

On Simplifying Mobile Mashup Application Development

Guan-Zhong Wu, Chen-Che Huang, Yui-Chey Teng, Jiun-Long Huang, Wei-Chih Ting, Yi-Yu Su and Pin-Chuan Liu
 Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, ROC
 Industrial Technology Research Institute, Hsinchu, Taiwan, ROC

Abstract—In this paper, we propose a novel integrated API service system as well as a Web API script to enable mobile mashup application developers to invoke Web APIs in a consistent way, and provide a consistent format for the responses of Web APIs, thereby relieving developers of striving to study the usage of various Web APIs.

Keywords: Mobile Mashup Applications, Web APIs, Smartphones.

I. INTRODUCTION

With the explosive growth of Web APIs and ubiquity of smartphones, the number of mobile mashup applications has surged in recent years [1]. However, there are some obstacles faced by the developers to study a large number of new APIs. After observing the Web APIs released by social networking Web sites (Facebook and Twitter), Web album Web sites (Picasa and Flickr), video sharing Web sites (YouTube and Flickr) and search engines (Google and Bing). We find that when developing mobile mashup applications, developers will face the following challenges.

- *Challenge 1: Diverse invocations of Web APIs*
 The invocations of Web APIs differ from one another since many famous Web sites publish their Web APIs in their own way. Such diversity makes developers strive to learn the usage of these Web APIs and it is time-consuming when creating a new mobile mashup application. For example, the Web API to search YouTube for the videos about “Olympic” is shown in Figure 1a while the Web API to search Flickr is as shown in Figure 1b. We can observe from Figure 1a and Figure 1b that the invocations of the search APIs of YouTube and Flickr are quite different.
- *Challenge 2: Diverse data formats of Web API responses*
 Data formats of the responses of Web APIs provided by different Web sites are also diverse, since there is no standard in the Web API responses. When developing a

```
http://gdata.youtube.com/feeds/api/videos?
v=2&q=Olympic&orderby=published&max-results=1
```

(a) Example of YouTube Search API

```
http://api.flickr.com/services/rest/?
method=flickr.photos.search&text=Olympic&
per_page=1&api_key=20cbedf2c444b8eb57e335581ed13a&
media=videos&extras=views&sort=date-posted-desc
```

(b) Example of Flickr Search API

Fig. 1. Video Search Web APIs

mobile mashup application using the Web APIs of various Web sites, the developers should first understand the data formats of the Web API responses and then implement several parsers to extract the information of interest from the Web API responses. For example, the number of views in the YouTube API response is stored in the field “viewCount” while the number of views in the Flickr API response is stored in the field “view.” Thus, the diverse of the data formats will bring burden to developers, thereby decreasing the performance of developers.

To overcome these challenges, we in this paper aim to simplify the development of mobile mashup applications by

- enabling developers to invoke Web APIs in a consistent way, and
- providing a consistent format for the responses of Web APIs.

II. PROPOSED APPROACH

To achieve the above goals, we propose a novel integrated API service system (abbreviated as the IAS system) as well as a Web API script so that the developers are able to invoke Web APIs without understanding the details of the Web APIs provided by various Web sites.

A. Web API Script

To simplify the invocations of Web APIs, we propose the Web API script so that the mobile mashup applications are able to use the Web API script to invoke Web APIs instead of using HTTP requests/responses to directly invoke these Web APIs. Here, we focus on the Web APIs of the top four most popular categories: Social, Video, Photo and Search [1]. The format of the Web API script is shown in Figure 2, and the descriptions of these parameters are as below.

- *category parameter:* The *category* parameter is used to specify the type of Web APIs (i.e., Social, Video, Photo or Search) that are invoked by the Web API script.
- *keyword parameter:* The *keyword* parameter is used to specify the query string of the Web APIs.
- *site parameter:* The *site* parameter is used to list the Web sites that the Web APIs invoked by the Web API script belong to.
- *orderby parameter:* The *orderby* parameter is used to specify how to order the search results.
- *maxresult parameter:* The *maxresult* parameter is used to specify the maximal number of search results obtained from each Web site.

```
category [keyword] site:site1[,site2...] orderby:[attribute]
[maxresult=n] field=field1[, field2,...]
```

(a) Web API Script Format

```
Video "Olympic" site:YouTube,Flickr orderby:published
maxresult=1 field:id,title,viewCount
```

(b) Web API Script Search Example

Fig. 2. Web API Script

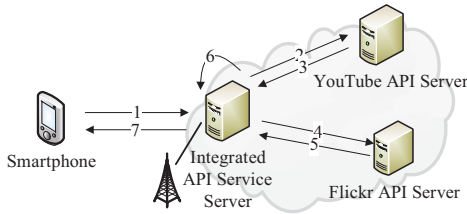


Fig. 3. Behavior of the Integrated API Service System.

- *field parameter*: The applications are able to express the fields in which they are interested by the *field* parameter. In addition, with the aid of the *field* parameter, the applications are also able to reduce the transfer size by asking the IAS system to eliminate unnecessary information in the responses of Web APIs [2].

We use the following example to show the usage of the proposed Web API script. Suppose a user would like to search YouTube and Flickr for the videos with keyword “Olympic.” The user would like to retrieve only one video from YouTube and Flickr, and the search result should be ordered by the publish time. For each video, the user only wants to know the id, the title and the view count. Based on the above requirement, the mobile mashup application should create the Web API script shown in Figure 2b to the IAS system.

B. Integrated API Service System

The proposed IAS system is a proxy deployed on Internet. Figure 3 shows the procedure that a mobile mashup application uses the proposed Web API script and the IAS system to invoke the Web APIs provided by YouTube and Flickr. Due to the space limitation, we use the Web API script shown in Figure 2b to describe the behavior of the IAS system.

- *Step 1*: The mobile mashup application sends the Web API script shown in Figure 2b to the IAS system.
- *Step 2*: After receiving the Web API script, the IAS system invokes the corresponding YouTube API (as shown in Figure 1a) to search for the videos with keyword “Olympic.”
- *Step 3*: YouTube API server returns the search result.
- *Step 4*: The IAS system invokes the corresponding Flickr API (as shown in Figure 1b) to search for the videos with keyword “Olympic.”
- *Step 5*: Flickr API server returns the search result.
- *Step 6*: The IAS system parses the search results of YouTube and Flickr (in XML format) to extract the desired fields specified in the Web API script, and then

```
{
  "count": "2",
  "data": {
    "YouTubeAPI": [
      {
        "id": "tag:youtube.com,2008:video:-rlxR63o0h0",
        "title": "Data journalism in action- the London Olympics",
        "viewCount": 3
      }
    ]
    "FlickrAPI": [
      {
        "id": "8603310092",
        "title": "Olympic Training",
        "viewCount": 5
      }
    ]
  }
}
```

Fig. 4. The Unified Search Result in JSON Format

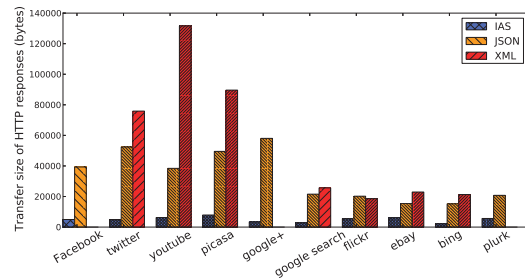


Fig. 5. Filter HTTP response from popular Web service.

organizes the desired fields into a unified search result (as shown in Figure 4). Since JSON format is more concise than XML format [2], the IAS system stores the unified search result in JSON format.

- *Step 7*: The IAS system returns the unified search result to the mobile mashup application.

III. PERFORMANCE EVALUATION

We employ *transfer size*, which is defined as the sum of sizes of the packets sent from and received by the smartphone, as the performance metric to measure the effect of the proposed IAS system and Web API script in transfer size. In Figure 5, “XML” and “JSON” indicate the transfer sizes of directly Web API invocations with responses in XML and JSON formats, respectively, while “IAS” indicates the transfer size of invoking Web APIs by the proposed IAS system and Web API script. Experimental result shows that, in addition to simplifying mobile mashup application development, the IAS system is also able to significantly reduce the transfer size resulting from Web API invocations.

REFERENCES

- [1] Programmable Web, <http://www.programmableweb.com/>.
- [2] C.-L. Tsai, H.-W. Chen, J.-L. Huang, and C.-L. Hu, “Transmission reduction between mobile phone applications and restful apis,” in *SAC*, 2011, pp. 445–450.