

ecrit
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2016

B. Rosen
Neustar
October 19, 2015

Validation of Locations Around a Planned Change
draft-rosen-ecrit-lost-planned-changes-03

Abstract

This document defines an extension to LoST (RFC5222) that allows a planned change to the data in the LoST server to occur. Records that previously were valid will become invalid at a date in the future, and new locations will become valid after the date. The extension adds two elements to the <findservice> request: a URI to be used to inform the LIS that previously valid locations will be invalid after the planned change date, and add a date which requests the server to perform validation as of the date specified. It also adds an optional TTL element to the response, which informs all queriers the current expected lifetime of the validation.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 21, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
3. <plannedChange> element	4
4. <locationInvalidated> object	4
5. uri Not Stored Warning	4
6. TTL in Response	5
7. Relax NG Schema	5
8. Security Considerations	8
9. IANA Considerations	8
9.1. Relax NG Schema Registration	8
9.2. LoST Namespace Registration	9
10. Normative References	9
Author's Address	9

1. Introduction

This document describes an update to the LoST protocol [RFC5222] which allows a <findservice> request to optionally add a URI and a date to be used with planned changes to the underlying location information in the server. The URI is retained by the LoST server, associated with the data record that was validated, and used to notify the LIS (the LoST client) when a location which was previously valid will become invalid. The date is used by the client to ask the server to perform validation as of a future date. In addition to this mechanism, the <lt;findserviceResponse> is also extended to provide a TTL for validation, after which the client should revalidate the location.

Validation of civic locations involves dealing with data that changes over time. A typical example is a portion of a county or province that was not part of a municipality is "annexed" to a municipality. Prior to the change, the content of the PIDF A3 element would be blank, or represent some other value and after the change would be the municipality that annexed that part of the county/province. This kind of annexation has an effectivity date and time (typically 00:00 on the first or last day of a month).

Records in a LIS must change around these kinds of events. The old record must be discarded, and a new, validated record must be loaded into the LIS. It is often difficult for the LIS operator to know

that records must be changed around such events. There are other circumstances where locations that were previously valid become invalid, such as a street renaming or renumbering event. As RFC5222 defines validation, the only way for a LIS to discover such changes was to periodically revalidate its entire database. Of course, this would not facilitate timely changes, is not coordinated with the actual change event, and also adds significant load to the LoST server. Even if re-validation is contemplated, the server has no mechanism to control, or even suggest the time period for revalidation

This extension allows the client to provide a stable URI that is retained by the server associated with the location information used in the request. In the event of a planned change, or any other circumstance where the LI becomes invalid, the server sends a notification to the URI informing it of a change. The notification contains the date and time when the LI becomes invalid.

Ideally, following such a notification, the LIS will prepare a new record to be inserted in its active database, that becomes active at the precise planned event date and time, at which point it would also delete the old record. However, the new record has to be valid, and the LIS would like to validate it prior to the planned change event. If it requests validation before the planned event, the server (without this extension) would inform the client that the location was invalid. This extension includes an optional "asOf" date and time in the request that allows the LoST server to provide validation as of the date and time specified, as opposed to the "as of now" implied in the current LoST protocol.

When it is not practical or advisable for the LIS to maintain stable URIs for all of its records, periodic revalidation can be still used to maintain the data in the LIS. However, the server should be able to control the rate of such revalidation. For this purpose, a new TTL element is included in the `lt;findserviceResponse>` which provides advice from the server to the LIS of when validation is suggested.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

"Server" in this document refers to the LoST server and "Client" is the LoST client, even when the server is performing an operation on the client.

3. <plannedChange> element

This document defines a new element to <findService> called 'plannedChange'. This element contains two attributes: 'uri' and 'asOf'. The 'uri' attribute MUST be a URI with a scheme of https. The URI will be stored by the server against the location in the request for subsequent use with the notification function defined below. To minimize storage requirements of at the server, the length of the URI MUST be less than 256 bytes. Each client of the server may only store one URI against a location, where "location" is defined by policy at the server, since a given unique location may have many combinations of LI elements that resolve to the same location. If the server receives a 'uri' for the same location from the same client, the URI in the request replaces the URI it previously retained. Policy at the server may limit how many uris it retains for a given location. A new warning is defined below to be used to indicate that the URI has not been stored. If the location in the request is invalid, the uri will not be stored and the warning will be returned.

The 'asOf' attribute contains a date and time. The server will validate the location in the request as of the date specified, taking into account planned changes. This allows the client to verify that it can make changes in the LIS commensurate with changes in the LoST server by validating locations in advance of a change.

4. <locationInvalidated> object

When the server needs to invalidate a location where the client provided a URI in <plannedChange>, the server executes an HTTPS POST containing <locationInvalidated> to the URI previously provided. This is the notice from the server to the client that the location may be invalid and should be revalidated. <locationInvalidated> contains an asOf attribute that specifies when the location may become invalid. If the date/time in asOf is earlier than the time the <locationInvalidated> was sent, the location may already be invalid and the LIS should take immediate action. If the POST operation fails, the server MAY retry the operation immediately, and if it fails again, retry the operation at a later time.

5. uri Not Stored Warning

A new warning is added to the exceptionContainer, 'uriNotStored'. This warning MUST NOT be returned unless the plannedChange element was found in the corresponding request. The warning is returned when the server decides not to store the URI found in the plannedChange element. As discussed above, this may occur because, among other reasons, the policy at the server limits how many URIs will be stored

against a specific location, the uri is not well formed or the policy at the server has some other restriction on the feature.

6. TTL in Response

A new 'ttl' element is added to the `lt:findserviceResponse`. The ttl element contains a date and time after which the client may wish to revalidate the location at the server. This element MAY be added by the server if validation is requested in the response. The form of the element is the 'expires' pattern, which allows explicit 'No Cache' and 'No Expiration' values to be returned. 'No Cache' has no meaning and MUST NOT be returned in TTL. 'No Expiration' means the server does not have any suggested revalidation period.

Selecting a revalidation interval is a complex balancing of timeliness, server load, stability of the underlying data, and policy of the LoST server. Too short, and load on the server may overwhelm it. Too long and invalid data may persist in the server for too long. The URI mechanism provides timely notice to coordinate changes, but even with it, it is often advisable to revalidate data eventually.

In areas that have little change in data, such as fully built out, stable communities already part of a municipality, it may be reasonable to set revalidation periods of 6 months or longer, especially if the URI mechanism is widely deployed at both the server and the clients. In areas that are quickly growing, 20-30 day revalidation may be more appropriate even though such revalidation would be the majority of the traffic on the LoST server.

When a planned change is made, typically the TTL for the affected records is lowered, so that revalidation is forced soon after the change is implemented. It is not advisable to set the expiration precisely at the planned change time if a large number of records will be changed, since that would cause a large spike in traffic at the change time. Rather, the expiration time should have a random additional time added to it to spread out the load.

7. Relax NG Schema

The Relax NG schema in [RFC5222] is extended to include:

```
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
default namespace ns1 = "urn:ietf:params:xml:ns:lost-plannedChange1"
```

```
##
##      Extension to Location-to-Service Translation (LoST) Protocol
##      to support a planned change to location data
```

```
##
##      plannedChange is used in the extensionPoint of
##      commonRequestPattern in a findService request
##
##      locationInvalidated is used by the LoST server to notify a
##      LIS that a previously valid location may be (or will become)
##      invalid
##
##      ttl is used in the extensionPoint of
##      commonResponsePattern in a findService response
##
##      uriNotStored is a new warning to be used in a
##      exceptionContainer in the warnings element of a
##      findServiceResponse
##
start =
  plannedChange
  | locationInvalidated
  | uriNotStored
##
##      plannedChange
##
div {
  plannedChange =
    element plannedChange {
      attribute uri {
        xsd:anyURI }?,
      attribute asOf {
        xsd:dateTime }?,
      extensionPoint+
    }
}

##
##      locationInvalidated
##
div {
  locationInvalidated =
    element locationInvalidated {
      attribute asOf {
        xsd:dateTime }?,
      extensionPoint+
    }
}

##
##      ttl
##
```

```
div {
  ttl =
    element ttl {
      expires,
      extensionPoint+
    }
}

##
##      uriNotStored
##
div {
  uriNotStored =
    element uriNotStored { basicException }
}

##
##      Patterns for inclusion of elements from schemas in
##      other namespaces.
##
div {

  ##
  ##      Any element not in the LoST namespace.
  ##
  notLostChange = element * - (ns1:* | ns1:*) { anyElement }

  ##
  ##      A wildcard pattern for including any element
  ##      from any other namespace.
  ##
  anyElement =
    (element * { anyElement }
     | attribute * { text }
     | text)*

  ##
  ##      A point where future extensions
  ##      (elements from other namespaces)
  ##      can be added.
  ##
  extensionPoint = notLostChanged*
}
```

8. Security Considerations

As an extension to LoST, this document inherits the security issues raised in [RFC5222]. The server could be tricked into storing a malicious URI which, when sent the locationInvalidated object could trigger something untoward. The server **MUST NOT** accept any data from the client in response to POSTing the locationInvalidated.

The server is subject to abuse by clients because it is being asked to store something and may need to send data to an uncontrolled URI. Clients could request many URIs for the same location for example. The server **MUST** have policy that limits use of this mechanism by a given client. If the policy is exceeded, the server returns the uriNotStored warning. The server **MUST** validate that the content of the uri sent is syntactically valid and meets the 256 byte limit. When sending the locationInvalidated object to the uri stored, the server **MUST** protect itself against common http vulnerabilities.

The mutual authentication between client and server when is **RECOMMENDED** for both the initial findService operation that requests storing the uri and the sending of the locationInvalidated object. The server should be well known to the client, and its credential should be learned in a reliable way. For example, a public safety system operating the LoST server may have a credential traceable to a well known Certificate Authority known to provide credentials for public safety agencies. Many of the clients will be operated by local ISPs or other service providers where the server operator can reasonably obtain a good credential to use for the URI. Where the server does not recognize the client, its policy **MAY** limit the use of this feature beyond what it would limit a client it recognized.

9. IANA Considerations

9.1. Relax NG Schema Registration

URI: urn:ietf:params:xml:schema:lost-plannedChange1

Registrant Contact: IETF ECRIT Working Group, Brian Rosen
(br@brianrosen.net).

Relax NG Schema: The Relax NG schema to be registered is contained in Section 5. Its first line is

```
default namespace = "urn:ietf:params:xml:ns:lost-PlannedChange1  
  
and its last line is  
  
}
```


9.2. LoST Namespace Registration

URI: urn:ietf:params:xml:ns:lost-plannedChange1

Registrant Contact: IETF ECRIT Working Group, Brian Rosen
(br@brianrosen.net).

XML:

```
BEGIN
<?xml version="2.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
    content="text/html; charset=iso-8859-1"/>
  <title>LoST Planned Change Namespace</title>
</head>
<body>
  <h1>Namespace for LoST Planned Change extension</h1>
  <h2>urn:ietf:params:xml:ns:lost-plannedChange1</h2>
  <p>See <a href="http://www.rfc-editor.org/rfc/rfc?????.txt">
    RFC?????</a>.</p>
</body>
</html>
END
```

10. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol", RFC 5222, DOI 10.17487/RFC5222, August 2008, <<http://www.rfc-editor.org/info/rfc5222>>.

Author's Address

Brian Rosen
Neustar
470 Conrad Dr
Mars, PA 16046
US

EMail: br@brianrosen.net