

THE POWER OF APIs:

Exploiting Bibliographic & Item Data

A Case Study by David Kay, Owen Stephens and Steve Schoen

NOVEMBER 2014

GETTING PRACTICAL ABOUT PRINCIPLES

In the White Paper entitled '*No Library is an Island: How APIs can empower libraries in an evolving service ecosystem*' (June 2014), we observed:

However wide-ranging the services encompassed in a single vendor platform, there will be increasingly more new services to integrate – to support the user, to integrate within the institution, and to optimize supply chain interactions. Consequently, no library system can be an island, and each system must therefore, like the web giants, provide APIs to enable extended services for new and unforeseen opportunities.

The White Paper goes on to enumerate the benefits of APIs (Application Programming Interfaces) in enabling more cost-effective support of essential library processes that serve the user and empower the library. By offering seamless integration based on reliable data, libraries can provide seamless synchronization without human intervention and secure software and data.

However, those principles and the expected efficiency and effectiveness gains outlined in the White Paper should only be of interest to libraries if they can be applied in practice to everyday operational settings that benefit real patrons, operations, or management.

This Case Study takes particular scenarios where bibliographic and item data is made available through such an API and explores how that might bring benefits to the patrons and services.

SCENARIOS

Integrated Library System (ILS) APIs typically have a keen focus on exposing bibliographic data and associated items. The assumption is that there is a wide range of useful things that a library will want to do with this data, such as delivering it through other user-facing systems, feeding it into management reporting processes, or simply exposing it to interested developers.

In reality, API provisioning is a tightrope walk between providing a general facility to be used according to local imagination and shaping something to do a specific task really well. For an API provider such as Innovative, balancing these extremes is crucial.

For a library organization wanting to get the most out of its systems, the only place to start is the specific, as nobody has the time or the budget to dream things up from the general.

In this Case Study, we examine a number of specific user stories that a library might wish to address to deliver operational benefits. The bonus is that, if a library system's API can support these, it is likely to be flexible enough to meet other requirements as they emerge.

The following user stories, benefitting both patrons and the library team, suggest how bibliographic and item data can be put to work, above and beyond the functions provided in Sierra or any other management system.

"The Sierra API provides efficient and quick access to library catalog data, especially around circulation status. We are excited to work with Innovative to significantly improve the user experience for libraries by integrating the Sierra platform with the BiblioCore catalog. We look forward to more API additions and to collaborating with Innovative to improve and extend our integration."

Patrick Kennedy – Co-Founder and President, BiblioCommons

SCENARIOS EXPLORED

The White Paper emphasizes that APIs can offer benefits and efficiencies by making it easier to interact with existing data, to work across disparate systems, and to work in new and different ways:

- **Improve efficiency through automation**
- **Enable data flow between systems**
- **Enabling new functionality and workflows**
- **Allow other than the system vendor to build on your systems and services**

Let's test this against our nine user stories, bearing in mind what you can do now within your ILS.

Use of the API	User Story example
1 - Provisioning a discovery layer such as EDS, BiblioCommons, Blacklight, VuFind, or your own search service	As a patron or library staff member, I want an up-to-date view of the catalog to be discoverable alongside other available resources so I can make the best plans for my studies.
2 - Providing data to third-party discovery services such as Trove or Digital Public Library of America	As a patron or library staff member, I want to be able to discover resources through external services.
3 - Building up lists of course readings and their availability, whether in a Reading List system, a Course Management system, or in some other form of reference management application	As a course leader, I want to check the stock levels and availability of recommended course readings so I can ensure suitable provision.
4 - Building lists of books for reading groups or for promotional purposes (e.g. bestseller lists)	As a library staff member, I want to promote books to patrons and support reading groups.
5 - Checking availability efficiently through a simple mobile phone or tablet App before the patron visits the library or scours the shelves	As a patron or librarian, I want to check availability and locations of print stock from my mobile device on the library floor in order to save time.
6 - Cross-checking the library catalog before purchasing from online stores such as Amazon, enabled through a browser extension	As a patron, I want to check availability in the library before purchasing a title online in order to choose the best option.
7 - Providing a service to update patrons on the latest addition to stock through a range of channels, such as RSS Feeds, Twitter, the library website and SMS Text alerts	As a patron, I want to be alerted of new titles in a chosen subject area, fiction genre, or by a specific author in order to make the most of the library service.
8 - Extending the management dashboard through visualization tools (such as Many Eyes) to track collection development activity	As collections team leader, I want to track and visualize key indicators regarding accessions and weeding as reflected in the public catalog.
9 - Automatically check a vendor eBook package against local stock information	As a librarian, I want to easily check which items in a vendor eBook package are already held by the library and in what quantities.

PROVISIONING A DISCOVERY LAYER

... such as EDS, BiblioCommons, Blacklight, VuFind, or your own search service.

Solution benefits and opportunities

Efficiency through automation: Regularity and reliability of updating
Data flow between systems: ILS to Discovery Layer
New functions & workflows: Extend to include reservation
Enabling others to build on LMS: Discovery Layer partners

Other Systems Involved

A discovery product such as EDS, BiblioCommons, Blacklight, VuFind, or similar product.

Development requirement

Typically requires some middleware or connector that retrieves bibliographic and holdings data from an ILS, inserts new or changed records into the discovery product, and removes deleted or suppressed records.

In some cases, the discovery product vendor may provide such connectors.

API Requirements

- Improve efficiency through automation
- Enable data flow between systems
- Enabling new functionality and workflows
- Allow other than the system vendor to build on your systems and services

Data Requirements

- Improve efficiency through automation
- Enable data flow between systems
- Enabling new functionality and workflows
- Allow other than the system vendor to build on your systems and services

Skills required

- Improve efficiency through automation
- Enable data flow between systems
- Enabling new functionality and workflows
- Allow other than the system vendor to build on your systems and services

Sustainability considerations

Once established, a connector between a metadata source and a discovery layer is expected to work with minimal maintenance.

Changes to any of the systems involved may necessitate changes to the connectors between the systems.

PROVIDING DATA TO THIRD PARTY DISCOVERY SERVICES

... such as Trove or Digital Public Library of America.

Solution benefits and opportunities

Efficiency through automation: Regularity and reliability of updating
Data flow between systems: ILS to third party services
New functions & workflows: Extend to include reservation
Enabling others to build on LMS: Third-party services such as Trove and the Digital Public Library of America offer the ability to build applications across many collections.

Other Systems Involved

Third-party 'discovery' services such as Trove (Australia) and the Digital Public Library of America.

Development requirement

Many major third-party discovery services support a variety of mechanisms to add, update, and remove records.

API Requirements

- Improve efficiency through automation
- Enable data flow between systems
- Enabling new functionality and workflows
- Allow other than the system vendor to build on your systems and services

Data Requirements

- Improve efficiency through automation
- Enable data flow between systems
- Enabling new functionality and workflows
- Allow other than the system vendor to build on your systems and services

Skills required

- Improve efficiency through automation
- Enable data flow between systems
- Enabling new functionality and workflows
- Allow other than the system vendor to build on your systems and services

Sustainability considerations

Once established, a connector between a metadata source and a discovery service is expected to work with minimal maintenance.

Changes to any of the systems involved may necessitate changes to the connectors between the systems.

BUILDING UP LISTS OF COURSE READINGS

... and their availability, whether in a Reading List system, a Course Management system, or in some other form of reference management application.

Solution benefits and opportunities

Efficiency through automation: Reduced manual keying
Data flow between systems: ILS to RLMS or CMS
New functions & workflows: Potentially generate acquisition requests
Enabling others to build on LMS: Faculty or systems suppliers

Other Systems Involved

A Reading List Management system (such as Talis Aspire, rebus:list, LORLS) and/or a Learning Management System/Virtual Learning Environment (such as Blackboard, Moodle, Desire2Learn, etc.).

Development requirement

Typically requires some middleware or connector that retrieves

bibliographic and holdings data from an ILS and adds to a reading list management system or similar.

In some cases, the vendor of a reading list management system or course management system may provide such a connector.

API Requirements

- Ability to carry out basic searching (ISBN, author/title) to retrieve bibliographic records
- Ability to retrieve item records based on the bibliographic record they are attached to

Data Requirements

- Simple metadata for bibliographic resources
- Item location details
- Item availability details (e.g. on shelf, on loan, due date)

Skills required

- Programming skills
- Experience working with RESTful APIs
- Knowledge of JSON format

Sustainability considerations

Once established, a connector between a metadata source and a reading list management system is expected to work with minimal maintenance.

Changes to any of the systems involved may necessitate changes to the connectors between the systems.

BUILDING LISTS OF BOOKS FOR READING GROUPS

...or for promotional purposes (e.g. bestseller lists)

Solution benefits and opportunities

Efficiency through automation: Reduced manual keying
Data flow between systems: ILS to an external application which can display content on the web
New functions & workflows: Support reading groups
Enabling others to build on LMS: Library team or third-party application developer

Other Systems Involved

External systems supporting information about book lists, such as the NYT Best Sellers API or the BBC Book club RSS feed.

Development requirement

Requires the development of an appropriate mechanism to retrieve details of lists from external sources, match these against books held by the library, and then display the results of this 'mashup.'

The exact mechanism could vary considerably; for example, this might be done through an extension to an existing content management system used for publishing content on the web.

API Requirements

- Ability to carry out basic searching (ISBN, author/title) to retrieve bibliographic records
- Ability to retrieve item records based on the bibliographic record they are attached to

Data Requirements

- Simple metadata for bibliographic resources
- Item location details
- Item availability details (e.g. on shelf, on loan, due date)

Skills required

- Programming skills
- Experience working with RESTful APIs
- Knowledge of JSON format

Sustainability considerations

Dependency on external data sources means it is likely that ongoing development and maintenance will be required to keep applications working and to ensure they work if external data sources make any changes to how they work.

CHECKING EFFICIENTLY ON AVAILABILITY

... through a simple mobile phone or tablet App before the patron visits the library or scours the shelves

Solution benefits and opportunities

Efficiency through automation: Just-in-time information
Data flow between systems: ILS to Mobile App
New functions & workflows: Potentially notify items missing from shelf
Enabling others to build on LMS: Library team or third-party application developer

Other Systems Involved

No other systems involved.

Development requirement

Requires the development of an appropriate 'app' for the relevant platform (e.g. iOS, Android), which could use the API to find items and report on their location and current availability.

API Requirements

- Ability to carry out basic searching (ISBN, author/title) to retrieve bibliographic records
- Ability to retrieve item records based on the bibliographic record they are attached to

Data Requirements

- Simple metadata for bibliographic resources
- Item location details
- Item availability details (e.g. on shelf, on loan, due date)

Skills required

- Programming skills
- Experience working with RESTful APIs
- Knowledge of JSON format
- Experience or knowledge of the relevant mobile/tablet development environments (iOS, Android, etc.) The knowledge needed will depend on the mobile/tablet platforms to be supported.

Sustainability considerations

Smartphone and Tablet operating systems are subject to ongoing and sometimes rapid change. It is likely that ongoing development and maintenance will be required to keep applications working and to ensure they work on new devices and operating systems as they are released.

In addition, any changes to the API is likely to necessitate changes to the mobile/tablet application.

CROSS-CHECKING THE LIBRARY CATALOG BEFORE PURCHASING

... from online stores such as Amazon, enabled through a browser extension

Solution benefits and opportunities

Efficiency through automation: Avoid multiple queries
Data flow between systems: ILS to Browser
New functions & workflows: Extend to include reservation
Enabling others to build on LMS: Library team or third-party application developer

Other Systems Involved

Browser (e.g. Firefox, Chrome)

Development requirement

Requires the development of a browser extension, which could use the API to carry out simple searches for items in the library.

API Requirements

- Ability to carry out basic searching (ISBN, author/title) to retrieve bibliographic records
- Ability to retrieve item records based on the bibliographic record they are attached to

Data Requirements

- Simple metadata for bibliographic resources
- Item location details
- Item availability details (e.g. on shelf, on loan, due date)

Skills required

- Programming skills
- Experience working with RESTful APIs
- Knowledge of JSON format
- Experience or knowledge of the relevant browser extension mechanisms. These vary between browsers, so required knowledge depends on the supported browsers.

Sustainability considerations

Browsers are subject to ongoing change, although the extension mechanisms used are unlikely to change as rapidly as the overall browser. It is likely that a small amount of ongoing development and maintenance will be required to troubleshoot issues reported by patrons and to ensure that the browser extensions continue to work.

In addition, any changes to the API are likely to necessitate changes to the browser extension.

PROVIDING A SERVICE TO UPDATE PATRONS ON THE LATEST ADDITIONS TO STOCK

... through a range of channels, such as RSS Feeds, Twitter, the library website, and SMS Text alerts

Solution benefits and opportunities

Efficiency through automation: Structure to push alert to choice of channels
Data flow between systems: ILS to RSS or Twitter, etc
New functions & workflows: Aligning the service with social media
Enabling others to build on LMS: The library team

Other Systems Involved

Other systems involved may depend on desired channels for the service. For example, social networks such as Twitter or Facebook may be involved.

Development requirement

Requires the development of an application that could use the API to carry out topic-based searches for materials in the library.

The application would need to be able to store details of a patron, the topic areas they wish to track, and the channels by which they want to hear about new items relevant to them.

The application would need to be able to post to the desired channels (e.g. RSS, Twitter, Face-book).

Finally, the application would require a user interface for the patron to manage their alerts (e.g. search terms used, channels to be used for notification, frequency of alerts).

API Requirements

- Ability to carry out searching by subject heading or classification to retrieve bibliographic records
- Ability to retrieve item records based on the bibliographic record they are attached to

Data Requirements

- Simple metadata for bibliographic resources
- Item location details

Skills required

- Programming skills
- Experience working with RESTful APIs
- Knowledge of JSON format
- Experience or knowledge of developing with the relevant distribution channels. The knowledge needed will depend on the channels to be supported.

Sustainability considerations

Some channels for distribution (e.g. RSS, email) are likely to be very stable, and a service using such channels is likely to need only minimal maintenance.

Other channels (e.g. Twitter, Facebook) are subject to change, and ongoing development and maintenance will be required to troubleshoot issues reported by patrons and to ensure that the channels continue to work.

In addition, any changes to the API are likely to necessitate changes to the service.

EXTENDING THE MANAGEMENT DASHBOARD

... through visualization tools (such as Many Eyes) to track collection development activity

Solution benefits and opportunities

Efficiency through automation: Data format suited to range of tools
Data flow between systems: ILS to visualization tool
New functions & workflows: None
Enabling others to build on LMS: The library team

Other Systems Involved

Any external products being used to analyze or visualize management information.

Development requirement

Requires the development of an application that could retrieve and store information related to collection development.

If an external product is being used to analyze or visualize management information, the application would need to output data in a format that could be consumed by the external application.

For example, the application could query for item records added during a date range and store this number in conjunction with a call number to allow analysis of how quickly a specific area of library stock is growing.

API Requirements

- Ability to retrieve bibliographic records based on date added, date updated, or date deleted
- Ability to carry out searching by subject heading or classification to retrieve bibliographic records
- Ability to retrieve item records based on date added, date updated, or date deleted
- Ability to retrieve item records based on the bibliographic record they are attached to

EXTENDING THE MANAGEMENT DASHBOARD (CONT.)

Data Requirements

- Simple metadata for bibliographic resources
- Counts of the number of items attached to a bibliographic record

Skills required

- Programming skills
- Experience working with RESTful APIs
- Knowledge of JSON format
- Experience developing database backed applications
- Experience or knowledge of the relevant visualization tools. The knowledge needed will depend on the visualization tools to be supported.

Sustainability considerations

An application that extracts basic reporting information can be expected to work with minimal maintenance.

Changes to any of the systems involved may necessitate changes to the connectors between the systems.

“We are excited to work closely with Innovative on APIs for Sierra that will improve the patron’s experience through EDS, the EBSCO Discovery Service. Our initial integration effort will leverage the Bibliographic and Item APIs to streamline catalog feeds into the EDS database, eliminating library staff manual effort. We are also collaborating with Innovative on a more seamless patron experience in the near future.”

Harry Kaplanian – Senior Director of Product Management, EBSCO

AUTOMATICALLY CHECK A VENDOR EBOOK PACKAGE AGAINST LOCAL STOCK INFORMATION

... against local stock information

Solution benefits and opportunities

Efficiency through automation: Avoid manual stock checking; avoid purchase of duplicate items
Data flow between systems: ILS to an external application
New functions & workflows: None
Enabling others to build on LMS: Library team

Other Systems Involved

Vendor systems or data sources (e.g. Excel Spreadsheets listing eBook package content).

Development requirement

Requires the development of an application which could use the API to carry out simple searches for items in the library.

API Requirements

- Ability to carry out basic searching (ISBN, author/title) to retrieve bibliographic records
- Ability to retrieve item records based on the bibliographic record they are attached to

Data Requirements

- Simple metadata for bibliographic resources
- Counts of the number of items attached to a bibliographic record

Skills required

- Programming skills
- Experience working with RESTful APIs
- Knowledge of JSON format

Sustainability considerations

Dependency on external data sources means it is likely that ongoing development and maintenance will be required to keep applications working and to ensure they work if the external data sources make any changes to how they work.

Where data is made available in a standard format (e.g. CSV files, Excel Spreadsheets), there is unlikely to be the need for frequent changes to any application.

GOING FURTHER

The user stories above deliberately focus on APIs that support getting bibliographic and item data out of the ILS. The addition of APIs that give access to other data and services would allow further user stories to be addressed.

For example, providing access to Patron data and Circulation services via an API could offer enhanced integration with discovery layers (as described in use case 1 above). This could enable patrons to create hold requests and manage their library account directly from a discovery layer or from another integrated system such as a mobile application (as described in use case 5 above).

Patron and Circulation APIs could also enable automatic holds to be made where a user has indicated an interest in a particular author or if they belong to a book club (extending the stories in use cases 3 and 4 above). API access to a 'renew' function could offer the ability to automatically renew items on behalf of a patron, and access to fine payment functions would open up the possibility of

integration with local online payment systems where they are available.

The addition of API mechanisms to update data within the ILS would create further opportunities - from updating or deleting item records to modifying bibliographic records, creating lists, updating authority records, and much more.

As outlined in the June 2014 White Paper '**No Library is an Island: How APIs can empower libraries in an evolving service ecosystem,**' with the right skills and imagination, libraries could take advantage of APIs to deliver real business benefits. The user stories described here are only the beginning.

David Kay and Owen Stephens are at Sero Consulting, a UK-based consultancy focused on library software technology. Steve Schoen is the API Product Manager at Innovative, a global library automation provider.

For more information:

Web: www.iii.com

Email: info@iii.com

Call: 800.878.6600



facebook.com/innovativeinterfaces



twitter.com/iii_Innovative