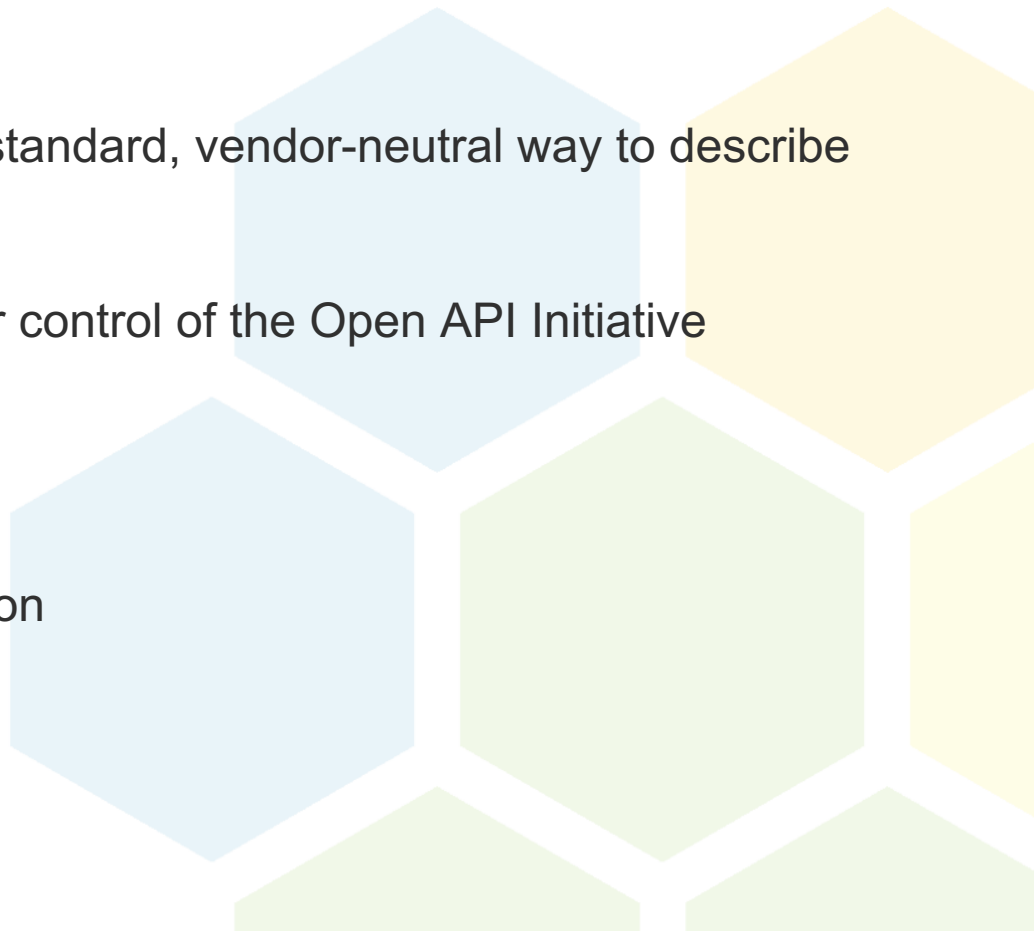


Automatic APIs with Swagger & SpringFox

Will Abson, Alfresco

Swagger Basics

- Swagger Specification provides a standard, vendor-neutral way to describe your REST API
- Developed by Swagger, now under control of the Open API Initiative (<https://www.openapis.org/>)
- Current version is 2.0
- Either JSON or YAML representation



JSON Example

From Activiti Process Services API

```
▼ {swagger: "2.0", info: {,...}, host: "act-test-env.alfresco.me", basePath: "/activiti-app/api",...}
  basePath: "/activiti-app/api"
  ► definitions: {RelatedContentRepresentation: {type: "object",...}, AppDefinitionSaveRepresentation: {type: "object",...},...}
  host: "act-test-env.alfresco.me"
  ► info: {,...}
  ▼ paths: {,...}
    ► /enterprise/account/integration: {get: {tags: ["account-integration"], summary: "Retrieve external account information",...}}
    ► /enterprise/admin/basic-auths: {get: {tags: ["admin-endpoints"], summary: "List endpoint authorizations",...},...}
    ► /enterprise/admin/basic-auths/{basicAuthId}: {get: {tags: ["admin-endpoints"], summary: "Get an endpoint authorization",...},...}
    ► /enterprise/admin/endpoints: {get: {tags: ["admin-endpoints"], summary: "List endpoints",...},...}
    ► /enterprise/admin/endpoints/{endpointConfigurationId}: {get: {tags: ["admin-endpoints"], summary: "Get an endpoint",...},...}
    ► /enterprise/admin/groups: {get: {tags: ["admin-groups"], summary: "Query groups", operationId: "getGroupsUsingGET_1",...},...}
    ► /enterprise/admin/groups/{groupId}: {get: {tags: ["admin-groups"], summary: "Get a group", operationId: "getGroupUsingGET",...},...}
    ► /enterprise/admin/groups/{groupId}/action/activate: {post: {tags: ["admin-groups"], summary: "Activate a group", operationId: "activateUsingPOST",...}}
    ► /enterprise/admin/groups/{groupId}/add-all-users: {,...}
    ► /enterprise/admin/groups/{groupId}/capabilities: {post: {tags: ["admin-groups"], summary: "Add capabilities to a group",...}}
    ► /enterprise/admin/groups/{groupId}/capabilities/{groupCapabilityId}: {delete: {tags: ["admin-groups"], summary: "Remove a capability from a group",...}}
    ► /enterprise/admin/groups/{groupId}/members/{userId}: {,...}
    ► /enterprise/admin/groups/{groupId}/potential-capabilities: {,...}
```

YAML Example

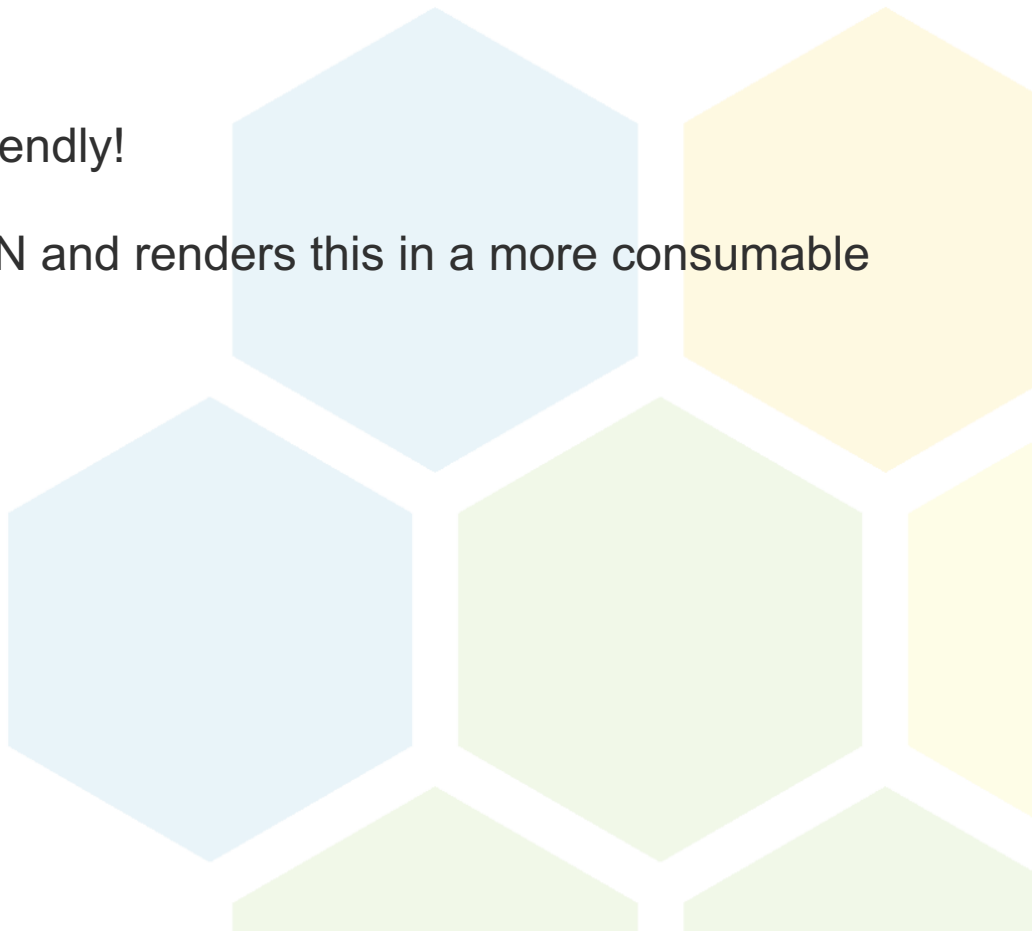
From content services specification at
<https://github.com/Alfresco/rest-api-explorer/blob/master/src/main/webapp/definitions/alfresco-core.yaml>

```
swagger: '2.0'
info:
  description: |
    **Core API**

    Provides access to the core features of Alfresco Content Services.
  version: '1'
  title: Alfresco Content Services REST API
  basePath: /alfresco/api/-default-/public/alfresco/versions/1
securityDefinitions:
  basicAuth:
    type: basic
    description: HTTP Basic Authentication
security:
  - basicAuth: []
consumes:
  - application/json
produces:
  - application/json
tags:
  - name: activities
    description: Retrieve and manage activities
```

Swagger UI

- Raw API spec is not very reader-friendly!
- Swagger UI takes your YAML/JSON and renders this in a more consumable format
- Also allows you to try out the API



Swagger Petstore 1.0.0

[Base url: petstore.swagger.io/v2]
<http://petstore.swagger.io/v2/swagger.json>

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [#irc.freenode.net, #swagger](http://irc.freenode.net). For this sample, you can use the api key `special-key` to test the authorization filters.

[Terms of service](#)

[Contact the developer](#)

[Apache 2.0](#)

[Find out more about Swagger](#)

Schemes

HTTP



Authorize



pet Everything about your Pets



POST

/pet Add a new pet to the store



PUT

/pet Update an existing pet



POST

/pet Add a new pet to the store



Parameters

Try it out

Name

Description

body * required

Pet object that needs to be added to the store

(body)

Example Value | Model

```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "doggie",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Parameter content type

application/json



Testing using Swagger-UI

- Modify parameters as needed and click 'Try it out!'
- Full headers and response body displayed
- Curl command given

The screenshot displays the Swagger-UI interface for a REST API. At the top, the 'Response Content Type' is set to 'application/json'. Below this, the 'Curl' section shows the command: `curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic YWRtaW5AYXBwLmFjdG12aXRpLnVbTprMw5nazBu'`. The 'Request URL' is `http://localhost:9999/activiti-app/api/enterprise/admin/tenants`. The 'Request Headers' section shows `{ "Accept": "application/json" }`. The 'Response Body' shows a JSON array: `[{ "id": 1, "name": "test" }]`. The 'Response Code' is `200`. The 'Response Headers' section shows a detailed set of headers including 'pragma', 'date', 'x-content-type-options', 'server', 'x-frame-options', 'transfer-encoding', 'content-type', 'cache-control', 'x-xss-protection', and 'expires'.

Response Content Type: application/json

[Try it out!](#) [Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic YWRtaW5AYXBwLmFjdG12aXRpLnVbTprMw5nazBu'
```

Request URL

```
http://localhost:9999/activiti-app/api/enterprise/admin/tenants
```

Request Headers

```
{
  "Accept": "application/json"
}
```

Response Body

```
[
  {
    "id": 1,
    "name": "test"
  }
]
```


Response Code

```
200
```

Response Headers

```
{
  "pragma": "no-cache",
  "date": "Wed, 15 Mar 2017 10:49:59 GMT",
  "x-content-type-options": "nosniff",
  "server": "Apache-Coyote/1.1",
  "x-frame-options": "DENY",
  "transfer-encoding": "chunked",
  "content-type": "application/json",
  "cache-control": "no-cache, no-store, max-age=0, must-revalidate",
  "x-xss-protection": "1; mode=block",
  "expires": "0"
}
```


Testing using Swagger-UI

Response Content Type application/json 
[Try it out!](#) [Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic YWRtaW5AYXBwLmFjdG12aXRpLmNvbTprMW5nazBu
```

Request URL

```
http://localhost:9999/activiti-app/api/enterprise/admin/tenants
```

Request Headers

```
{  
  "Accept": "application/json"  
}
```

Response Body

```
[  
  {  
    "id": 1,  
    "name": "test"  
  }  
]
```

Response Code

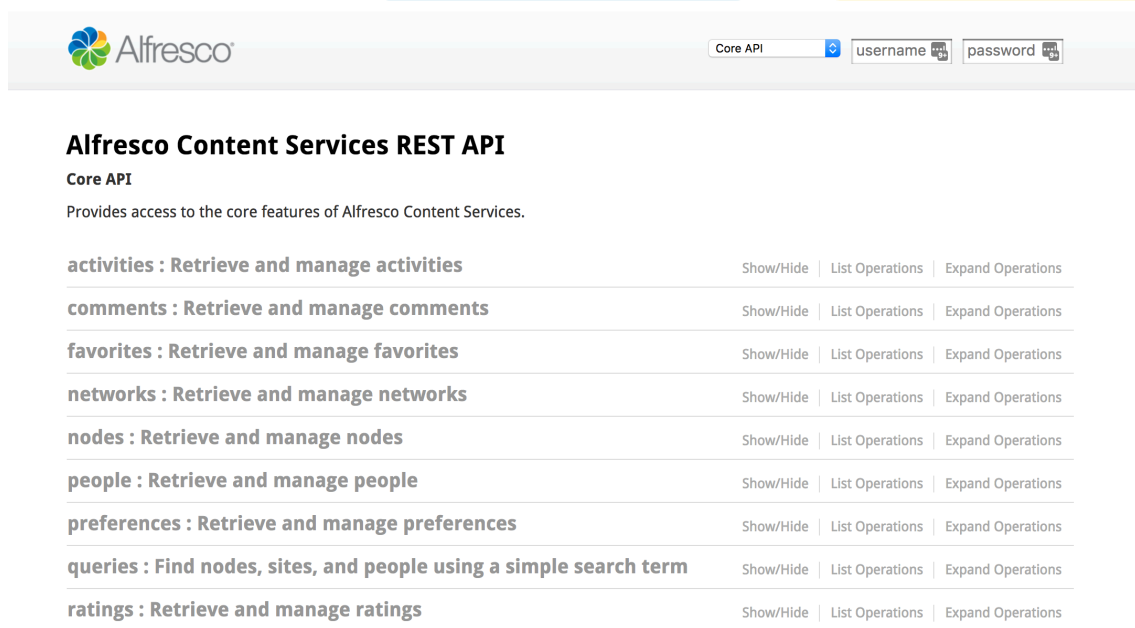
```
200
```

Response Headers

```
{  
  "pragma": "no-cache",  
  "date": "Wed, 15 Mar 2017 10:49:59 GMT",  
  "x-content-type-options": "nosniff",  
  "server": "Apache-Coyote/1.1",  
  "x-frame-options": "DENY",  
  "transfer-encoding": "chunked",  
  "content-type": "application/json",  
  "cache-control": "no-cache, no-store, max-age=0, must-revalidate",  
  "x-xss-protection": "1; mode=block",  
  "expires": "0"  
}
```

Modifying Swagger UI

- Single HTML page with JQuery
- Supports token-based auth, easily modifiable to support other schemes (e.g. Basic auth)



Alfresco Content Services REST API

Core API
Provides access to the core features of Alfresco Content Services.

activities : Retrieve and manage activities	Show/Hide	List Operations	Expand Operations
comments : Retrieve and manage comments	Show/Hide	List Operations	Expand Operations
favorites : Retrieve and manage favorites	Show/Hide	List Operations	Expand Operations
networks : Retrieve and manage networks	Show/Hide	List Operations	Expand Operations
nodes : Retrieve and manage nodes	Show/Hide	List Operations	Expand Operations
people : Retrieve and manage people	Show/Hide	List Operations	Expand Operations
preferences : Retrieve and manage preferences	Show/Hide	List Operations	Expand Operations
queries : Find nodes, sites, and people using a simple search term	Show/Hide	List Operations	Expand Operations
ratings : Retrieve and manage ratings	Show/Hide	List Operations	Expand Operations

Use of Swagger within Alfresco

- Content Services team introduce use of Swagger's OpenAPI spec to implement contractual definitions of all API
- Process Services / ADF team utilise Swagger-Codegen to generate Alfresco JS-API
- SpringFox introduced into Process Services app by Jean-Marie Pascale

Design first or code first?

Design First Approach

When developer experience matters

When ensuring good communication

Code First Approach

When Speedy Delivery Matters

When Developing Internal APIs

<https://swaggerhub.com/blog/api-design/design-first-or-code-first-api-development/>

Introducing SpringFox

Automated JSON API documentation for API's built with Spring, so great for existing APIs

1. Add annotations to your code
2. SpringFox generates JSON API description
3. SpringFox bundles a custom Swagger-UI and provides Spring MVC sample code to expose this in your app

SpringFox Operation

- Scans packages at startup for **@RestController** annotations to find all APIs with **@RequestMapping**
- Uses Spring MVC, Swagger and custom SpringFox annotations to determine API information

Springfox Annotations

Swagger `@Api` annotations used to provide basic metadata (tags, description) on the group of API operations

```
@RestController
@RequestMapping(
    .....value = "/enterprise/admin/users",
    .....produces = MediaType.APPLICATION_JSON_VALUE
)
@Api(
    .....tags = "admin-users",
    .....description = "Manage users"
)
```

Springfox Annotations

Swagger `@ApiOperation` annotation used to provide information on the API itself

```
@RequestMapping(value = "/{userId}", method = RequestMethod.GET)
@ApiOperation(
    ..... value = "Get a user"
)
public UserRepresentation getUser(@PathVariable Long userId, @RequestParam(required=false) boolean summary) {
    ... return userService.getUser(userId, summary);
}
```


Springfox Advanced Annotations

Parameters picked up from Spring `@RequestParam` annotation, Swagger annotation `@ApiParam` can be used to provide more information, e.g. description, default values

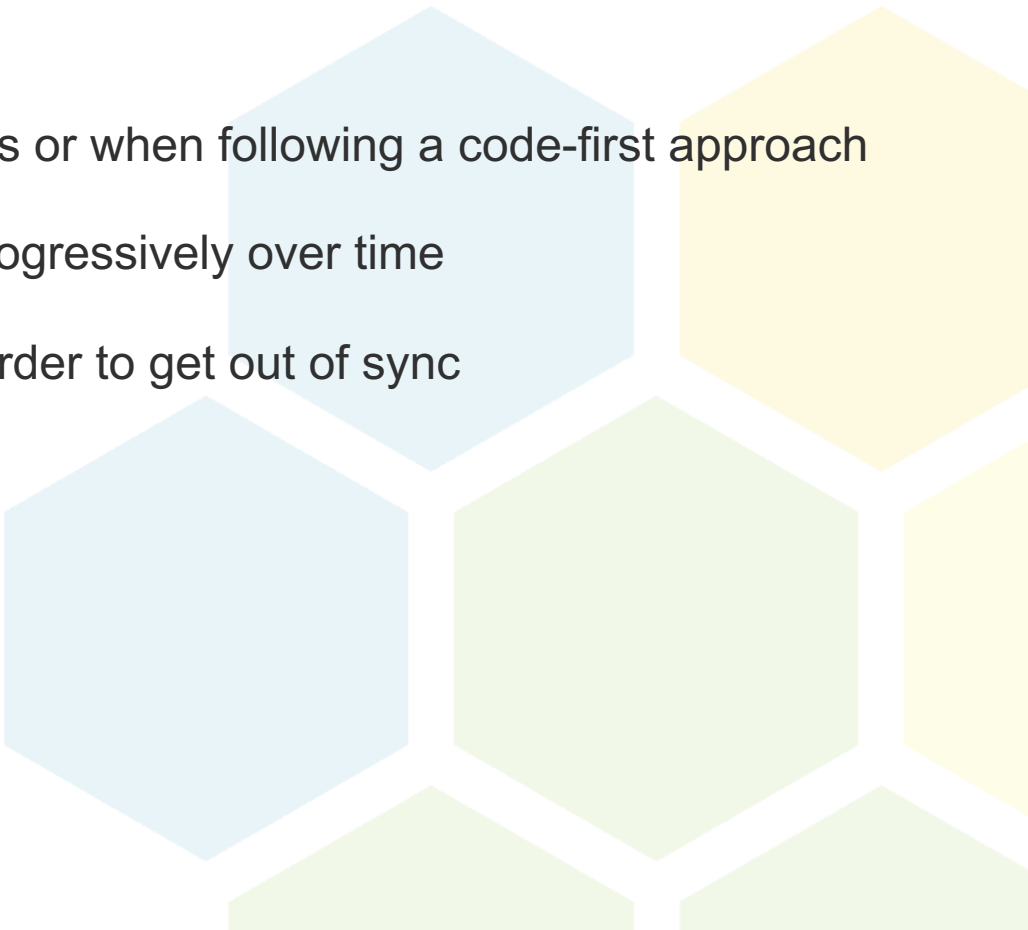
```
public ResultListDataRepresentation<AbstractUserRepresentation> getUsers(  
    @RequestParam(required=false) @ApiParam(value = "Search filter, matches any text within the user's first name, last name, full name or email") String filter,  
    @RequestParam(required=false) @ApiParam(value = "User status, case-insensitive", allowableValues = "PENDING,INACTIVE,ACTIVE,DELETED") String status,  
    @RequestParam(required=false) @ApiParam(value = "Account type, case-insensitive", allowableValues = "TRIAL,ENTERPRISE") String accountType,  
    @RequestParam(required=false) @ApiParam(value = "Sort order, case-sensitive", allowableValues = "createdAsc,createdDesc,emailAsc,emailDesc,nameAsc,nameDesc", defaultVal  
    @RequestParam(required=false) @ApiParam(value = "Company name") String company,  
    @RequestParam(required=false) @ApiParam(value = "Alternative to `page`/`size`-based paging provided for historic reasons only. Will be ignored if `page` is specified.")  
    @RequestParam(required=false) @ApiParam(value = "Zero-based page number, optionally specify `size` to change the page size", defaultValue = "0") Integer page,  
    @RequestParam(required=false) @ApiParam(value = "Maximum number of users to return", defaultValue = "100") Integer size,  
    @RequestParam(required=false) @ApiParam(value = "Group ID that users should be members of") Long groupId,  
    @RequestParam(required=false) @ApiParam(value = "Tenant ID that users must be within, number-only without any `tenant_` prefix") Long tenantId,  
    @RequestParam(required=false) @ApiParam(value = "Whether to return results in short summary format. By default full results are returned.", defaultValue = "false") bool
```

Springfox Advanced Annotations

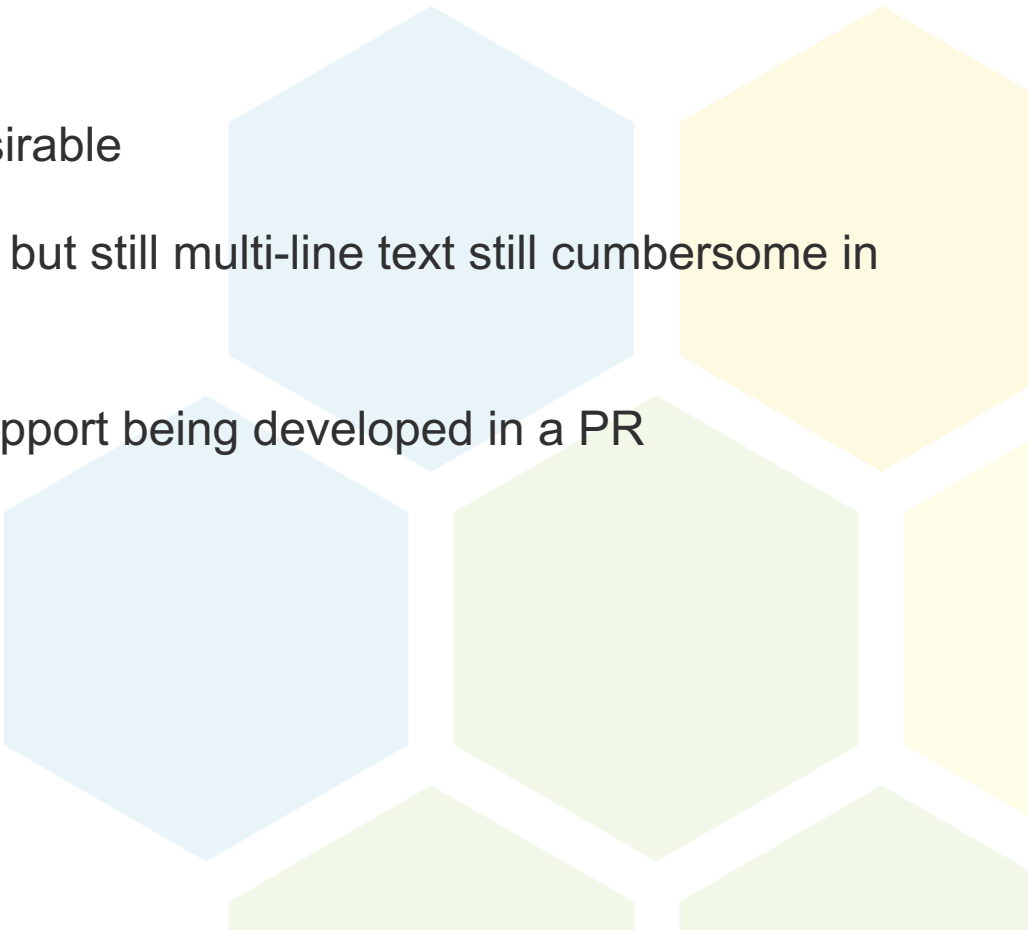
- Extended info on response body (if not implicit in return type) and response status via `@ApiResponse` and `@ApiResponses` (multiple)

```
@ApiResponses({  
    ..... @ApiResponse(code = 400, message = "If the specified `tenantId` value does not exist"),  
    ..... @ApiResponse(code = 403, message = "If the current user does not have the tenant manager or tenant admin capabilities."  
})
```

Benefits

- ✓ Great for documenting existing APIs or when following a code-first approach
 - ✓ Extra descriptions can be added progressively over time
 - ✓ API docs is inline in the code so harder to get out of sync
- 
- A decorative graphic in the bottom right corner consisting of a cluster of hexagons in light blue, light green, and light yellow colors, arranged in a honeycomb-like pattern.

Cons

- Code-first approach not always desirable
 - Longer descriptions use Markdown but still multi-line text still cumbersome in Java strings (e.g. Code examples)
 - JSON-only at present but YAML support being developed in a PR
- 
- A decorative background on the right side of the slide consisting of several overlapping hexagons in light blue, light green, and light yellow colors.

More Information

- <http://swagger.io/swagger-ui/>
- <http://springfox.github.io/springfox/>
- Find me later for code examples!





Speaker contacts

Email: will.abson@alfresco.com

Twitter: [@wabson](https://twitter.com/wabson)