

Design and Implementation of Multimedia Social Services on Elgg

Meng-Yen Hsieh, Yin-Te Tsai, Ching-Hsien Hsu, Chao-Hung Hung,
and Kuan-Ching Li

Abstract This chapter designs and implements multimedia social services over an open social network engine, denoted as Elgg. Elgg with basic elements such as blogging, e-portfolios, and file sharing possesses a plug-in architecture, so that developers can design plug-ins to provide particular social services. The three plug-ins about video, photo, and blog with a number of Elgg's APIs based on the Representational State Transfer architectural style are provided in the study. Therefore, external applications can easily announce and manage data on Elgg through these APIs. Not only providing details on the design of the Elgg APIs, but we also describe their request and response formats in REST styles. The prototype services are validated with a simple client. The data including text and multimedia from external applications can be automatically separated into the blogs and albums on Elgg.

Keywords Elgg • RESTful • Social media

M.-Y. Hsieh (✉) • C.-H. Hung • K.-C. Li
Department of Computer Science and Information Engineering, Providence University,
Taichung 43301, Taiwan
e-mail: mengyen@pu.edu.tw

Y.-T. Tsai
Department of Computer Science and Communication Engineering,
Providence University, Taichung 43301, Taiwan

C.-H. Hsu
Department of Computer Science and Information Engineering, Chung Hua University,
Hsinchu 30012, Taiwan

1 Introduction

Currently, most people surf social media on Internet in order to maintain individual data, share multimedia data with others, and harness real-time social activities of their friends. The users who have already authenticated can often maintain and announce their multimedia data of daily life with sharing permission in social media. Some textual descriptions or tags for multimedia data must be given in order to depict user context. Consequently, to provide convenient management services for multimedia in social media is significant. Due to mobile and wireless technologies, users operate multimedia data of social media through not only WEB but also mobile APP. Therefore, this study designs internal web services for external applications to deliver user's multimedia data to social networks. An open social network engine, denoted as Elgg, is used, where we have developed a number of plug-in functions, denoted as plug-ins, for multimedia services and open RESTful APIs for external applications. The plug-ins also support linking mechanisms with external social media on Internet. The chapter is organized as follows: we describe literatures and technologies related to our contribution in Sect. 2. Section 3 introduces the design issues of the plug-ins over Elgg for multimedia social sharing. The RESTful APIs are also described. The implementation results are presented and discussed in Sect. 4, and finally, Sect. 5 concludes this chapter.

2 Related Work

The RESTful style based on service-oriented architecture (SOA) uses a uniform resource locator (URL) address to determine resource in Web servers. External applications perform HTTP methods through URL addresses to operate Web resources. In the past, we have developed a Web application and mobile APP about tourism [1, 2]. The application provides some open Web services, and therefore, the mobile APP can use the services to exchange data with the application. Garrett et al. [3] have presented a design project on Elgg for general educations. The customized Elgg has blogs and wikis to assist students in collaborative and individual learning. The project allows students and teachers to customize their profile and to increase interaction, better than traditional course management systems without user interaction. In the experience result, only 15 % of the students reported low levels of satisfaction with the Elgg. More than 60 % of people reported that the Elgg helped them about course experience and social relationship. Weng et al. [4] have designed a social learning environment on Elgg. The social learning relations are divided into the four categories, Object2Object, User2User, Direct relation, and Composed relation, while user and object are defined as the two main portions in the Elgg.

3 Elgg Plug-ins for Multimedia Service

This study designs RESTful APIs for multimedia Web services over Elgg. Using the APIs with URLs, external applications can directly communicate with multimedia services through HTTP protocols with the four methods, POST, GET, PUT, and DELETE, corresponding to the basic operations, create, read, update, and delete (CRUD) for remote resources in backend servers. Two multimedia plug-ins named as Photo Albums and My Videos are proposed for individual multimedia management. Figure 1 depicts our Elgg platform with the plug-ins and RESTful APIs for multimedia services.

3.1 Photo Albums and My Videos Plug-ins

After setting up the Photo Albums, users own a photo board on the Elgg. Users can configure the board, such as the limited number of albums and access permission. Four alternative types of access permissions are provided, public, friend, authentication, and private. The Photo Albums plug-in supplies the functions:

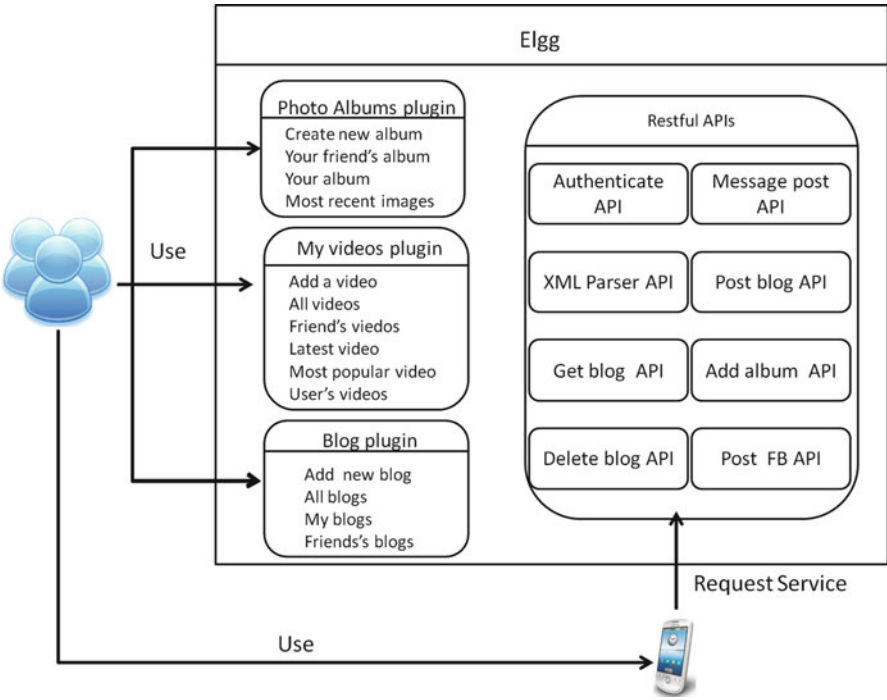


Fig. 1 Multimedia plug-ins and APIs on Elgg

Creating new albums. Users create a new album with initial descriptions including an album title, description, tags, and some configuration parameters. One of the configuration parameters is to set up access permissions on content in Photo Albums. Users can decide to upload images by one-by-one or batch mechanisms.

Browsing an album. Users easily open self or friend's album. Everyone who surfs the Elgg can browse the album, while the permission is set with the public privilege. Only friends open the album, while the owner configures the album with the friend privilege. Users that have been authenticated on Elgg are able to browse the albums with the authentication privilege.

Browsing the latest albums. Users browse the latest photos in the albums through a particular obvious link.

Maintaining an album. The authenticated users maintain their individual photos in the albums by the three functions, addition, edition, and deletion.

Like and sharing. Photos in an album are able to be shared to social networks, while users browse them and press the buttons linking to other social media. Users can press the "Like" button to show signs of liking some photos.

Downloading and commenting for photos. By this function, users comment on the photos that they saw in albums. Besides, the photos also can be downloaded easily.

The other plug-in, My Videos, assists users in announcing video data on Elgg. If one user steps up the plug-in in the Elgg's dashboard, his or her profile will present a window board for video presentation. The board can be set up with various presentation styles based on user preference. The plug-in includes the functions:

Uploading new videos. Owners upload video files to their video boards. The board embedded with a multimedia player includes simple functions, where users can play video. Each uploaded video is attached with one of the access permissions.

Browsing individual videos. The browsing function pattern presents that the users with different access permissions have different video files on their browsing pages. By linking functions, owners easily view the videos that they uploaded before. In addition, the videos with the private privilege are only presented for their owners.

Browsing all public videos. Another link denoted as "All videos," drives a browsing page to present all of the videos with the public privilege on Elgg.

Browsing friend's videos. Users can browse the videos in their friend boards, while the videos are set up with the friend privilege by their friends. A link denoted as "Friend's videos" drives the page listing with these videos.

Browsing latest videos. A user can browse the most new recent videos that other users announced with the public privilege recently, while pressing the link function denoted as "Latest videos."

Browsing most popular videos. A user can browse the most popular videos attaching with a high number of reviews. The page presents the popular videos in order based on how many times they have been seen.

Deleting a video. A valid user can delete his uploaded video when he is browsing it.

Table 1 Request and response data of the RESTful APIs

Services	Request format for service paths (URL)	Resp. (JSON)
Authentication	<code>/json/?method=user.login&username={name}&password={passwd}&api_key={key}</code>	Token
Posting	<code>/json/?method=travel.post&username={name}&title={title}&body={body}&api_key={key}&auth_token={token}</code>	Success or fail
Getting	<code>/json/?method=travel.get&guid={id}&api_key={key}&auth_token={token}</code>	ID and title or blog article
Updating	<code>json/?method=blog.delete&guid={id}&api_key={key}&auth_token={token}</code>	Success or fail

3.2 RESTful APIs

Our Elgg provides not only the plug-ins but also Web services for multimedia management. External applications can maintain multimedia data of the plug-ins by the CRUD methods of HTTP protocol. We address that multimedia data for external applications consists of one XML-based description and some corresponding images.

The APIs are released for the Web services as follows:

Authentication service. The API, named as “user.login,” supplies the authentication service. An external application must propose a login page to receive user’s account and password in order to deliver a HTTP request to the API. The request URL follows a format described in Table 1, and the authentication data are included as some parameters in the request. The request has to be protected by SSL during delivery. The API will return a token, after checking that the authentication data is valid. The token is used by the application to contact with the other APIs.

Multimedia posting service. The three APIs for this service, denoted as XML parser API, Post blog API, and Add album API, are used to deliver multimedia data to Elgg. The service has three steps for posting, while receiving the multimedia files. First, the service interprets the description data through the XML Parser API to form a blog data. Second, the Post blog API announces the data to the blog of Elgg and forwards the ID and title of the blog article to the Add album API. Finally, the API creates a novel album related to the blog’s ID and title and arranges the uploaded images into the album. The method for posting, named as “travel.post,” is applied into the service.

Multimedia getting service. The API, denoted as “travel.get,” assists external applications in searching and getting a blog article according to the article ID in the Elgg’s blog. The service also provides all of the IDs and titles of blog articles, while the external application requests the service and sets the ID parameter to the particular value, “all.” Generally, the external application shall get all of the IDs and titles for users, and they select an interesting article title to browse.

Multimedia updating service. The Elgg engine only supports HTTP get and post methods. We adopt the creation and deletion operations for the updating service. External applications update the same multimedia data according to the same identity of the multimedia description. While accepting the multimedia data with the same XML filename, the service deletes the old data by a method denoted as “blog.delete” and uses the Post blog API to create a novel announcement on the Elgg’s blog.

Other services. The proposed architecture provides other services such as Message Post API and Post FB API to assist external applications in announcing message data to the wall pages of Elgg and Facebook.

4 Implementation

This section introduces the implemented services on Elgg. Figure 2 describes a sample of external applications as a client. The client must obey the request formats of the RESTful APIs in order to assist users in managing multimedia data to our Elgg. The three alternative methods, travel_post, travel_get, and travel_delete, are presented corresponding to the multimedia services of the Elgg, after users have been authenticated. Users will be able to announce a XML file with three image files to the Elgg, if performing the travel_post method. Figure 3a gave the interfaces of the four plug-ins on Elgg, and Fig. 3b presented a blog article with the title, Taichung 1D, from the XML file that the client had uploaded before. We give a description for the interface as follows:

Photo Albums (no. 1). The plug-in lists all of the albums with the photos public on Elgg. User can create an album though the admin. interfaces of the plug-in or the external applications supporting with the RESTful APIs, described in Sect. 3. Users announce photos with some alternative privileges in the albums.



Fig. 2 Example of a simple client supporting the RESTful APIs

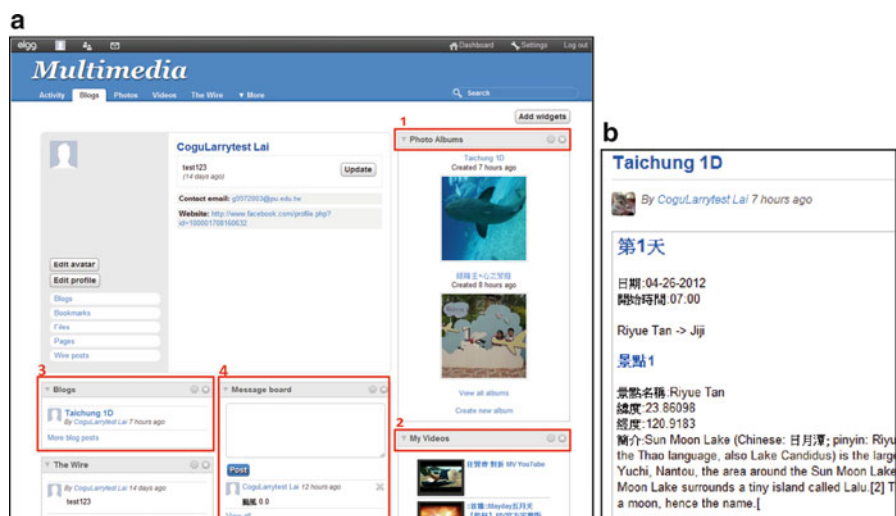


Fig. 3 Elgg for multimedia service: (a) the four plug-ins, blogs, message board, photo albums, and my videos; (b) a blog article for the multimedia description from external applications

My Videos (no. 2). All videos without classification are involved in the plug-in.

Users only manage videos through the admin. interfaces, after they have been authenticated on Elgg. Users announce videos with some alternative privileges.

Blogs (no. 3). The plug-in as one of the basic functions on original Elgg lists all of the blog articles. The client in Fig. 2 can deliver a XML description with few photos to Elgg in order to create a blog article from the XML and an album with the photos.

Message board (no. 4). The plug-in as one of the basic functions on original Elgg allows users to announce a short message on the board.

5 Conclusion

In this chapter, we develop two multimedia plug-ins and RESTful APIs on Elgg. The plug-ins service people to announce multimedia data in social networks. The RESTful APIs are proposed for that external applications can communicate multimedia data to the blogs and albums. The communication protocols are designed and acted based on the determined URL formats. In order to enhance social presence, the multimedia data on Elgg can also be linked with other social media.

Acknowledgments This investigation was supported by National Science Council, Taiwan, under grants NSC 100-2221-E-126-001-, NSC 101-2221-E-126-006-, and NSC 101-2221-E-126-002-.

References

1. Hsieh M-Y, Lin H-Y, Li K-C (2011) A web-based travel system using Mashup in the RESTful. *Int J Comput Sci Eng* 6(3):185–191
2. Hsieh M-Y, Lin H-Y, Yeh C-H, Li K-C, Wu BS (2012) A mobile application framework for rapid integration of ubiquitous web services. In: The 9th IEEE international conference on ubiquitous intelligence and computing, Fukuoka, 2012
3. Garrett N, Thoms B, Soffer M, Ryan T (2007) Extending the Elgg social networking system to enhance the campus conversation. In: Second annual design research in information systems (DESRIST), Pasadena, 2007, pp 14–15
4. Weng M, Chen Y-H, Yen N, Chang H-P, Shih T, Lin F (2012) A prototype of learning environment for social learning on Elgg social platform. In: IET international conference on frontier computing, Xining, 2012

Intelligent Technologies and Engineering Systems

Juang, J.; Huang, Y.-C. (Eds.)

2013, XXII, 1065 p. 650 illus., 19 illus. in color.,

Hardcover

ISBN: 978-1-4614-6746-5