

# 1. What are Alma Cloud Apps?

# What are Alma Cloud Apps?

## Concept:

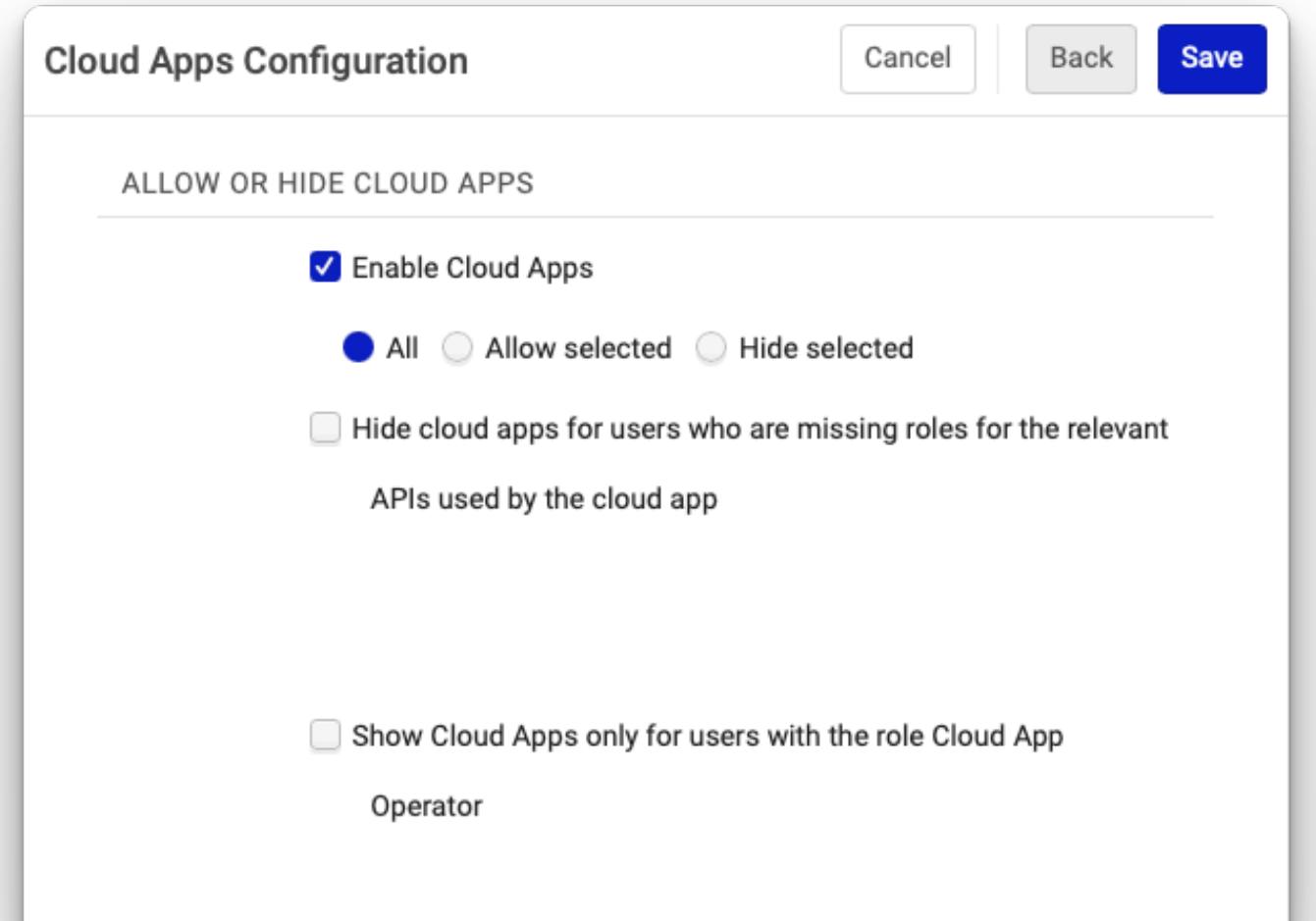
- Custom extensions for Ex Libris Alma platform
- Run directly in a sidebar in Alma
- Extend functionality via **Alma REST API** and external APIs

## Key Benefits:

- Integrated user experience
- Workflow automations & efficiency improvements
- Deploy via Cloud App Store (no separate hosting needed)
- Can be shared across institutions

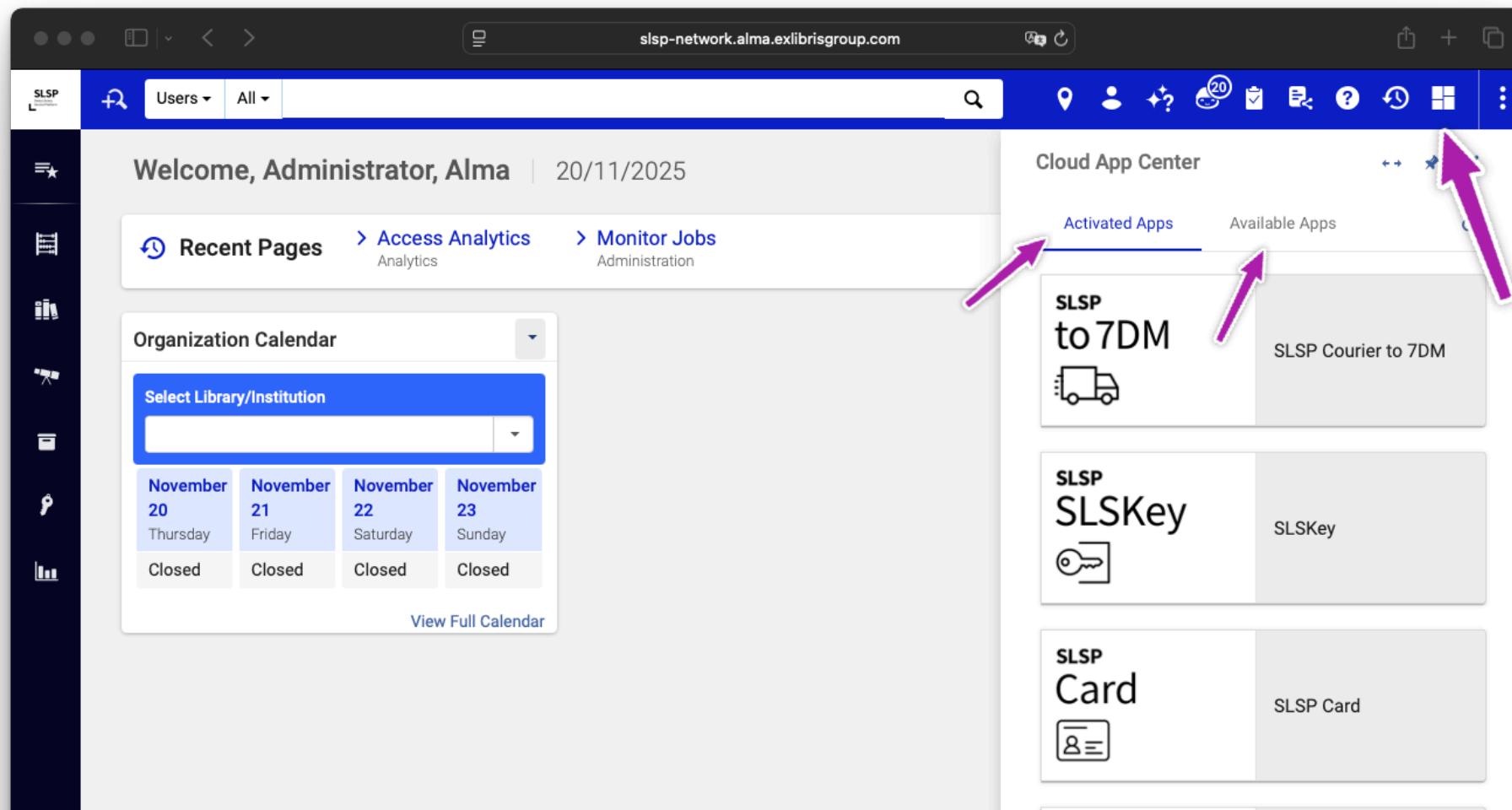
# How to activate and use them?

- Activate them in the Institution Zone via: Configuration > General > Cloud Apps



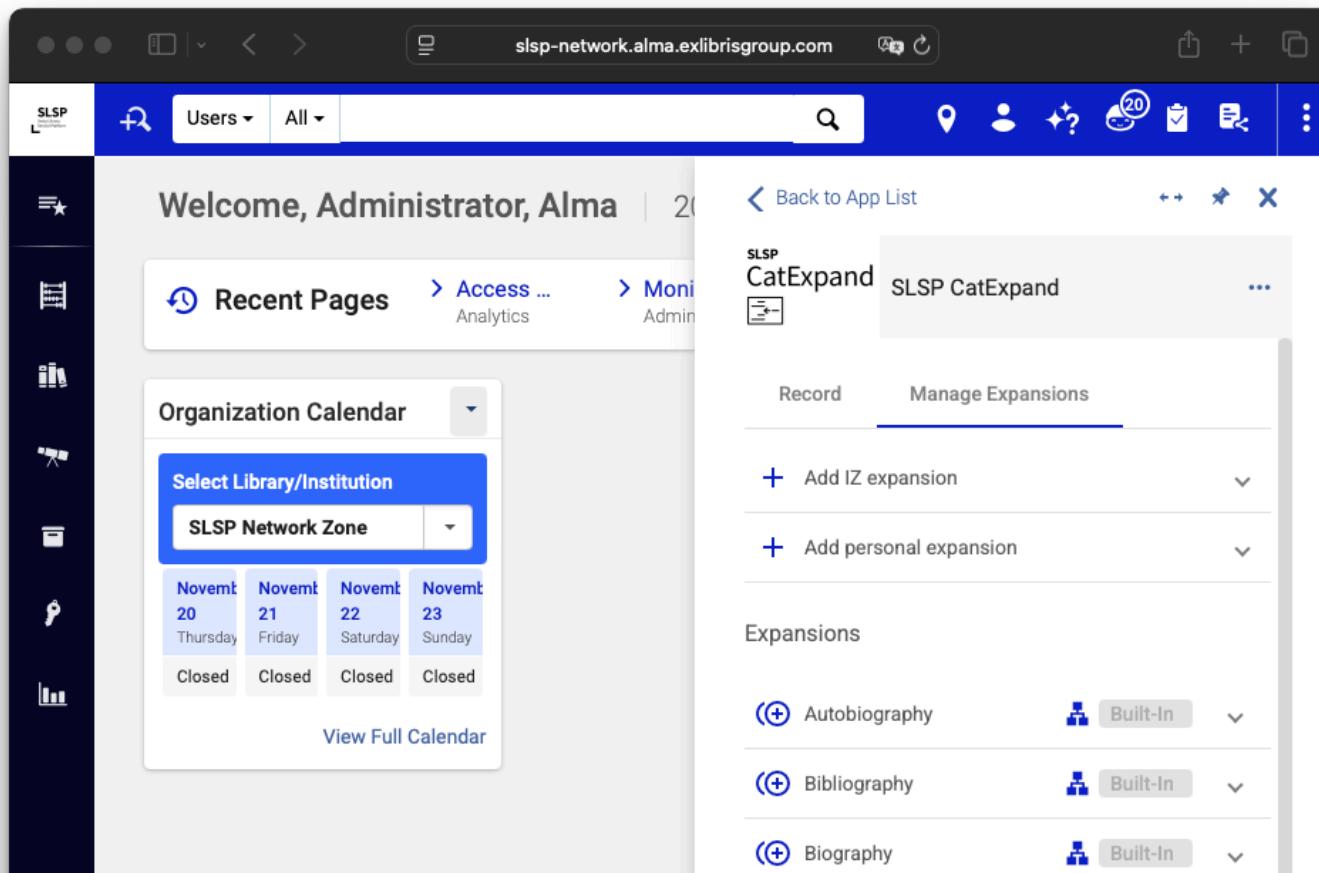
# How to activate and use them?

- Use the Cloud App Store to install and configure apps

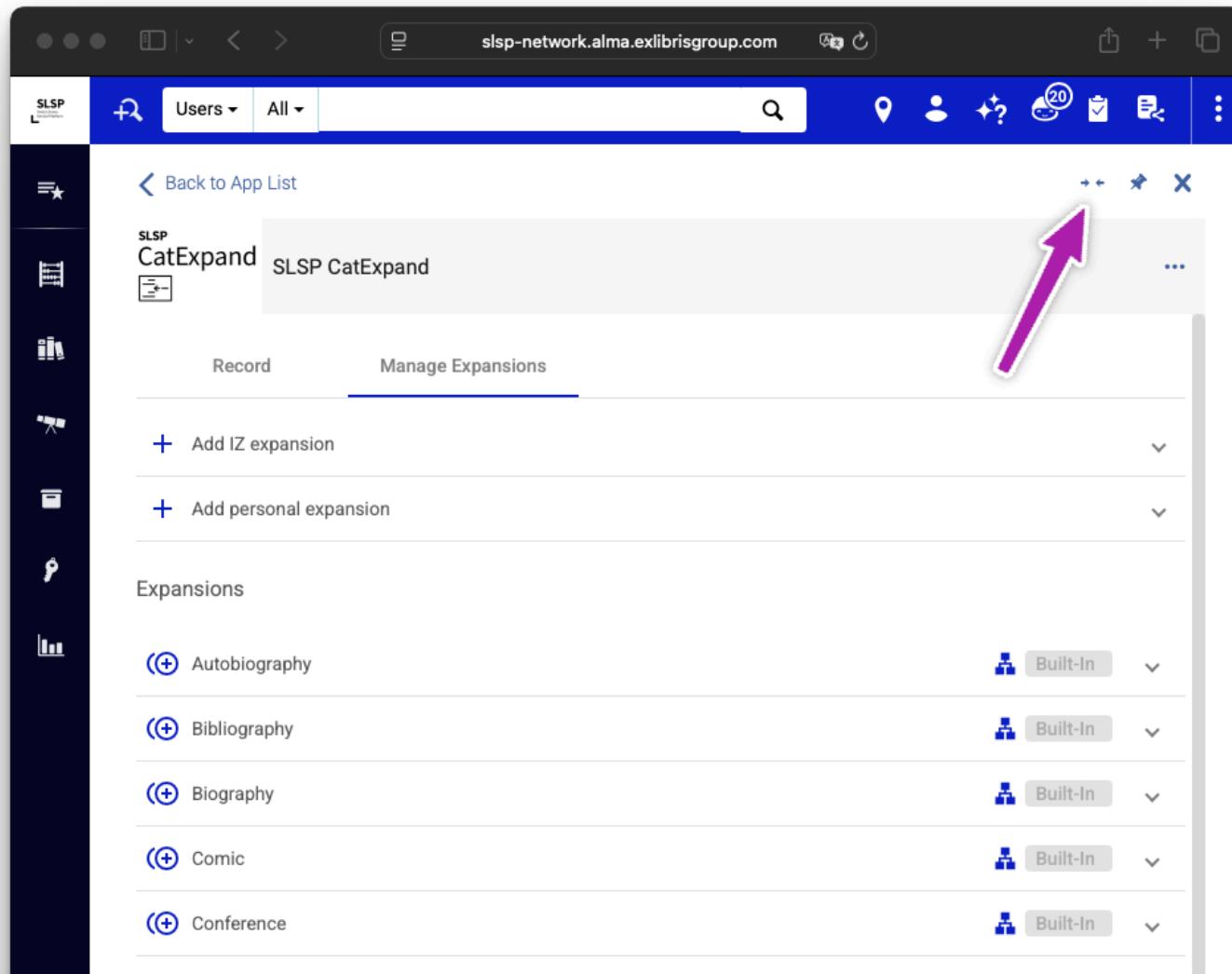


# Two Types of Cloud Apps

- **Full-page apps:** Standalone applications in Alma sidebar
  - sidebar can be resized to full width, if needed



- Full-page apps: Standalone applications in Alma sidebar
  - sidebar can be resized to full width, if needed



- **Dashboard widgets:** Small components on Alma dashboard
  - Quick access to important info or actions

The screenshot shows the Alma dashboard interface. On the left is a vertical sidebar with icons and labels: ExLibris (top), Alma QA, Market, Acquisitions, Resources, Research, Discovery, Fulfillment, Admin, and Analytics (bottom). The main header bar includes the ExLibris logo, search fields for 'All titles' and 'Keywords', and a navigation menu. Below the header, a welcome message reads "Welcome, Implementor, Ex Libris | 19/06/2023". A navigation menu offers links to "Recent Pages", "Add Digital Representation" (under Resource Management), and "Manage Collections" (under Resource Management). A central widget titled "Number of users in each user-group" displays the following data in a table:

Group	Code	Amount
Employee	EMPLOYEE	0
Alumni	ALUMNI	0
Graduate Student	STUDENT_GRADUATE	1923
Undergraduate Student	STUDENT_UNDERGRADUATE	3489
Visitor	GUEST	7
Faculty	FACULTY	454
Staff	STAFF	10

## 2. Technical Foundation & SDK APIs

# Technical Framework

Built on:

- **Angular 18** (HTML + TypeScript)
- **RxJS** for reactive programming, async data streams
- **Cloud App SDK library** ( [@exlibris/exl-cloudapp-angular-lib](https://github.com/exlibris/exl-cloudapp-angular-lib) )

**Key Principle:**

Apps interact with Alma through dedicated SDK services

# Cloud App SDK & CLI

## What is it?

- Official development toolkit for building Alma Cloud Apps
- CLI tool + Angular library ( [@exlibris/exl-cloudapp-angular-lib](#) )
- Provides scaffolding, local dev server, and build tools

## Maintained by:

- Ex Libris Group (official support)
- Open source on GitHub
- Regular updates "twice a year"

We'll use it in the hands-on session!

# Cloud App SDK Services Overview

The SDK provides **6 core services** for interacting with Alma:

1. **Events Service** - Page context & navigation
2. **Settings Service** - User-specific settings
3. **Configuration Service** - Institution-wide configuration per app
4. **Alert Service** - User notifications
5. **Store Service** - Local data storage
6. **REST Service** - Alma API calls

Each service is injected via Angular Dependency Injection

# Events Service

**Purpose:** Access page context and control navigation

**Key Methods:**

- `onPageLoad()` - Subscribe to page changes
- `getInitData()` - Get logged in user info, institution, language
- `entities$` - Observable of current entities
- `refreshPage()` / `home()` / `back()` - Navigation, but limited

**Example:**

```
eventsService.entities$.subscribe(entities => {
    // React to current entities viewed by user (e.g., ITEM, USER)
});
```

# Settings Service

**Purpose:** Store per-user preferences

**Key Methods:**

- `get()` / `set()` / `remove()` - Store user preferences
- Persisted in Alma per user + per app, across sessions/devices

**Examples:**

- UI preferences, favorites, last search, filter settings

# Configuration Service

**Purpose:** Store institution-wide settings

**Key Methods:**

- `get()` / `set()` / `remove()` - Store app configuration
- Only users with admin roles can set, all users can read

**Examples:**

- API keys, default values, feature toggles



# Alert Service

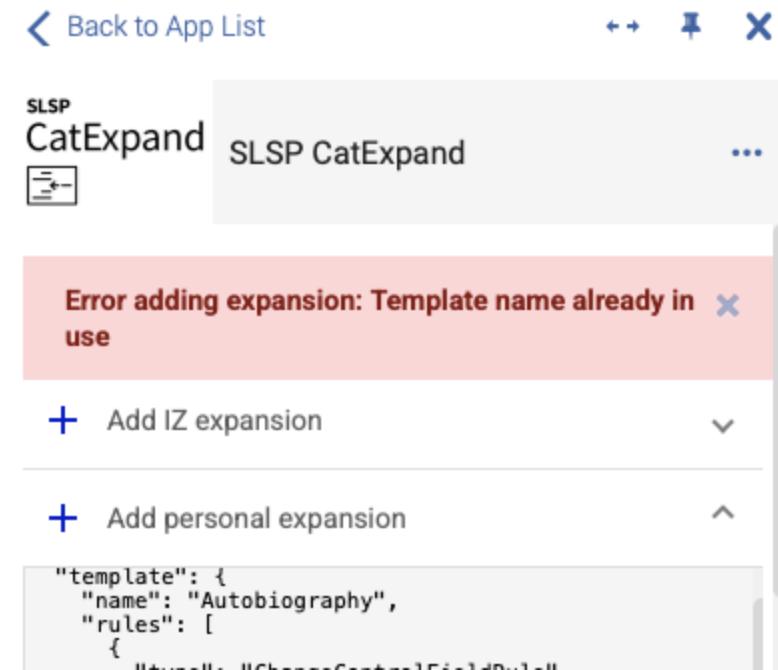
**Purpose:** Display messages to users

**Methods:**

- `success()`, `info()`, `warning()`, `error()` - Show alerts

**Example:**

```
alertService.success('Item updated!');  
alertService.error('Error adding expansion: ' + error.message,  
  { autoClose: false });
```



# Store Service

**Purpose:** Local browser storage for temporary data

**Key Features:**

- Store temporary data in browser
- Not persisted across sessions/devices
- Useful for caching, temporary state

**Remember:**

- For persistent user data → use Settings Service
- For persistent config → use Configuration Service

# REST Service

**Purpose:** Data retrieval and manipulation via Alma API

**Why it's the most important:**

- Core functionality for most Cloud Apps
- Direct access to Alma data (items, users, loans, etc.)
- Enables CRUD operations on Alma resources

**Key Features:**

- **Automatic authentication** - Uses logged-in user's credentials
- **Permission-based** - User needs appropriate Alma roles
- **IZ API access only** - Accesses Institution Zone data

 REST Service: [developers.exlibrisgroup.com/cloudapps/docs/api/rest-service](https://developers.exlibrisgroup.com/cloudapps/docs/api/rest-service)

Alma API: [developers.exlibrisgroup.com/alma](https://developers.exlibrisgroup.com/alma)

# Accessing Network Zone (NZ) API

**Problem:** REST Service only accesses Institution Zone (IZ) API

**SLSP Use Case:** Need access to Network Zone (NZ) data

- Examples: SLSP Card, SLSP CatExpand

**Solution:** Use Cloud App Proxy for NZ API access

**How it works:**

- Proxy acts as external API endpoint
- Configure in Cloud App manifest
- User roles are still checked (permission-based)
- Enables NZ data retrieval in multi-tenant environment

# Using External APIs

## Common Use Cases:

- External databases, web services, third-party integrations
- Data enrichment (covers, bibliographic data)
- **Custom backends** with database & scheduled jobs
  - Example: SLSP <> 7DM integration
  - Backend handles DB & batch operations, Cloud App provides UI

# External APIs: Technical Details

## Requirements:

- Configure **CSP** (Content Security Policy) in manifest.json
- Must comply with **CORS** restrictions
- May need backend proxy for CORS-restricted APIs

# Capabilities & Boundaries

What Cloud Apps CAN do:

-  Access and manipulate data via Alma REST API
-  React to the current context (e.g., active record or page)
-  Custom workflows and automations
-  Integration with external systems and APIs

## What Cloud Apps CANNOT do:

- ✗ Modify Alma's main UI (navigation, forms, MDE, etc.)
- ✗ Limited to data accessible via Alma REST API
- ✗ Perform batch operations (max 10 concurrent calls)
- ✗ Run background jobs or scheduled tasks

## 4. Angular Basics

# Angular Fundamentals

## Core Concepts:

- **Components** - UI building blocks
- **Templates** - HTML with Angular syntax
- **Services** - Business logic & data
- **Dependency Injection** - Service management

## You'll use:

- TypeScript (typed JavaScript)
- RxJS (reactive programming)
- Angular CLI (development tools)

# RxJS & Asynchronous Patterns

**RxJS = Reactive Extensions for JavaScript**

- Frontend is inherently asynchronous (API calls, user interactions)
- RxJS makes this manageable with consistent patterns

**Common Pattern:**

```
this.restService.call('/users')
  .pipe(
    map(users => users.filter(u => u.active)),
    catchError(error => of([]))
  )
  .subscribe(activeUsers => {
    this.users = activeUsers;
  });
}
```

**Key Concepts:**

## 6. Publishing & Lifecycle

# Cloud App Store & Publishing

## Process:

1. Build production version ( `eca build` ) and verify build is successful
2. Upload code to GitHub and create a release
3. Submit app to Ex Libris App Center (Developer Network)
4. Await review and approval
5. ... for updates, create new GitHub releases

# Beta Versions & Testing

## What are Beta versions?

- Pre-release versions for testing with real users
- Available alongside stable version
- Users can opt-in to beta testing

## Benefits:

- Test new features before full release
- Gather feedback from real usage
- Safe rollback to stable version if issues arise

---

More information

View the app on:  | Ex Libris App Center

Version: v1.0 | [\(Try v.1.1-beta\)](#) | Help

# IZ Restrictions

## What are IZ Restrictions?

- Control which institutions can install your app
- Set with `relevantForInst` field in `manifest.json`
- App won't appear in App Center for other institutions

## Use Cases:

- **SLSP-specific apps** - Restrict to SLSP institutions only
- **Custom institutional apps** - Single institution only
- **Pilot programs** - Limit to participating institutions

# Security Considerations

## Understanding the Security Model:

- Cloud Apps introduce third-party code into Alma environment
- Apps run in sandboxed iframe with security restrictions
- Public apps reviewed by Ex Libris before initial publication
- Update review process is unclear - updates are deployed quickly

## Transparency Requirements:

- Cloud Apps code must be open source (for public apps)
- External API connections defined in `manifest.json`
- Clear visibility into what resources apps access

# Security: Risks & Protection

What malicious apps could do:

- **Data exfiltration** - Steal patron data, circulation history, send externally
- **Data manipulation** - Alter records, loans, fines via API
- **Phishing** - Fake login forms inside Alma

How to Protect:

-  Only allow apps from trusted sources in your IZ
-  Review source code & manifest.json before installation and updates

## 7. Reference & Resources

# Resources

## Official Documentation:

- Cloud Apps Docs: [developers.exlibrisgroup.com/cloudapps](https://developers.exlibrisgroup.com/cloudapps)
- Alma API Docs: [developers.exlibrisgroup.com/alma](https://developers.exlibrisgroup.com/alma)
- SDK Getting Started: [Cloud Apps SDK](#)
- App Center Examples: [developers.exlibrisgroup.com/appcenter](https://developers.exlibrisgroup.com/appcenter)

## Our Workshop Repository:

- This presentation
- Development setup instructions
- Sample app code

# **Hands-on Time!**

Let's build something together!

**Remember:** This is collaborative - ask questions, share ideas!

# Prerequisites & Setup

## 1. IDE Setup:

- Use your preferred IDE or **recommended: VS Code**

## 2. Get the Workshop Repository:

- **Option A:** Git (recommended)

```
git clone https://github.com/Swiss-Library-Service-Platform/clouddemo
```

- **Option B:** Download ZIP

- Go to: <https://github.com/Swiss-Library-Service-Platform/clouddemo>
  - Click "Code" → "Download ZIP"