

PatREST: Patron REST v1.3 : Specification Overview

Last revision : September 14, 2006

Author: John Blyberg <blybergj@aadl.org>

Table of Contents

1.0 PURPOSE

1.1 ABOUT

1.2 LICENSE

2.0 ACCESSING THE SERVICE

2.1 URL FORMAT

2.2 XML RESULTS

3.0 THE FUNCTIONS

3.1 SEARCHING THE CATALOG

3.2 VIEWING A SINGLE RECORD

3.3 VIEWING TOP ITEMS DATA

3.4 VIEWING NEW ITEMS DATA

3.5 VIEWING CHECKED-OUT ITEMS

3.6 VIEWING ON-HOLD MATERIAL

4.0 ADDITIONAL INFORMATION

4.1 ADDITIONAL RESOURCES

4.2 ACKNOWLEDGMENTS

1.0 PURPOSE

The purpose of this document is to provide an overview of the PatREST schema, developed and deployed by Ann Arbor Public Library. This document has been written to serve two purposes. 1) To provide clear and concise documentation to the PatREST API for developers. 2) To provide schematic information to other libraries that may consider implementing a user-friendly API and who may wish to provide a compatible abstraction layer to applications written for PatREST.

1.1 ABOUT

PatREST (Patron REST) is an XML specification developed at the Ann Arbor District Library for the purpose of providing a simple and easy method of accessing various data and methods. The PatREST service is intended to be used by both professional and amateur programmers as it's data objects are clean, simple and intuitive. The idea behind having a simple interface to online library services is to bring library-oriented development tools into the hands of non-librarians--the library users themselves.

1.2 LICENSE

PatREST, this document, and the PatREST schema are licensed under GPL version 2 (<http://www.gnu.org/licenses/gpl.txt>). Libraries and vendors may feel free to claim they are providing a PatREST API if their abstraction layer complies with this document.

2.0 ACCESSING THE SERVICE

The PatREST service is hosted on a web server and like many other XML-based APIs can be accessed via regular port 80 queries. PatREST can be hosted on custom ports, but the standard method of presenting it is on port 80. Developers may want to provide an easy way of specifying port number in their app.

2.1 URL FORMAT

PatREST attempts to follow the recommended REST-style URL format. Instead of using the standard '?foo=' and '&bar=' convention, all URL path positions have their own correlating meaning or value. Typically, the URL path begins with a path to invoke PatREST, followed by the function call, then any supporting variables. For example:

<http://your.server.org/rest/functioncall/supporting/data/variables/>

This allows developers to quickly and easily add your web service component into their code, and simplifies the invocation.

2.2 XML RESULTS

Both successful and unsuccessful attempts to make a call to the PatREST interface will result in a XML-encoded result that can be parsed by many software languages.

PatREST is also XSLT-friendly, meaning that custom style-sheets may be applied to the xml so that it can be displayed in new and interesting ways.

3.0 THE FUNCTIONS

The following is a list of the functions currently available via PatREST. Note that even though this specification can be ported to any language at any site, I will be using the system at Ann Arbor District Library to provide examples.

3.1 SEARCHING THE CATALOG

The URL format for performing a search looks like the following:

`/rest/search/[search type]/[url-encoded search string]/[# results per page]/[page #]`

The search type can be one of the following and should be in lowercase:

Trigger	Value
title	Search by title
author	Search by author
callnum	Search by call number
keyword	Keyword search
subject	Subject search
gvtdocnum	Search by government document number
stdnum	Search by standard number (lie ISBN)
titlekey	Search by title key

Trigger	Value
controlnum	Search by the control number
barcode	Search by barcode number
record	Search by record number
bibnum	Search by bibliographical number
itemnum	Search by item number

No matter which search-type you choose, the result set will always be returned in the same format.

Using the # of results per page option allows coders to scale the number of results in a single response. The Page # option tells PatREST which page of results to display given the results/page value.

The result-set looks like this:

```
<SearchResult>
  <searchtype>[Search type from URL]</searchtype>
  <searchstring>[Search string from URL]</searchstring>
  <Records xmlns:p="http://www.aadl.org"
xmlns:xlink="http://www.w3.org/1999/xlink">
  ....
    <Record id = "[Record #]" xlink:href="[Link to human-readable re-
cord]" xlink:title="[Title of Material]">
      <bibnum>[Bibliographical #]</bibnum>
      <fulltitle>[Full title of material]</fulltitle>
      <xmlrecordlink>[URL to the REST RECORD]</xmlrecordlink>
    </Record>
  ....
</Records>
</SearchResult>
```

The ellipses denote XML code that may repeat itself within the result. Essentially, that is the “hit-list” portion of the result. As you can see, PatREST emphasizes simplicity in the data it requires as well as the results it provides.

3.2 VIEWING A SINGLE RECORD

Using an item’s bibliographical number, it is possible to view a title’s record. The syntax for that PatREST call looks like:

/rest/record/[Bibliographical #]/

PatREST will return a result that looks like this (example from AADL):

```
<Record xmlns:p="http://www.aadl.org"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <bibnum>[Bib #]</bibnum>
  <callnum>[Call #]</callnum>
  <author xlink:href="[link to url-encoded PatREST author search" >
[Author]</author>
  <fulltitle>[Full title]</fulltitle>
  <title>[Title]</title>
  <edition>[Edition info]</edition>
  <pubinfo>[Publication info]</pubinfo>
  <desc>[Description]</desc>
  <isbn>[ISBN]</isbn>
  <lang>[Language]</lang>
  <copies>[# copies in system]</copies>
  <catdate>[Catalogued date]</catdate>
  <mattype>[Material type]</mattype>
  <altisbn>[Multiple Alternate ISBNs from OCLC's xISBN service]</altisbn>
  <audience_level>[A 0-1 scale of audience level complexity from OCLC]</
audience_level>
  <audience_level_percentile>[Where this material is in audience percen-
tile, also from OCLC]</audience_level_percentile>
  <avail>[# copies available]</avail>
  <coverimglink xlink:href="[Link to cover image]">[Link to cover image]</
coverimglink>
  <ccimglink xlink:href="http://www.aadl.org/cat/ccimg/1271127">[Link to
]</ccimglink>
  <recordlink
xlink:href="http://www.aadl.org/cat/seek/record=1271127">http://www.aadl.org/
cat/seek/record=1271127</recordlink>
</Record>
```

All data associated with the record is returned in a clean and easy-to-access format.

3.3 VIEWING TOP ITEMS DATA

Using PatREST, it's possible to access lists of popular items--also known a *top items*. The syntax for viewing top item data is:

/rest/top/[material type]/[results per page]/[page #]

Material type can be: books, dvd, cd, or bocd (books on cd)

XML results from a top items query looks like:

```
<TopItems>
  <RankLow>[Search type from URL]</RankLow>
```

```

    <RankHigh>[Search string from URL]</RankHigh>
    <MaterialType>[Specified material type]</MaterialType>
    <Records xmlns:p="http://www.aadl.org"
xmlns:xlink="http://www.w3.org/1999/xlink">
....
    <Record id = "[Record #]" xlink:href="[Link to human-readable re-
cord]" xlink:title="[Title of Material]">
        <bibnum>[Bibliographical #]</bibnum>
        <fulltitle>[Full title of material]</fulltitle>
        <author>[Author of material if avail.]</author>
        <isbn>[Material's ISBN if avail.]</isbn>
        <imglink>[Cover image link if avail.]</imglink>
        <xmlrecordlink>[URL to the REST RECORD]</xmlrecordlink>
        <recordlink>[URL to catalog record]</recordlink>
        <rank>[Popularity Ranking]</rank>
        <materialtype>[Material type]</materialtype>
    </Record>
....
</Records>
</TopItems>

```

The MaterialType field will not be present if no material type is specified in the URL.

3.4 VIEWING NEW ITEMS DATA

Much like the top items data, new items data return an XML object of newly catalogued material. The URL syntax for a new items query looks like:

```
/rest/new/[material type]/[URL-encoded subject match]/[results per page]/[page #]
```

Material type can be: books, dvd, cd, or bocd (books on cd)

Subject match can be any string used to match an items subject heading. Like a regular search, it must be URL-encoded before the function is called.

XML results from a top items query looks like:

```

<NewItem>
    <MaterialType>[Specified material type]</MaterialType>
    <SubjectSearch>[Specified subject search]</SubjectSearch>
    <Records xmlns:p="http://www.aadl.org"
xmlns:xlink="http://www.w3.org/1999/xlink">
....
    <Record id = "[Record #]" xlink:href="[Link to human-readable re-
cord]" xlink:title="[Title of Material]">
        <bibnum>[Bibliographical #]</bibnum>

```

```

    <fulltitle>[Full title of material]</fulltitle>
    <author>[Author of material if avail.]</author>
    <isbn>[Material's ISBN if avail.]</isbn>
    <imglink>[Cover image link if avail.]</imglink>
    <xmlrecordlink>[URL to the REST RECORD]</xmlrecordlink>
    <recordlink>[URL to catalog record]</recordlink>
    <materialtype>[Material type]</materialtype>
  </Record>
....
  </Records>
</NewItem>

```

MaterialType and SubjectSearch fields will be present only if their corresponding values are specified in the URL.

3.5 VIEWING CHECKED-OUT ITEMS

One of PatREST's advanced features is its ability to return an XML object of a user's checked-out material. This functionality may be called using the following URL syntax:

```
/rest/checkouts/[unique token]
```

The unique token is a 32-character MD5 encrypted hash that is unique to each user. This is the same hash used for the patron-specific RSS feeds on the AADL website and can be found on the MyAccount page. Other libraries attempting to implement this PatREST function may choose to use another authentication mechanism as long as it adheres to this URL structure.

This function's XML object looks like:

```

<Checkouts xmlns:p="http://www.aadl.org"
xmlns:xlink="http://www.w3.org/1999/xlink">
....
  <Record id = "[Record #]" xlink:href="[Link to human-readable record]"
xlink:title="[Title of Material]">
    <bibnum>[Bibliographical #]</bibnum>
    <fulltitle>[Full title of material]</fulltitle>
    <title>[Short title of the material]</title>
    <author>[Author of material if avail.]</author>
    <isbn>[Material's ISBN if avail.]</isbn>
    <imglink>[Cover image link if avail.]</imglink>
    <xmlrecordlink>[URL to the REST RECORD]</xmlrecordlink>
    <recordlink>[URL to catalog record]</recordlink>
    <duedate>[Due-date in standard YYYY-MM-DD format]</duedate>
  </Record>
....

```

</Checkouts>

3.6 VIEWING ON-HOLD MATERIAL

PatREST can also return items on hold and their current status. To access this function, use the following URL syntax:

/rest/holds/[unique token]

The unique token is a 32-character MD5 encrypted hash that is unique to each user. This is the same hash used for the patron-specific RSS feeds on the AADL website and can be found on the MyAccount page. Other libraries attempting to implement this PatREST function may choose to use another authentication mechanism as long as it adheres to this URL structure.

This function's XML object looks like:

```
<Holds xmlns:p="http://www.aadl.org"
xmlns:xlink="http://www.w3.org/1999/xlink">
....
  <Record id = "[Record #]" xlink:href="[Link to human-readable record]"
xlink:title="[Title of Material]">
    <bibnum>[Bibliographical #]</bibnum>
    <fulltitle>[Full title of material]</fulltitle>
    <title>[Short title of the material]</title>
    <author>[Author of material if avail.]</author>
    <isbn>[Material's ISBN if avail.]</isbn>
    <imglink>[Cover image link if avail.]</imglink>
    <xmlrecordlink>[URL to the REST RECORD]</xmlrecordlink>
    <recordlink>[URL to catalog record]</recordlink>
    <holdstatus>[Current status of the hold]</holdstatus>
    <pickuploc>[Pickup location]</pickuploc>
    <canceldate>[Due-date in standard YYYY-MM-DD format]</canceldate>
  </Record>
....
</Holds>
```

4.0 ADDITIONAL INFORMATION

PatREST is an evolving specification. This document will change as PatREST does. As the author of PatREST, my intention is to ensure that the specification will always be backward-compatible. That is to say that as the specification evolves, existing applica-

tions will not break due to a change in the XML schema. PatREST has been developed with extensibility in mind.

4.1 ADDITIONAL RESOURCES

Please feel free to experiment with the AADL PatREST implementation. If you would like to use it at your own institution, you may find AADL's XSLT file useful. It can be found at:

<http://www.aadl.org/misc/opac.xsl>

4.2 ACKNOWLEDGMENTS

Many thanks to Ed Vielmetti for being the perpetual beta-tester.