

# Running the API Gauntlet

Getting Over a Quarter of a Million Patron Records Out of and Into Alma

University of Cambridge Libraries

Lesley Gray, Deputy Director – Digital Initiatives and Strategy

# INTRODUCTION

# libraries@cambridge: the collegiate university

- About 100 libraries across the University and colleges
  - Cambridge University Libraries (University Library and 23 department/faculty libraries)
  - 31 department & faculty libraries
  - 31 college libraries
  - 13 associated institution libraries
- 93 libraries use Alma

# Voyager to Alma

Voyager

10 Voyager databases

Library/database specific policies

Duplicate bibliographic records

Multiple vendor records

Multiple patron records



Alma

Single institution in Alma

Institutional policies

Single bibliographic record

Single vendor records

Single patron records

Oladeji Famakinwa, Head of Digital Services – Library and Web

# **PATRON DATA CHALLENGES**

# Voyager data architecture

- Ten voyager databases
- Custom clustering and universal borrowing (UB) setup
- Local library patron files and circulation policies
  - home, stub and standalone patron records

# Alma patron data migration

- Took several iterations to get to a 'good enough' state
- Unfortunately patrons:
  - did not have their library associations
  - user groups were not set properly (from 642 to 37)
  - unable to log into Primo or Alma

# Choice of approach

- Why?
- Get data out
  - SIS export – convenient and available out of the box!
- Change the data
- Get data in
  - User API
    - Reliably change every field of every record
    - Needed to rearrange primary identifiers
    - We were going to use the API for management anyway



James Howe, Senior Software Developer

# WORKING WITH THE USER API

# Process overview

## January 2018

- Before release
- Export 305,810 records
- Modify 188,058 records

## June 2018

- Live system
- Export 308,781 records
- Modify 181,890 records
- Delete 126,728 records

# Process overview

- Export users from Alma
  - Standard SIS FTP user export
- Convert XML to JSON stream
  - Code-generated from XSD schema
- Transform and partition the records
  - Pipeline of python scripts
- Write changes back to Alma
  - Python asyncio tool calling the API

# Problem #1 – Inaccessible Records

- Duplicate identifiers
  - 568 duplicate primary identifiers, affecting over a thousand records
  - API behaviour inconsistent
- Missing identifiers
  - 160 records with no identifiers at all
  - Impossible to interact via API
- Solution – lots of clicking
  - The Alma UI was perfectly capable of editing and deleting these records

# Problem #2 – Inconsistent identifier uniqueness

- Duplicate inactive additional identifiers can be added
- Until one of them becomes active
- Any API update that doesn't remove the inactive ids is rejected
- Cannot search for inactive identifiers
- Solution
  - Multiple passes to find all the duplicates
  - Retry failed updates at the end

# Problem #3 – UI lets you create duplicate identifiers

- Create user with primary identifier “ONE”
- Add additional identifier “TWO”
- Deactivate the “TWO” identifier
- Create second user with primary identifier “TWO”
- Activate the “TWO” identifier on the first user
- You now have Problem #1 and the API will do horrible things

# Problem #4 – Cannot move primary id in one call

```
{  
  "primary_id": "ONE",  
  "user_identifier": []  
}
```



```
{  
  "primary_id": "TWO",  
  "user_identifier": [{  
    "id_type": {"value": "01"},  
    "value": "ONE",  
    "status": "ACTIVE"  
  }]  
}
```



```
{  
  "primary_id": "THREE",  
  "user_identifier": []  
}
```

# Problem #5 – Data does not conform to XSD schema

```
<contact_info>  
  <addresses/>  
  <emails/>  
  <phones/>  
</contact_info>
```

```
<xs:complexType name="emails">  
  <xs:sequence>  
    <xs:element name="email" type="email" minOccurs="1"  
                maxOccurs="unbounded">  
      </xs:element>  
    </xs:sequence>  
</xs:complexType>
```



# Problem #6 – UI saves invalid data

- Particular edits in the Alma UI can "lock out" the API
  - That is, a GET followed by an unchanged PUT will be rejected as invalid
- Bulk adding roles
  - Saved as inactive without scope or parameters
- Solution
  - Remove any items that are missing required attributes
  - Special-case known invalid values

# Problem #7 – Country code table is not enforced

- Unlike other coded fields the country is not validated
- Cannot tell whether the country will actually be displayed
- Fulfilment UI can set the country code to "Null"
- Solution
  - Use the ISO 3166-1 codes
  - Extra verification outside of Alma

# Problem #8 – More primary identifier issues

- Generated UUIDs for the first migration
  - 9eb55a2c-35c9-4b53-999f-0529de2553d5
- One of the Create User forms doesn't generate ids for you
- Our payment integration had a 30-char max for identifiers
- Sometimes Alma displays the id instead of the user's name
- Solution – change them all again
  - Re-encoded the UUIDs to be shorter
  - T22VU-LBVZFFVH-GM7AUU54-JKT2U

# In Summary

- Identifier system is flawed
- Alma UI doesn't play by the rules

# API Recommendations

- Allow specifying `primary_id` for `user_id_type`
- Provide some edit serialisation method
  - A revision identifier (hash of record, incrementing number)
  - Add an `ETag` header to GET responses
  - Include in the record (for export, search and webhooks)
  - Support `If-Match` et al. on API requests
- Allow PUT to create records
- Use the XSD schema internally
- Offer JSON-stream user exports

# QUESTIONS