

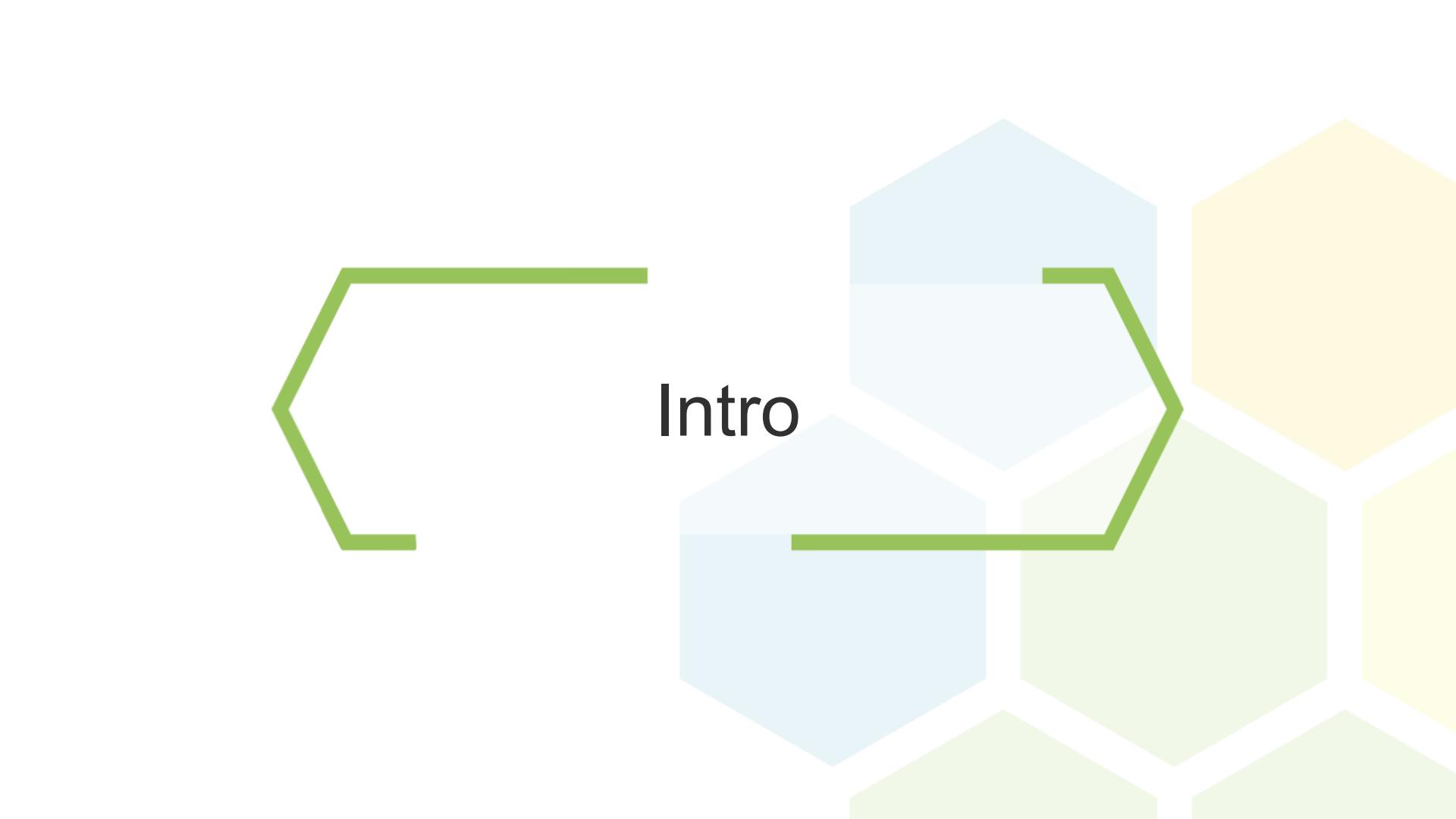
# Records Management: the API story

David Webster, Staff Software Engineer, Alfresco

# Next 30 mins:

- Intro (me, team, product)
- Records Management -> Governance Services
- API overview
- Demo





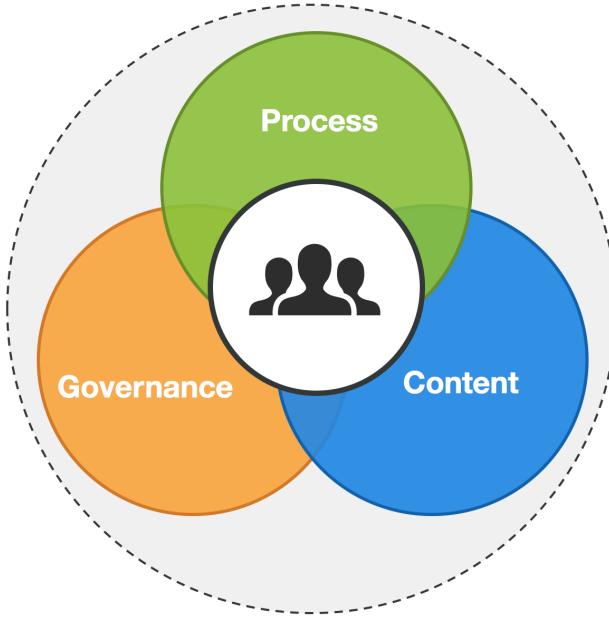
# Intro

# David Webster

- Alfresco Engineer since 2010
- A lead engineer in GS team
- Working on RM
- Previously on Share & other UI stuff
- Order of the Bee member
- Alfresco community fan
- Father/husband
- Search and Rescue volunteer



# Governance Services vs Records Management



# Why Governance Services?

- Governance includes more than just management of records
- Governance applies to all content, including that outside of RM site
- RM module is currently how we ship our governance capabilities, but that might change.

# 12 months in review:

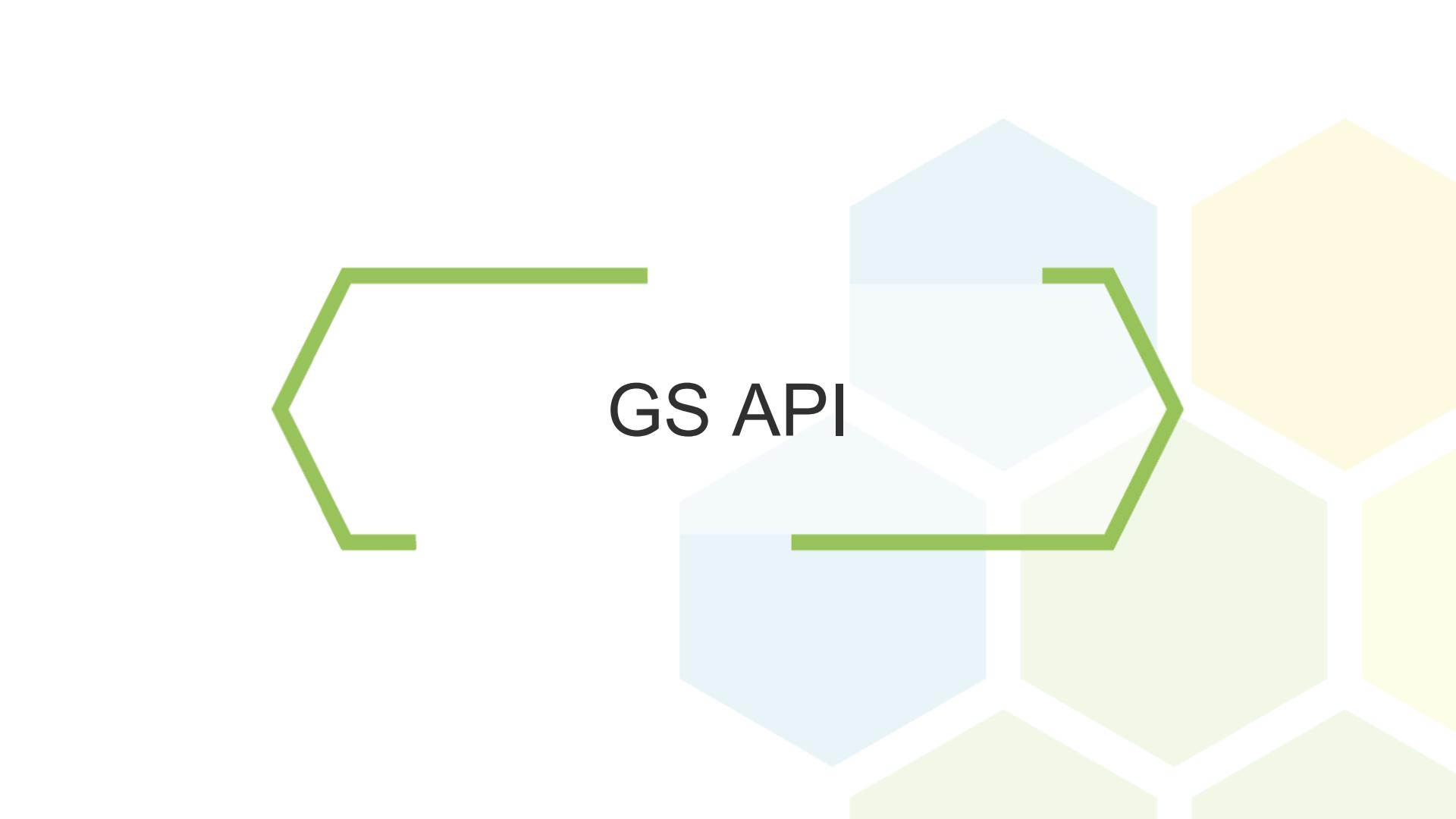
- 21 Releases (2 major Enterprise, 3 Community, 3 Service packs, 13 Hot fixes)
- Team grown to 18 (11 Maidenhead, 6 Romania, 1 Sydney)
- Over 2100 JIRAs worked on.
- Re-established GitHub mirror
- Travis-CI builds
- First community PR

When	What
April 2016	<b>V2.4.b, V2.4</b>
May 2016	V2.3.0.4, V2.3.0.5
June 2016	V2.5-EA1
August 2016	V2.3.0.6, V2.5-EA2, V2.4.0.1 <b>V2.5</b>
September 2016	V2.3.0.7, <b>V2.5.a</b> , V2.2.0.1
October 2016	V2.3.0.8, V2.2.0.2
November 2016	V2.5.0.1, <b>V2.3.1</b>
January 2017	V2.3.0.9
February 2017	<b>V2.5.1</b> , V2.4.0.2
March 2017	V2.2.0.3
April 2017	V2.3.1.1, <b>V2.4.1</b> , <b>V2.5.b</b>

Especially for BeeCon:

2.5.b released  
yesterday afternoon

(5.2 compatibility)



**GS API**

# GS v1 REST APIs

- APIs as a feature
- Debuts in RM 2.6
- Existing extension points and old REST (v0) APIs and Java APIs are still there.
- You should use v1 REST APIs in preference.
- API first development
- Match v1 APIs from elsewhere in Digital Business Platform.

# Scope of the APIs

- Complete File Plan CRUD
- Filing records
- Declaring content as records
- Update record metadata



# Explore APIs

- API Explorer: <http://localhost:8080/gs-api-explorer/>
- Postman: <https://github.com/Alfresco/gs-api-postman-collection>

# API Explorer

The screenshot shows a web browser window displaying the Alfresco Governance Services REST API Explorer at `localhost:8080/gs-api-explorer/`. The interface includes a header with the Alfresco logo, user authentication fields for 'admin' and a password placeholder, and a navigation bar with links for 'GS Core API', 'List Operations', and 'Expand Operations'. The main content area lists various API endpoints categorized by their purpose:

Category	Action
file-plans : Retrieve and manage file plans	Show/Hide   List Operations   Expand Operations
files : Perform operations on non-record files	Show/Hide   List Operations   Expand Operations
gs-sites : Retrieve and manage the RM site	Show/Hide   List Operations   Expand Operations
record-categories : Retrieve and manage record categories	Show/Hide   List Operations   Expand Operations
record-folders : Retrieve and manage record folders	Show/Hide   List Operations   Expand Operations
records : Perform record specific operations	Show/Hide   List Operations   Expand Operations
transfer-containers : Retrieve and manage transfer containers	Show/Hide   List Operations   Expand Operations
transfers : Retrieve and manage transfers	Show/Hide   List Operations   Expand Operations
unfiled-containers : Retrieve and manage unfiled records containers	Show/Hide   List Operations   Expand Operations
unfiled-record-folders : Retrieve and manage unfiled record folders	Show/Hide   List Operations   Expand Operations

At the bottom of the page, a footer note indicates the base URL: [ BASE URL: /alfresco/api/-default-/public/gs/versions/1 , API VERSION: 1 ]

# Postman

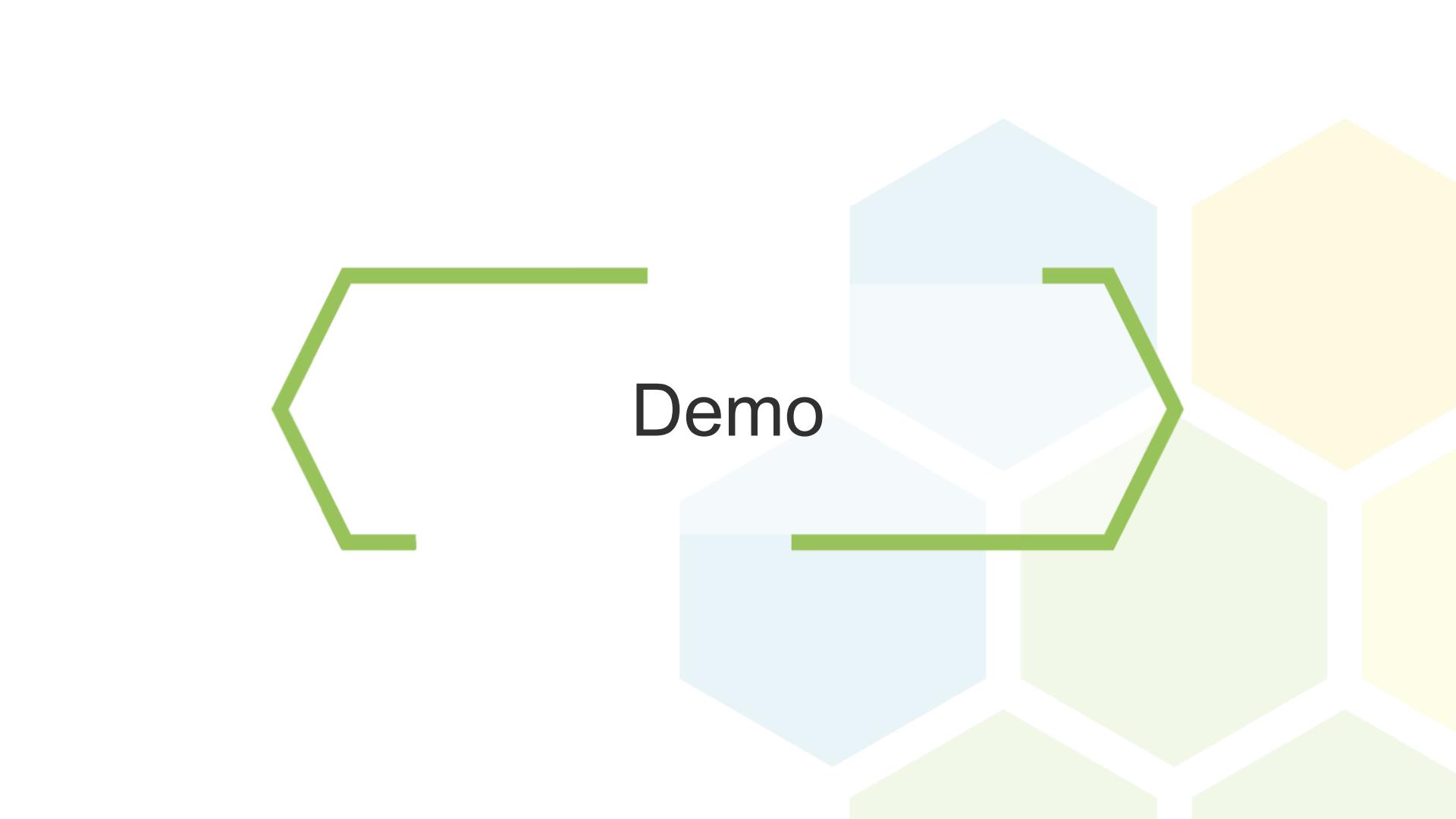
The screenshot shows the Postman application interface. The left sidebar displays a collection structure under 'GS API Demo' with various requests (POST, GET, PUT, DELETE) across different categories like '01 RM Site', '02 File Plans', '03 Record Categories', etc. The main workspace shows a POST request for '05-Create Root Category'. The 'Tests' tab contains a snippet of JavaScript code:

```
1 tests["Status code is 201"] = responseCode.code === 201;
2
3 var jsonData = JSON.parse(responseBody);
4 postman.setGlobalVariable("ig.root.category.id", jsonData.entry.id);
```

The 'Body' tab shows the response body in JSON format:

```
1 {
2   "entry": {
3     "aspectNames": [
4       "mm:initialRecordDefinition",
5       "mm:recordComponentIdentifier",
6       "mm:recordCategoryCustomProperties",
7       "cm:title",
8       "mm:filePlanComponent",
9       "cm:auditable"
10    ],
11    "createDate": "2017-04-23T21:29:55.577+0000",
12    "createdByUser": {
13      "id": "admin",
14      "displayName": "Administrator"
15    },
16    "modifiedDate": "2017-04-23T21:29:55.577+0000",
17    "modifiedByUser": {
18      "id": "admin",
19      "displayName": "Administrator"
```

The top right corner shows the status: Status: 201 Created Time: 293 ms Size: 957 B.



Demo









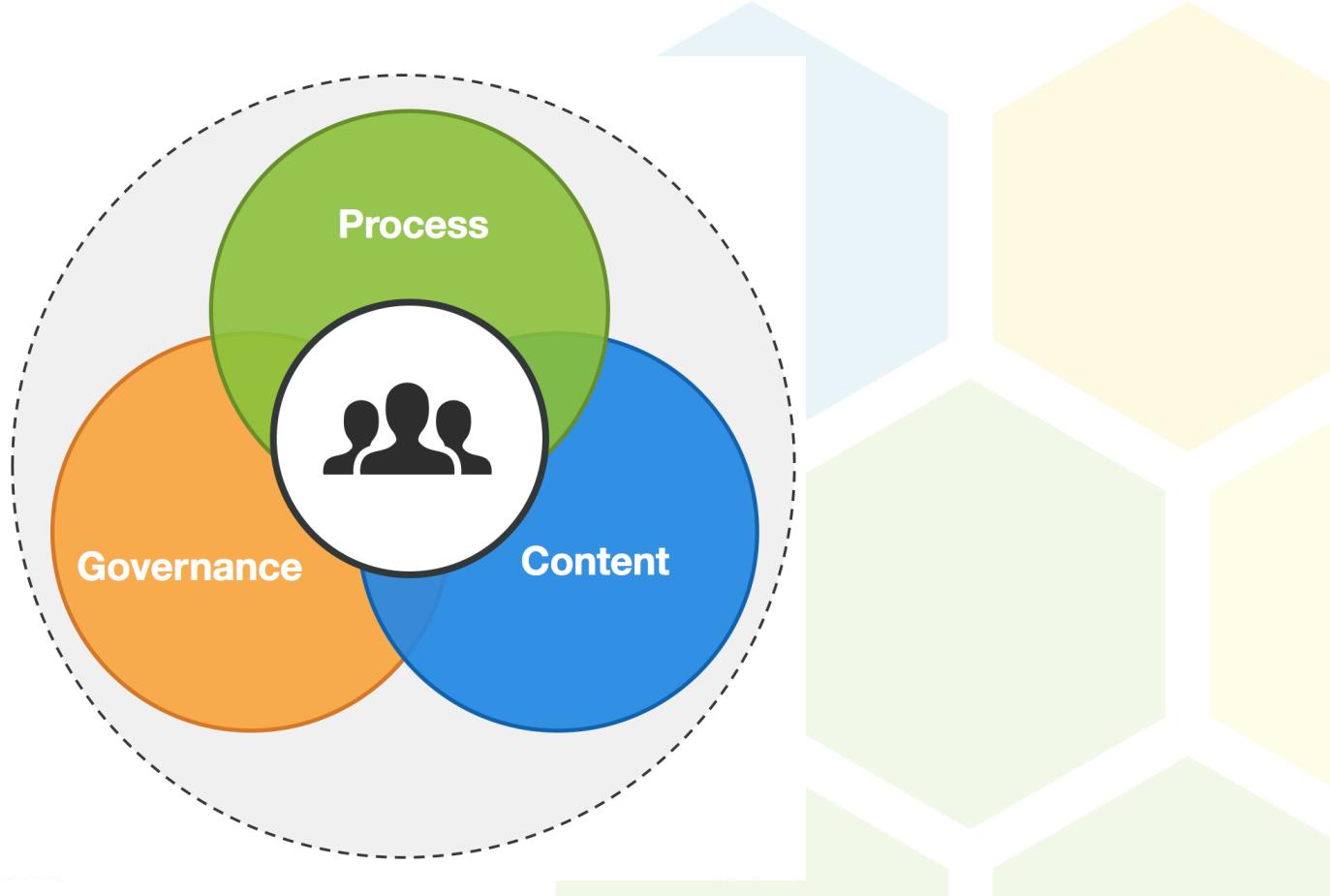




The API story

Your API story  
continues...

# Process?



# Recap

- Governance key part of Digital Business Platform
- Governance Services APIs in RM 2.6
- Match core v1 APIs
- Cover file plan set up & record ingestion
- Docs: <http://localhost:8080/gs-api-explorer/>
- Postman collection: <https://github.com/Alfresco/gs-api-postman-collection>





# Thanks

David.Webster@alfresco.com  
@davidcognite