LCP for Lending 1.0

Status: Draft
Editors: Jean-Philippe Bougie (De Marque), Hadrien Gardeur (Feedbooks)

1. Overview

1.1 Purpose and Scope

This section is informative

This specification, Lightweight Content Protection (LCP) for Lending (1.0), defines an extension to the LCP 1.0 vocabulary in order to support lending and other time-limited usages. It describes a set of Web Services to be implemented by the Content Provider as well as a behavior for Reading Systems to respect.

Its main objective is to support lending in public libraries where licensing and usage constraints mean a limited number of concurrent licenses may be issued, both from the standpoint of the

User, who may only check out a certain number of publications concurrently, as well as from the public library, who may only be able to grant a limited number of copies to its users concurrently.

LCP for Lending builds on and requires a valid LCP 1.0 License, in particular with proper values for the `end` element in the user rights.

1.2. Terminology

This specification uses the following EPUB and LCP terms:

Publication
A logical document entity consisting of a set of interrelated resources and packaged in an EPUB Container, as defined by the EPUB 3 specifications.

User
An individual that consumes an EPUB Publication using an EPUB Reading System.

EPUB Reading System (or Reading System)
A system that processes EPUB Publications for presentation to a User in a manner conformant with the EPUB 3 specifications.

License Document
Document that contains references to the various keys, restrictions that are applied to the LCP encrypted files and user information. When embedded in a Publication this file is always located at `META-INF/license.lcpl`.

Additionally, it uses the following terms:
Lending
A central authority, typically a public library, giving a time-limited License for a User to access the content.

Content Provider
In LCP, the authority that grants the License Document and the Publication to the User, usually a Web Server

Early Return
The ability for a User to signify to the Licence Issuer that he wishes to invalidate his license before the described time.

1.3. LCP for Lending Overview and Example

This section is informative

LCP can already describe time-limited licenses using the `start` and `end` elements in the `rights` section of the Licensing Document. LCP for Lending builds on that by allowing workflows that necessitate further communication with the Content Provider, such as early returns or license renewals. The License Document MUST contain these two elements to be considered for LCP for Lending.

A Licensing Issuer granting a Lending or Time-Limited License MAY add up to three elements to the License Document. These elements describe URLs that will allow a Reading System to:

- Check the availability of the current License
- Ask the Content Provider to revoke or renew the license

Calling the availability URL will allow a Reading System to fetch updated information about the License: whether any User action has resulted in voiding the License or Renewing the license, if the revocation or renewal actions are available at the time, and if an updated license is available for retrieval.

Calling the other 2 URLs (renewal and revocation) will trigger a server-side check on the possibility of doing the requested actions. In particular, a Content Provider should have business rules about the renewal. Typically, those are whether the User has an account in good standing, if no other user has signified interest in accessing the same publication, or simply if a limited number of renewals has been reached. The presence and behavior of these business rules is implementation-dependent and thus out of scope for this specification.

If the request for either a revocation or a renewal is accepted, a new License will be emitted and transmitted to the client. This new License will contain updated values for the start and end dates. The Reading System then has to replace the current license with the updated one in the container, and then simply follow the normal behavior.

1.4 Conformance Statements

The keywords 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].

All sections of this specification are normative except where identified by the informative status label "This section is informative". The application of informative status to sections and appendices applies to all child content and subsections they may contain.

All examples in this specification are informative.

2. Extensions to the License Document

2.1. Introduction

This section is informative

LCP for Lending does not add any structural elements to the LCP License Document. It only requires additional elements in the `links` object.

2.2. In relation to the `links` object

Link back to LCP spec to the links section
Will be specified in the registry

2.3. License Availability

The `availability` element in the `links` object will define a URL that can be called. Its content type should be something specific to LCP (or application/json as default).

Calling this URL will return a JSON object, called the Lending Document, that will describe the state of the current License.

2.4. License Revocation and Renewal

Simple revocation (for early returns) will be specified with the `return` property of the `links` object. The content type for this link should follow LCP (`application/vnd.readium.lcp.license.1-0+json`)
A renewal of the license will be specified with the `renew` property of the `links` object. The content type for this link should also follow LCP (`application/vnd.readium.lcp.license.1-0+json`)
These properties are optional, and their absence signifies a lack of supported functionality.

3. Server Interactions

3.1. Introduction

This section is informative

The Reading System will interact with the Content Provider by using unauthenticated, HTTP 1.1 requests, preferably made over HTTP. Two services will be to acquire information (the state of the license, and the updated License Document itself), and two services to ask for two possible operations to be made on the license, to revoke or renew it.

3.2. Conformance Statements

...

3.3. Checking License Availability

A Reading System MAY, at any time, make a HTTP request to the URL provided in the `availability` item of the License Document. The host and path must be kept as is.

This request MUST use the GET method and MUST have an `Accept` header containing `application/vnd.readium.lcp.lending.1-0+json`.

In addition, a Reading System SHOULD identify itself using the `User-Agent` header.

No other parameters SHOULD be added, either in the request path or body.

The Content Provider SHOULD return with a valid HTTP response containing a status code and the body of the response.
A Reading System MAY follow redirects in the 300 range according to the rules of the HTTP specification. Otherwise, a Reading System may only consider a response valid if it has a 200 status code and a content type of `application/vnd.readium.lcp.lending.1-0+json`.

A lack of response, a response with any other code or any other content type MAY be ignored or MAY be presented to the User.

If given cache headers per the HTTP 1.1 specification, the Reading System SHOULD attempt to respect those headers.

The response MUST follow the rules of JSON encoding and have the following structure:

```
{
   “id”: “234-5435-3453-345354” //Refers to the License Identifier
   “valid”: false,
   “message”: “Your loan has expired”,
   “updated”: “2013-01-13T00:00:00Z”,
   “start”: “2013-01-13T00:00:00Z”,
   “end”: “2013-01-27T00:00:00Z”,
   “links”: {
```

```
     "license": { "href":
"http://example.org/licenses/12343?signature=..." },
     "return": { "href":
"http://example.org/licenses/12343?signature=..."},
     "renew": {"href":
"http://example.org/license/1234/renew?signature=.."}
  },
  "renewal": {
     "end": "2013-02-12T12:34:00Z",
     "message": "You're out of renewals for this license",
     "message": "Our public library allows 3 renewals assuming this
and that"
  },
  "events": [
     {kind: "returned", agent: "ReadiumSDK1.0", time:
"2014-01-13T00:00:00Z", message: "", device: ""}
     ]
}
```

The following fields are part of the document root.

Name	Value	Format/data type	Required?
id	The License Identifier	String	Yes
valid	Whether the license can still be used	Boolean	Yes
message	A human-readable message giving the current state of the License	String	No
updated	The point in time at which the license was last modified	ISO8601 in a string	Yes
start	The start of the license	ISO8601 in a string	Yes
end	The current end of the license	ISO8601 in a string	Yes

In addition, a link object is required and has the following fields

Relation	Semantics	Required?
license	The URL at which the up-to-date license must be fetched	Yes
return	The URL at which a Reading System may ask for an early return of the License	No
renew	The URL at which a Reading System may ask for an extension to the License	No

Each link item has the following fields:

Name	Value	Format/data type	Required?
`href`	Link location	URI	Yes
`title`	Title of the link	String	No
`type`	Expected MIME media type value for the external resources	MIME media type	No, but highly recommended
`length`	Content length in octets	Integer	No

Return and renewal links should only be available if the Content Provider supports and allows these features, and the license is still active (has not been returned). The refresh link is available to get an updated version of the license.

The renewal object, if present, has the following structure:

Name	Value	Format/data type	Required?
`end`	The proposed end date that would be set by a license renewal	ISO8601 in a string	Yes
`message`	A human-readable message explaining the conditions of the renewal, or if unavailable (temporarily or permanently), the reasons	String	Yes

The events item contains an array of events items, each of which must respect this structure:

Name	Value	Format/data type	Required?
`kind`	The type of event, either renewal or early_return	String	Yes
`time`	The time at which the event occured	ISO8601	Yes
`device`	An identifier for the device, typically named by the User	String	Yes
`agent`	A user agent string representing the type of Reading System	String	No
`message`	A human-readable message explaining the event	String	No

3.4. Revoking a license

If available and upon User request, Reading Systems MAY attempt to revoke the License, in essence asking for an early return. In order to do so, Reading Systems MUST make a HTTP 1.1 request to the specified `revoke` URL using the POST method. The `Accept` Header MUST at least specify the current content type according to the LCP specification as handled by the Reading System. In addition, a Reading System SHOULD identify itself using the `User-Agent` header. No additional request body or query string parameters are required.

A successful request will receive a response with a 200 status code containing a new license, similar to the previous one, but with the end date set to the time at which the license was returned. The Reading System should then proceed to verify the Licence Document according to the LCP specification and then replace the License embedded in the Package.

3.5. Renewing a license

If available and upon User request, Reading Systems MAY attempt to renew the License. In order to do so, Reading Systems MUST make a HTTP 1.1 request to the specified `renew` URL using the POST method. The `Accept` Header MUST at least specify the current content type according to the LCP specification as handled by the Reading System. In addition, a Reading System SHOULD identify itself using the `User-Agent` header. No additional request body or query string parameters are required.

A successful request will receive a response with a 200 status code containing a new license, similar to the previous one, but with the end date set to the time at which the license will now expire. The Reading System should then proceed to verify the Licence Document according to the LCP specification and then replace the License embedded in the Package.

4. Reading System Behavior

On a first activation of a License, the Reading System SHOULD:

- Check for the presence of an availability item in the links object;
- If the Reading System has internet connectivity, it MUST attempt a request on the availability URL, specifying it is a first activation;
- Otherwise, plan to visit that URL as soon as Internet connectivity is established.

Every quarter of the duration of the License, the Reading System SHOULD repeat the same procedure. Additionally, the Reading System MAY repeat the procedure every time a new reading session is opened by the User.

When a Reading System can not reach the availability URL:

- If the current License Document is valid, the Reading System MUST grant access to the content.
- If the current License Document is expired, the Reading System MUST deny access to the content and provide a message telling the User he might attempt to get connectivity in order to re-validate the license.

Upon receiving a successful response from the Availability Service:

- If the `valid` element is set to false, the Reading System MUST revoke access to the Content and MUST inform the user. The Reading System MAY use the `message` provided in the Lending Document.
- If the `updated` date from the Lending Document is more recent than that of the License Document, the Reading System MUST attempt to refresh the License Document using the `license` link.

Upon obtaining a new License Document:

- The Reading System MUST replace the License Document embedded in the Package.
- The Reading System MUST re-evaluate the validity of the License using the new License Document.

Upon a User request to Revoke the License:

- If Internet Connectivity is present, the Reading System MUST call the Revocation Service using the prescribed method.
- Upon a valid and successful response, the Reading System MUST refresh the License Document and MUST inform the User that its license has been returned.
- If no response has been obtained from the server or a response status is in the 500 range, a Reading System MAY retry the service again.

Upon a User request to Renew the License:

- If Internet Connectivity is present, the Reading System MUST call the Revocation Service using the prescribed method.

- Upon a valid and successful response, the Reading System MUST refresh the License Document and MUST inform the User that its license has been extended.
- If no response has been obtained from the server or a response status is in the 500 range, a Reading System MAY retry the service again.

Normative references

[JSON] The application/json Media Type for JavaScript Object Notation (JSON).
[Publications] *EPUB Publications 3.0.1*.
HTTP
LCP

Informative references

...