The Design of RESTful Services for Indoor Location Based Service

Jaegeol Yim

Department of Computer Engineering
Dongguk University at Gyeongju, Korea
yim@dongguk.ac.kr

Abstract. Location based services (LBS) are so useful that they should be available inside a building: a huge underground shopping center, a huge building, a shopping mall, and so on where GPS (Global Positioning System) signal is not available. An LBS which is provided inside a building is called an indoor location based service (ILBS). Meanwhile, the web service has been proved to be an excellent practice of software reuse. It is also well known that RESTful service is a real Web service. So, RESTful services for ILBS are in demand. As the first step of developing RESTful services for ILBS, this paper introduces the design of RESTful services for ILBS.

Keywords: LBS, ILBS, Web Service, REST

1 Introduction

Location-Based Services (LBS) in daily life can be found everywhere nowadays. Navigation, emergency management, logistics, taxi management, etc are examples of LBS. LBS are so useful that providing indoor LBS (ILBS) is very desirable. For example, huge buildings like COEX are especially demanding LBS. The essential ingredients of ILBS system include rendering drawings, indoor positioning, indoor moving object database, etc.

Meanwhile, it is well known that web services [1-3] are an excellent practice of software reuse. That is if the essential ingredients of ILBS system are published as web services, then numerous ILBS developers will easily implement their ILBSs using the published web services. It is also well known that RESTful services are real web services. Therefore, this paper introduces our design of RESTful services for ILBS.

2. Related Works

The essential ingredients of ILBS system include rendering drawings (A drawing is to ILBS as a map is to LBS), indoor positioning, indoor moving object database, etc [4,

5]. REST (Representational State Transfer) is an architectural style proposed by Roy Fielding [6]. The authors of [7] identify the characteristics of REST.

3. Functional Requirements

The followings are the functional requirements of the rendering drawing module:

- A user can upload a private drawing (only authenticated users can retrieve a private drawing).
- A user can upload a public drawing (anyone can retrieve a public drawing). - A user can retrieve a specific drawing.
- A user can retrieve the user's private drawings, filtering by tags is allowed.
- A user can retrieve the user's public drawings, filtering by tags is allowed. - A user can delete the user's private and public drawings.
- An authenticated user can delete any drawings.

The followings are the functional requirements of indoor positioning:

- It provides a result of WLAN (wireless local area network) based trilateration.
  - A user can upload location information of APs (access points).
  - A user can retrieve APs.
  - An authenticated user can update an existing AP. - An authenticated user can delete an AP.
  - It returns user's current location obtained by the trilateration method
- It provides a result of WLAN (wireless local area network) based K-NN method.
  - A user can upload a pair (coordinates, RSSIs) where RSSIs is an array of (MAC address, RSSI) pairs. RSSI is the signal strength of the signal from the AP identified by MAC address. A user collects RSSIs at the point of coordinates and uploads (coordinates, RSSIs).
  - An authenticated user can retrieve (coordinates, RSSIs), filtering by coordinates is allowed.
  - An authorized user can delete (coordinates, RSSIs).
  - A user can request the result of K-NN providing RSSIs

- It provides a result of the Kalman filter process.
  - A user can assign a value to the initial location.
  - A user can designate Q value.
  - A user can designate R value.
  - A user can request the result of the Kalman filter process providing a sequence of measured locations
  - A user can request the result of the Kalman filter process providing a sequence of measured RSSIs

- It provides a result of map matching.
  - A user can retrieve a drawing and filtering by tag is allowed.

- A user can assign a value to the initial location.
- A user can request the result of the map matching process providing a sequence of locations.

The followings are the functional requirements of the database:

– For each of the users, it periodically receives measured (Time, Location) tuple. - For each of the users, it estimates the user's track.

– For each of the users, it can estimates the user's current location with the user's track.

- Given a measured (Time, Location) tuple, it stores the tuple if Location is different from the estimated location.

– A user can retrieve (Time, Location) tuples. - A user can retrieve points of interest

The remaining part of user requirement analysis has been omitted because of the space limit.

4. Designing the URI templates

As we discussed in the earlier sections, a RESTful service is a service designed to embrace the "Web" from the ground up. "Web" stems from the Web architecture where resources contain hyperlinks to other resources, thereby creating a Web of resources. For interacting with resources, HTTP is used as a standard communication protocol. HTTP defines a standard set of methods, status codes, and headers. Therefore, in this section we identify resources and URI templates.

We'll need the ability to address the following types of resources (because of the space limit, only a part of the resources is listed):

An individual drawing
A user's collection of private drawings
A user's collection of public drawings
The collection of all public drawings
The collection of all APs
The collection of fingerprints, where (Time, Location) tuple is sometimes called a fingerprint
An individual user account
A specific user's public profile

Once the resources are identified, we design the URI templates

You can identify all public drawings named "museum" using

http://230.247.239.177/drawingservice?name=museum.

You can identify all of kim's drawings named "museum" using

http:// 230.247.239.177/drawingservice/kim?name=museum.

We'll use the following templates to identify a user's complete list of drawings:

/users/ {username /drawings

Plus, like before, we can filter drawings by using "name" scoping information:

/users/ {username}/drawings?name= {name}

Since drawings really belong to a specific user, it might make sense to make individual bookmark identifiers relative to a particular user as shown here: /users/{username}/drawings/{id}

For the publicly accessible drawings collections, we'll only support GET requests. We'll return a 200 ("OK") when the requests are successful. If the URI doesn't identify a known user, we'll return 404 ("Not Found"). For individual drawing resources, we'll support GET, POST, PUT, and DELETE requests. Because of the space limit, we omit the reaming part of design.

Our service authenticate users. If an unauthenticated user attempts to access a private drawing, the service returns a 401 ("Unauthorized") response. For the authentication, we use a custom Hash Message Authentication Code (HMAC) approach.

5. Conclusion

ILBS applications running on mobile devices is getting popularity. IPTV is known as the next killer application. Therefore, web services for location based mobile IPTV application will be demanded by app developers. Since it is known that a RESTful service is a real web service, the design of RESTful services introduced in this paper will contribute to promoting mobile IPTV. For the further research, we are implementing RESTful services for ILBS.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0002611) and (Grants No. C0033172) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012.

References

1. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applications Springer, (2004)
2. Stal, M.: Web Services: Beyond Component-Based Computing. Comm. ACM, vol. 55, no. 10, (2002)
3. Kook, Y., Lee, **J.,** Kim, W.: A Study on Data Gathering based on Agent for APC in FA. J. of Advanced Science and Technology, Vol. 2, 27--36 (2009)
4. Lee, G., Yim, J., Han, C.: Implementation of an Indoor-Location-Based Interactive VoD Player Using Web Services. in printing
5. Yim, J., Le, T.: Development of a Mobile VOD System Based on RESTful Web Services. in printing
6. http://www.ics.uci.eduf—fielding/pubs/dissertation/top.htm
7. http://www.xfront.com/REST-Web-Services.html