

Alma and Primo Analytics Perl REST API

**IGeLU Developers Day. 8 Sep 2016.
David Lewis, Curtin University Library.**

What you need to know

- You understand general programming concepts (eg parse trees)
- You are familiar with debugging
- You program in Perl
- You are familiar with Perl paradigms (eg regex) and modules (eg DBI)
- You know how to find and install Perl modules
- You already know how to create a BI Analysis and what are filters
- You are familiar with the Analysis editor

SQL select and Analytics

- BI Analysis = select statement
- Queries and result sets
 - Rows and named columns
 - Select mms_id, title,

MMS ID	Title	Created	Modified	Creator	Modifier	Link
9921232400 01951	Introduction	2015-01-01	2015-12-31	Me	Me	http://abc.org/video?v=xZ1i_Lktv
.
.
.

Perl DBI

```
my $dbh = DBI->connect("DBI:oracle:...", $uid, $pwd);  
  
my $qs = "select mms_id,title from bibrec where title like ?";  
my $sth = $dbh->prepare($qs);  
  
$sth->bind_param(1, "Introduction%");  
  
$sth->execute;  
while (my $row = $sth->fetchrow_hashref) {  
    print $row->{MMS_ID} . ", " . $row->{TITLE} . "\n";  
}  
  
$sth->finish;  
$dbh->disconnect;
```

← create DBI handle

← prepare SQL statement handle

← set query parameter

← execute and process

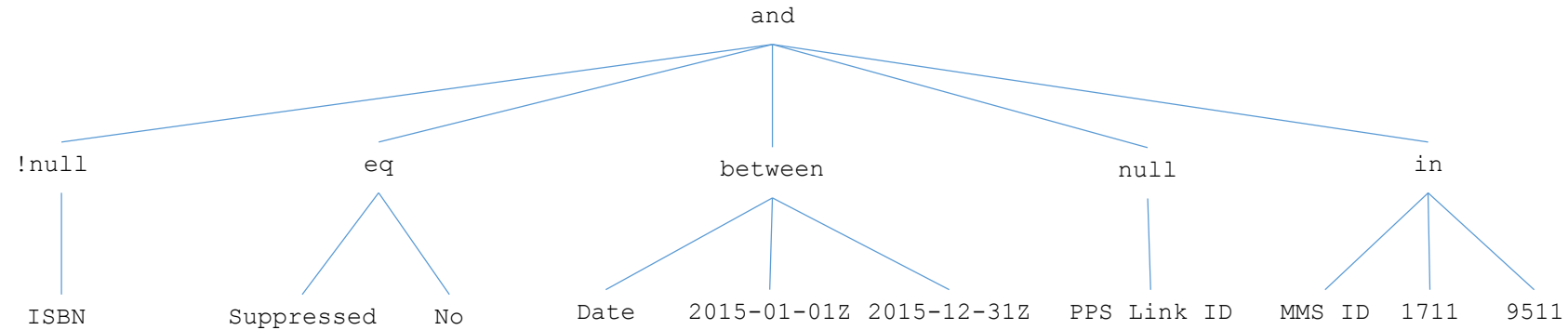
Raw Analytics REST API Programming

- Create REST client connection
- Prepare XML filter
- Create REST API GET URL
 - /almaws/v1/analytics?...
- Add API key to HTTP request header (or URL)
- Execute analysis via HTTP GET
- Retrieve XML response
- Parse response
- Check for errors
- Cycle through rows and process
- Work with continuation token (if any)

Analytics Filters

```
<sawx:expr xsi:type="sawx:logical"
  op="and"
  xmlns:saw="com.siebel.analytics.web/report/v1.1"
  xmlns:sawx="com.siebel.analytics.web/expression/v1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <sawx:expr xsi:type="sawx:comparison" op="notNull">
    <sawx:expr xsi:type="sawx:sqlExpression">"Physical Items"."Bibliographic Details"."ISBN"</sawx:expr>
  </sawx:expr>
  <sawx:expr xsi:type="sawx:comparison" op="equal">
    <sawx:expr xsi:type="sawx:sqlExpression">"Bibliographic Details"."Suppressed From Discovery"</sawx:expr>
    <sawx:expr xsi:type="xsd:string">No</sawx:expr>
  </sawx:expr>
  <sawx:expr xsi:type="sawx:comparison" op="between">
    <sawx:expr xsi:type="sawx:sqlExpression">"Bibliographic Details"."Modification Date"</sawx:expr>
    <sawx:expr xsi:type="xsd:date">2015-01-01Z</sawx:expr>
    <sawx:expr xsi:type="xsd:date">2015-12-31Z</sawx:expr>
  </sawx:expr>
  <sawx:expr xsi:type="sawx:comparison" op="null">
    <sawx:expr xsi:type="sawx:sqlExpression">"Portfolio"."Pps link id"</sawx:expr>
  </sawx:expr>
  <sawx:expr xsi:type="sawx:list" op="in">
    <sawx:expr xsi:type="sawx:sqlExpression">"Bibliographic Details"."MMS ID"</sawx:expr>
    <sawx:expr xsi:type="xsd:integer">1711</sawx:expr>
    <sawx:expr xsi:type="xsd:integer">9511</sawx:expr>
  </sawx:expr>
</sawx:expr>
```

Parse Trees and Simplified Filters



```
and (
  !null ( "Physical Items"."Bibliographic Details"."ISBN" ) ,
  eq ( "Bibliographic Details"."Suppressed From Discovery" , "No" ) ,
  between ( "Bibliographic Details"."Modification Date" , "2015-01-01Z" , "2015-12-31Z" ) ,
  null ( "Portfolio"."Pps link ID" ) ,
  in ( "Bibliographic Details"."MMS ID" , [ "1711" , "9511" ] )
)
```

Perl REST API (Alma and Primo)

```
my $conf = Alma2::Config::YAML->new(Alma2::Config::YML_FILE);
my $dbh = Alma2::Analytics::Client->connect($conf->get('production.analytics'));
```

} get connection

```
my $types = [ 'string' ];
my $sfilter = 'eq ( "E-Inventory"."Bibliographic Details"."Title" , ? )';
my $filter = Alma2::Analytics::Filter->new();
$filter->initialize($sfilter, $types);
```

} create filter generator

```
my $title = "Introduction";
my $xml = $filter->compile([ $title ]);
```

} compile xml filter

```
my $columns = [ 'MMS_ID', 'TITLE' ];
my $qs = '/shared/My University/Reports/EUS/MMSID_TITLE';
my $sth = $dbh->prepare($qs, $columns);
$sth->bind_param($xml);
```

} prepare query

```
$sth->execute();
while ( my $row = $sth->fetchrow_hashref() ) {
    print $row->{MMS_ID}.'.', '$row->{TITLE}."\n";
}

```

} execute, process

```
$sth->finish();
$dbh->disconnect();
```


Perl REST API (ideal)

```
my $conf = Alma2::Config::YAML->new(Alma2::Config::YML_FILE);
my $dbh = Alma2::Analytics::Client->connect($conf->get('production.analytics')); } get
connection

my $sth = $dbh->prepare(
    query => '/shared/My University/Reports/EUS/MMSID_TITLE',
    columns => [ 'MMS_ID', 'TITLE' ],
    filter => 'eq ( "E-Inventory"."Bibliographic Details"."Title", ? )',
    type => [ 'string' ]
);

$sth->bind_param(1, "Introduction"); } set query
values

$sth->execute();
while (my $row = $sth->fetchrow_hashref()) { } execute and
    print $row->{MMS_ID}."", ".$row->{TITLE}."\n"; process results
}

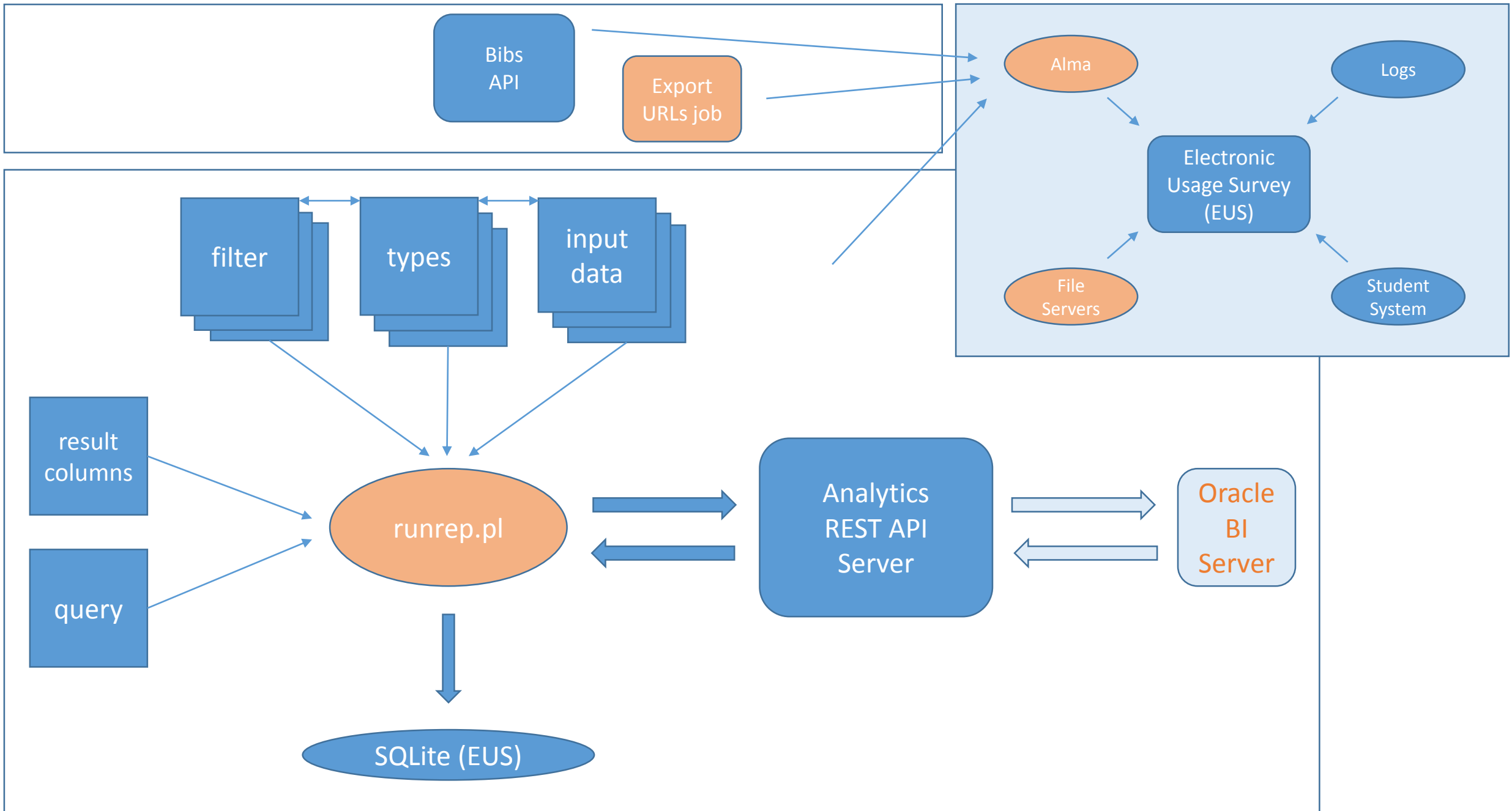
$sth->finish();
$dbh->disconnect();
```

The Modules

- Supporting utilities and classes
 - Struct/Skywalker, XPC, Error, Logger
 - Util: Core, DB, DateTime
 - Mail: Unix mailer
 - Config: Configuration file .yml
- REST client
 - Transport layer
 - Connection, Timeout, Retries
 - URI Escaping
 - Error Handling
- Analytics Client
 - High level DBI-style interface
 - Client, Statement, RowStruct, Filter, Parser, DataWriter

runrep.pl

- Analytics Export Limitations
 - 64,000+ rows
- Batch execution script
 - Unlimited results (in theory)
 - Execute any analysis you create
 - Job
 - Batch up inputs (options: query, filters+types+data , columns)
 - runrep.pl run job
 - Output target file (.csv .xls .xlsx SQLite)
 - runrep.pl save job
- Automation
 - Merge Alma data with other Library and University data sources
 - Copyright compliance reporting (Electronic Usage Survey)



Download. Use.

- Bit Bucket
 - Git repository
 - Tar file
 - GPL
 - Repository : <https://bitbucket.org/curtin-library/almaanalytics>
- Read the instructions
- Email
 - d.lewis@curtin.edu.au
 - Bugs, comments, suggestions, help
 - Subject: Perl Analytics API : Bug | Comment | Suggestion | Help
 - Include as much detail as possible
 - Explain what you are trying to do. Code snippets if possible.
 - Error messages, log files
 - Operating system and version
 - Perl version and environment (Perlbrew?)
 - Any other details
 - Fixes are welcome if you have any and would like to contribute

Thank you!