Mobile Self-Checkout for Customers

Empowering Customers with Polaris Technology

IUG Virtual Conference 2022

Mark Jarrell

ONLINE APPLICATIONS DEVELOPER, Richland Library, Columbia, SC

mjarrell@richlandlibrary.com

Intro

- Case study - something we built for our customers using the Leap API that's included with Polaris
- Motivations for why & how we built this
- Statistics
- Technologies used
- Demo of the self-checkout

Background

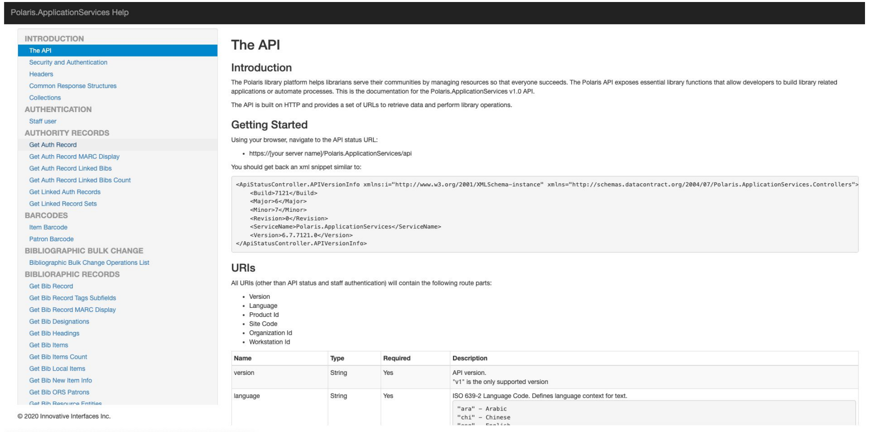
- Richland Library went “fine free” in 2019
- COVID-19 pandemic 2020
- Customer looking for a way to do a “touchless” checkout
- Changed name to “mobile self-checkout”
- Immediately check out materials using their own device
- Scan the barcodes of the items they want to check out
- RFID tags still in use, but not for security of items at gates

Statistics

- 3,373 item checkouts in 2021 (281 per month)
- 437 customers
- Context: 836,326 total checkouts of physical items at library locations in 2021

Technologies and APIs used

- Web based documentation for Leap API: https://<leap.yourlibrary.org>/polaris.applicationservices/help



The API

Introduction

The Polaris library platform helps librarians serve their communities by managing resources so that everyone succeeds. The Polaris API exposes essential library functions that allow developers to build library related applications or automate processes. This is the documentation for the Polaris.ApplicationServices v1.0 API.

The API is built on HTTP and provides a set of URLs to retrieve data and perform library operations.

Getting Started

Using your browser, navigate to the API status URL:

- https://[your server name]/Polaris.ApplicationServices/api

You should get back an xml snippet similar to:

```
<ApiStatusController.APIVersionInfo xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/Polaris.ApplicationServices.Controllers">
    <Build>7121</Build>
    <Major>6</Major>
    <Minor>7</Minor>
    <Revision>0</Revision>
    <ServiceName>Polaris.ApplicationServices</ServiceName>
    <Version>6.7.7121.0</Version>
</ApiStatusController.APIVersionInfo>
```

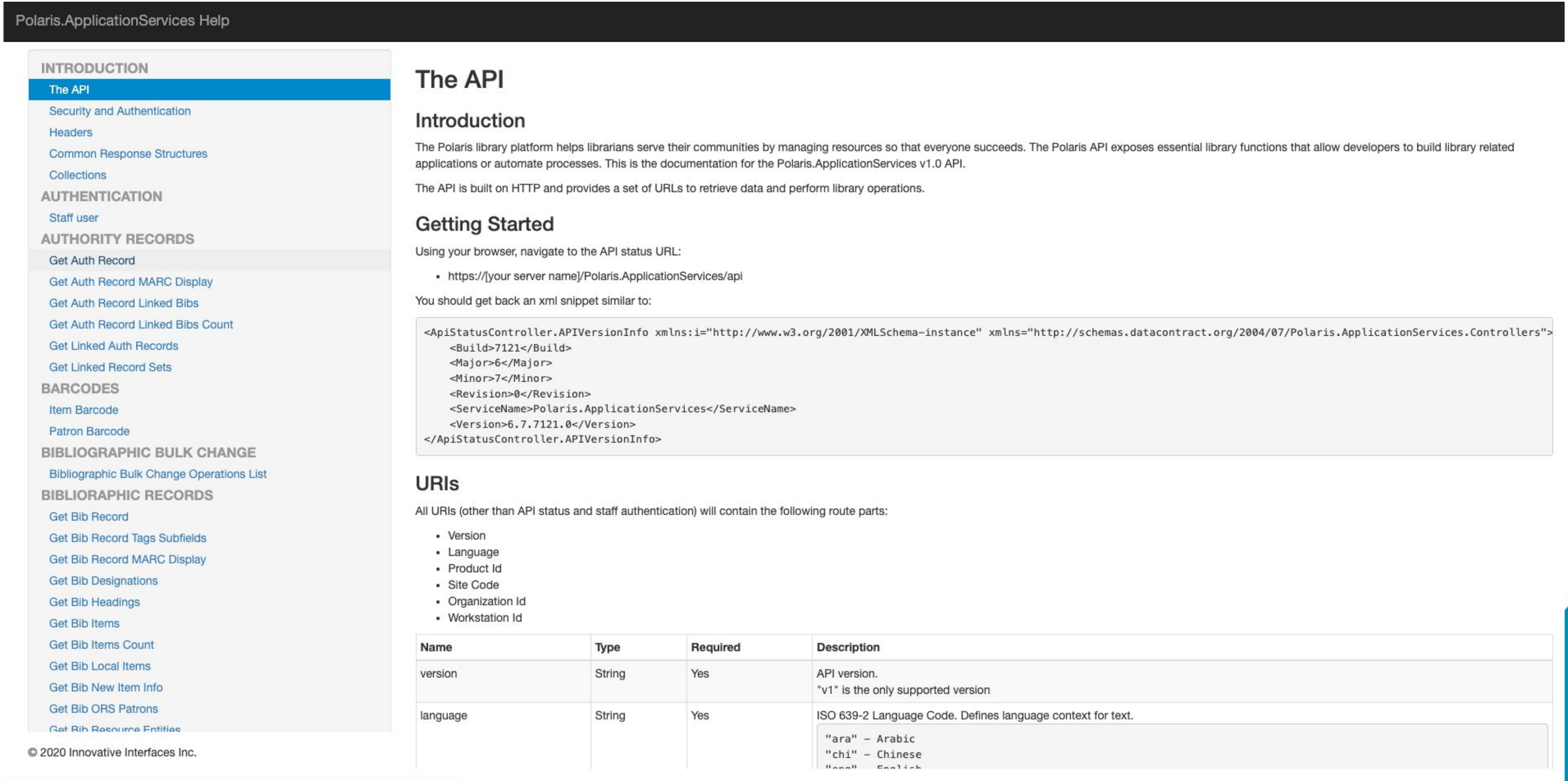
URIs

All URIs (other than API status and staff authentication) will contain the following route parts:

- Version
- Language
- Product Id
- Site Code
- Organization Id
- Workstation Id

Name	Type	Required	Description
version	String	Yes	API version. "v1" is the only supported version
language	String	Yes	ISO 639-2 Language Code. Defines language context for text. "ara" - Arabic "chi" - Chinese "eng" - English

© 2020 Innovative Interfaces Inc.



Polaris.ApplicationServices Help

INTRODUCTION
- The API
- Security and Authentication
- Headers
- Common Response Structures
- Collections

AUTHENTICATION
- Staff user

AUTHORITY RECORDS
- Get Auth Record
- Get Auth Record MARC Display
- Get Auth Record Linked Bibs
- Get Auth Record Linked Bibs Count
- Get Linked Auth Records
- Get Linked Record Sets

BARCODES
- Item Barcode
- Patron Barcode

BIBLIOGRAPHIC BULK CHANGE
- Bibliographic Bulk Change Operations List

BIBLIORAPHIC RECORDS
- Get Bib Record
- Get Bib Record Tags Subfields
- Get Bib Record MARC Display
- Get Bib Designations
- Get Bib Headings
- Get Bib Items
- Get Bib Items Count
- Get Bib Local Items
- Get Bib New Item Info
- Get Bib ORS Patrons
- Get Bib Resource Entities

© 2020 Innovative Interfaces Inc.

The API

Introduction

The Polaris library platform helps librarians serve their communities by managing resources so that everyone succeeds. The Polaris API exposes essential library functions that allow developers to build library related applications or automate processes. This is the documentation for the Polaris.ApplicationServices v1.0 API.

The API is built on HTTP and provides a set of URLs to retrieve data and perform library operations.

Getting Started

Using your browser, navigate to the API status URL:

- https://[your server name]/Polaris.ApplicationServices/api

You should get back an xml snippet similar to:

```
<ApiStatusController.APIVersionInfo xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/Polaris.ApplicationServices.Controllers">
    <Build>7121</Build>
    <Major>6</Major>
    <Minor>7</Minor>
    <Revision>0</Revision>
    <ServiceName>Polaris.ApplicationServices</ServiceName>
    <Version>6.7.7121.0</Version>
</ApiStatusController.APIVersionInfo>
```

URIs

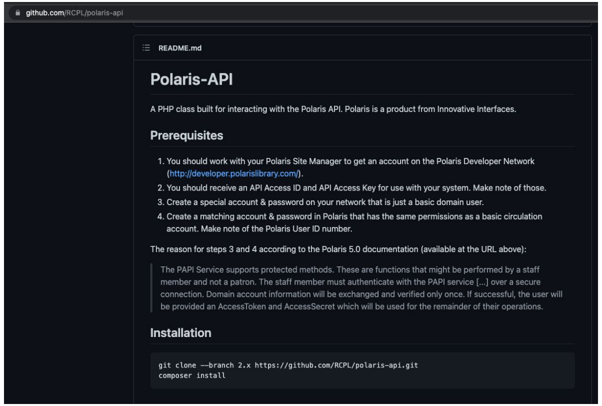
All URIs (other than API status and staff authentication) will contain the following route parts:

- Version
- Language
- Product Id
- Site Code
- Organization Id
- Workstation Id

Name	Type	Required	Description
version	String	Yes	API version. "v1" is the only supported version
language	String	Yes	ISO 639-2 Language Code. Defines language context for text. "ara" – Arabic "chi" – Chinese "eng" – English

Technologies and APIs used

- The main portion of this that's contributed is on our polaris-api Github project: https://github.com/RCPL/polaris-api



Polaris-API

A PHP class built for interacting with the Polaris API. Polaris is a product from Innovative Interfaces.

Prerequisites

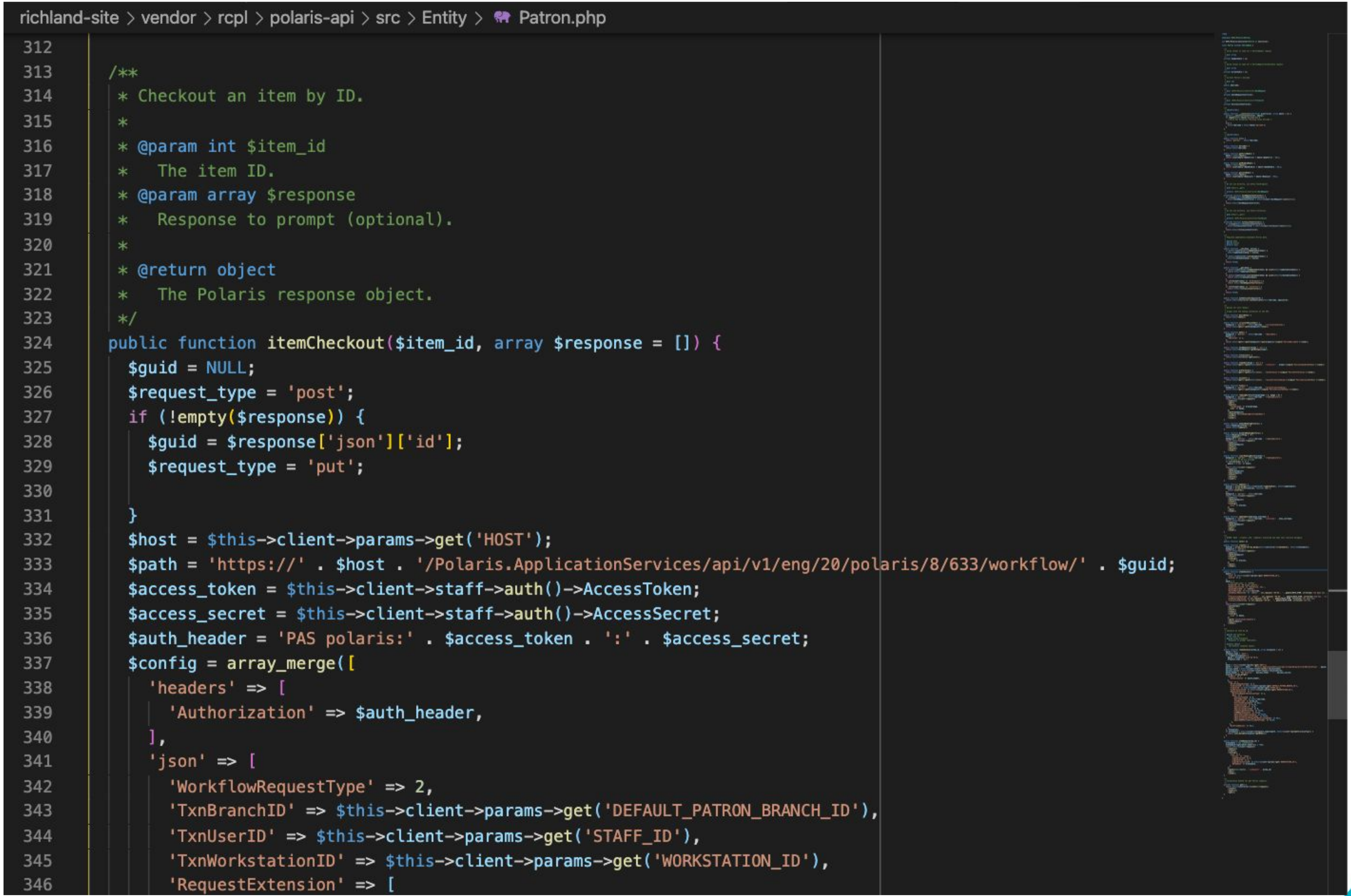
1. You should work with your Polaris Site Manager to get an account on the Polaris Developer Network (http://developer.polarislibrary.com/).
2. You should receive an API Access ID and API Access Key for use with your system. Make note of those.
3. Create a special account & password on your network that is just a basic domain user.
4. Create a matching account & password in Polaris that has the same permissions as a basic circulation account. Make note of the Polaris User ID number.

The reason for steps 3 and 4 according to the Polaris 5.0 documentation (available at the URL above):

> The PAPI Service supports protected methods. These are functions that might be performed by a staff member and not a patron. The staff member must authenticate with the PAPI service [...] over a secure connection. Domain account information will be exchanged and verified only once. If successful, the user will be provided an AccessToken and AccessSecret which will be used for the remainder of their operations.

Installation

```
git clone --branch 2.x https://github.com/RCPL/polaris-api.git
composer install
```



richland-site > vendor > rcpl > polaris-api > src > Entity > Patron.php

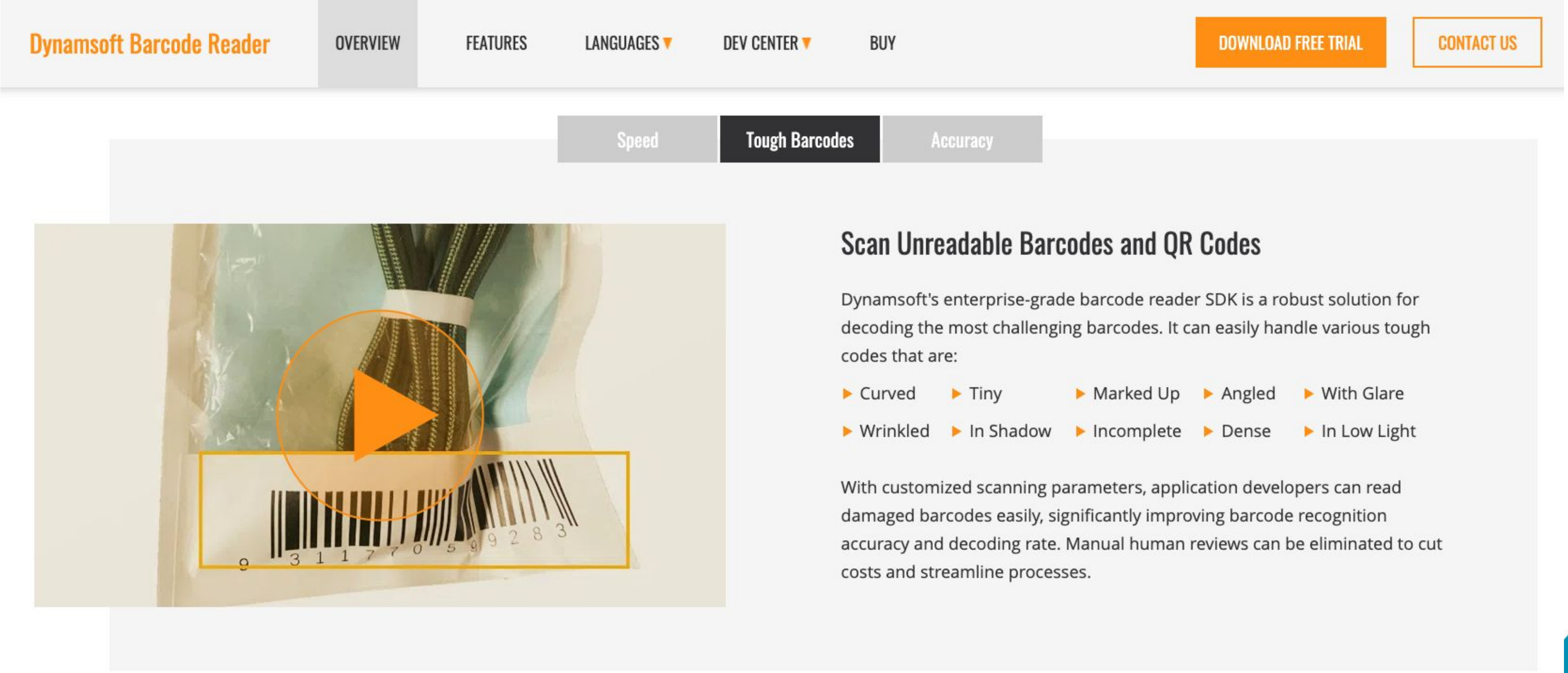
```php

  /**
   * Checkout an item by ID.
   *
   * @param int $item_id
   *   The item ID.
   * @param array $response
   *   Response to prompt (optional).
   *
   * @return object
   *   The Polaris response object.
   */
  public function itemCheckout($item_id, array $response = []) {
    $guid = NULL;
    $request_type = 'post';
    if (!empty($response)) {
      $guid = $response['json']['id'];
      $request_type = 'put';

    }
    $host = $this->client->params->get('HOST');
    $path = 'https://' . $host . '/Polaris.ApplicationServices/api/v1/eng/20/polaris/8/633/workflow/' . $guid;
    $access_token = $this->client->staff->auth()->AccessToken;
    $access_secret = $this->client->staff->auth()->AccessSecret;
    $auth_header = 'PAS polaris:' . $access_token . ':' . $access_secret;
    $config = array_merge([
      'headers' => [
        'Authorization' => $auth_header,
      ],
      'json' => [
        'WorkflowRequestType' => 2,
        'TxnBranchID' => $this->client->params->get('DEFAULT_PATRON_BRANCH_ID'),
        'TxnUserID' => $this->client->params->get('STAFF_ID'),
        'TxnWorkstationID' => $this->client->params->get('WORKSTATION_ID'),
        'RequestExtension' => [
```

Technologies and APIs used

- Drupal website using our Polaris API library with a React-based frontend

Technologies and APIs used

- Dynamsoft Barcode Reader - https://www.dynamsoft.com/barcode-reader/overview/



Scan Unreadable Barcodes and QR Codes

Dynamsoft's enterprise-grade barcode reader SDK is a robust solution for decoding the most challenging barcodes. It can easily handle various tough codes that are:

- Curved
- Tiny
- Marked Up
- Angled
- With Glare
- Wrinkled
- In Shadow
- Incomplete
- Dense
- In Low Light

With customized scanning parameters, application developers can read damaged barcodes easily, significantly improving barcode recognition accuracy and decoding rate. Manual human reviews can be eliminated to cut costs and streamline processes.

Technologies and APIs used

- Dynamsoft Barcode Reader - https://www.dynamsoft.com/barcode-reader/overview/
- Pricing is here: https://www.dynamsoft.com/store/dynamsoft-barcode-reader
- Contact them and ask for a quote on non-profit pricing

Demo!

Future

- Idea Lab 2 recent ideas:
- Improve Leap API documentation: https://idealab.iii.com/apichallenge/Page/ViewIdea?ideaid=23938
- Polaris API: Provide method to check out material in PAPI: https://idealab.iii.com/apichallenge/Page/ViewIdea?ideaid=23874

Tangent!

- Another open source project from Richland Library is available to other Libraries (not just Polaris libraries) at:
- https://libraryintercept.com

intercept

Any Questions?

RICHLAND LIBRARY®