrich the website, instead of simply fetching information from APIs.

Due to some security setting or policy of those provider, many APIs require the website to be online so that they can work properly. This is figured out by setting a vhost in apache conf.

Conclusion

Summary

Screenforme is not simply a mashup of remote APIs that displays information retrieved from datasources. It is also a highly integrated website with local social network service. People can actually interact with our database and produce web content.

Comparison with similar websites

Compare with rottentomatoes

Rottentomatoes has sharing and rating functions and it's a comprehensive movie database. However, it is not offering feedback from friends now. People can only see his friends' reviews and rating. We have add this feature in our project.

Compare with imdb

Although imdb is a powerful library of movie data, it's social network is not as good as its comprehensiveness. imdb can only share one page on the facebook post wall without commenting and inviting functions.

Something else new

Although HTML5 has not completely replaced current standards, it remain to be the future trend. In the development of our project, new elements, such as <section>, <article>, <header> and <nav>, are used to enrich the semantic content of documents. New attributes have been introduced for the same purpose, while old elements and attributes have been removed. This is an attempt of the new technology.

Limitations

1. The relying on information retrieved from APIs means we have no control on the data access. All the functions of the websites are based on the connection with the data source. If APIs are changed or gone, the workload of fixing is beyond imagination.
2. Although there won't be any security issue regards to Facebook account, the massive using of browser script makes the website vulnerable. There might be some loopholes that attacks could use to modify local database.

Future work

This website could be further developed if the following features were implemented:

1. A local database that stores data retrieved from api. Every time an user make a query, the retrieved information will be merged and display.
2. More social features could be added: adding friends, private message, notification system.
3. Advanced search: user can search specific movies using more customized conditions.
4. More secured user management system.

Bibliography

[1] Hanson, J.Jeffrey. Mashups: Strategies for the Modern Enterprise. Pearson Education, 2009.

[2] Chow, Shu-Wai. PHP Web 2.0 Mashup Projects. Packt Publishing Ltd, 2007.

[3] Brown, Steven Tracy. Dynamic Apache with Ajax and JSON. O'Reilly Media, 2009.

[4] Curioso, G Andrew. Apache with PHP5. O'Reilly Media, 2007.

[5] Graham, Wayne. Facebook API Developers Guide. Apress, 2008.