Developing Applications with Alfresco’s Unified REST APIs

Will Abson, Alfresco

Alfresco JavaScript API

- A set of JavaScript libraries to allow easy access to Alfresco’s REST APIs
- Part of the Application Development Framework (ADF)
- Used internally by all ADF (Angular2) Components and demo apps
- Supports Node.js scripts as well as client-side JS

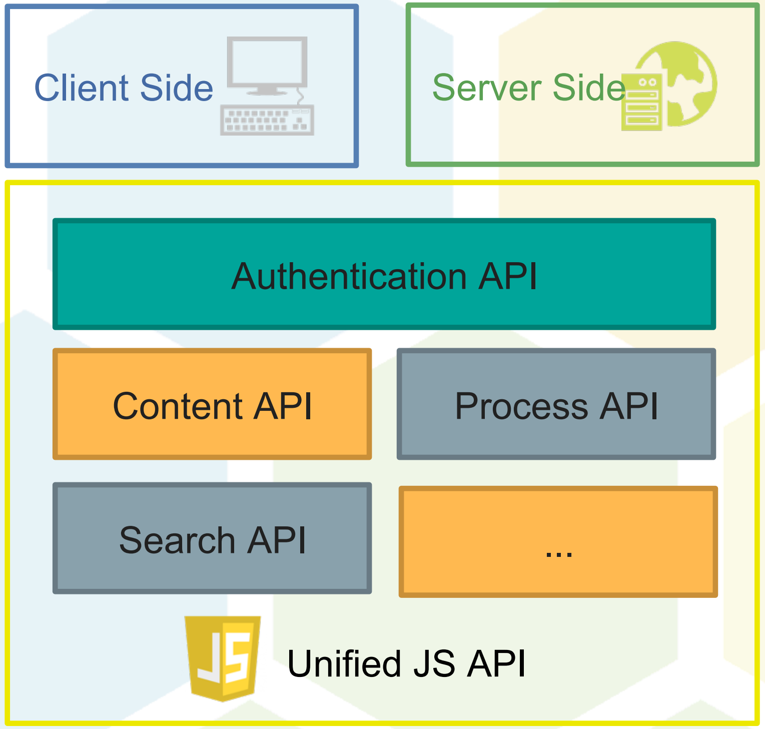
Alfresco JavaScript API

How to import the library in a server side Node.js project:

```
var AlfrescoApi = require('alfresco-js-
api');
```

How to import the library in a client side JavaScript project:

```
<script src="node_modules/alfresco-js-api/
alfresco-js-api.js"></script>
```



Using the API

Node Example

```
var AlfrescoApi = require('alfresco-js-api');

var alfrescoJsApi = new AlfrescoApi({ provider:'ECM' });
```

Node Example

```
var AlfrescoApi = require('alfresco-js-api');

var alfrescoJsApi = new AlfrescoApi({ provider:'ECM' });

alfrescoJsApi.login('admin', 'admin').then(function (ticket)
{
    console.log('Login called successfully', ticket);
    // logic goes here
}, function (error) {
    console.error('Login error', error);
});
```

Authentication Notes

- Usual mechanism returns a promise, though events also supported
- Provider must be one of "**`ECM`**", "**`BPM`**" or "**`ALL`**"
- If you already have a ticket use **`loginTicket()`** instead of **`login()`** method (ticket will be validated) or **`ticketEcm`** or **`ticketBpm`** in constructor (no validation)
- Specify base URL(s) in the constructor using **`hostBpm`** and **`hostEcm`** properties, e.g. "**`http://localhost:8080`**", or context values using **`contextBpm`** and **`contextEcm`**

Browser Security – CORS

- Both content and process servers allow the use of CORS to enable the browser to make **cross-origin** requests to the backend APIs
- The HTTP Origin is the combination of protocol + hostname + port, e.g. http://server.myco.com:7777
- For Content Services use "enablecors" JAR by Gethin James (v5.1+) or uncomment sections in **`web.xml`**
- For Process Services set property **`cors.enabled=true`** in **`activiti-app.properties`**
- Alternatively you can use a proxy to front your ADF app and content and process backends under the same origin

Browser Security – CSRF

- Process Services applies a stateless CSRF protection to all API endpoints including the public REST APIs
- This can be turned off in **`activiti-app.properties`** but the JS-API will automatically send appropriate cookie + header values to work around the protection (unless **`disableCsrf`** property is set to **`true`**)

Making API Requests

- Top-level API class allows accessing all APIs via properties
- Process Services API group: **`alfrescoJsApi.activiti`**
- Content Services Core API group: **`alfrescoJsApi.core`**
- Content Services Search API group: **`alfrescoJsApi.search`**
- Content Services web scripts: **`alfrescoJsApi.webScript`**
- Below each primary group the API is broken down further, corresponding to the logical groupings in the API Explorer

Content Services Core APIs

APIs broken down further into sub-groups

- Activities: **`alfrescoJsApi.core.activitiesApi`**
- Nodes: **`alfrescoJsApi.core.nodesApi`** or alias **`alfrescoJsApi.nodes`**
- Sites: **`alfrescoJsApi.core.sitesApi`**
- People: **`alfrescoJsApi.core.peopleApi`**

Inside each group are the individual endpoints, e.g.
`alfrescoJsApi.core.nodesApi.createNode()`

Reference: https://github.com/Alfresco/alfresco-js-api/tree/master/src/alfresco-core-rest-api

Node Example

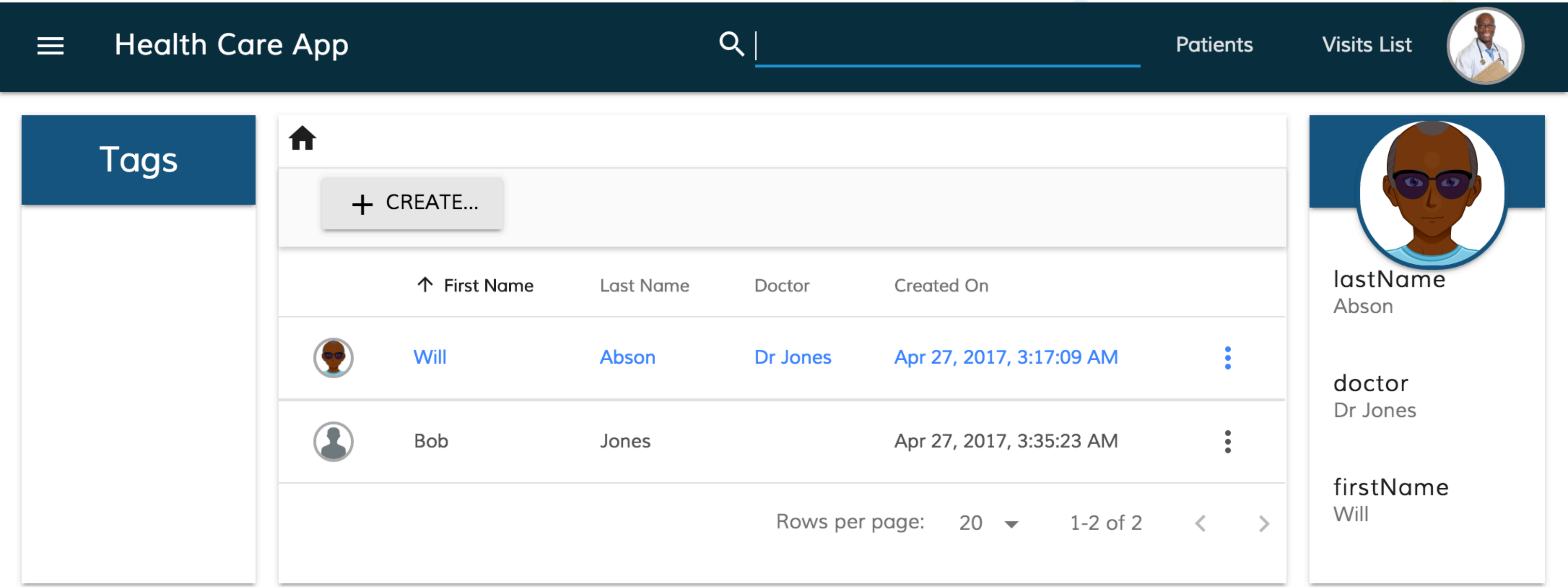
```
var AlfrescoApi = require('alfresco-js-api');
var alfrescoJsApi = new AlfrescoApi({ provider:'ECM' });

alfrescoJsApi.login('admin', 'admin').then(function (ticket) {
    console.log('Login called successfully', ticket);
    alfrescoJsApi.core.nodesApi.createNode('-root-', {
        name: 'test',
        nodeType: 'cm:folder'
    }).then(function(data) {
        console.log('Created folder', data);
    });
}, function (error) {
    console.error('Login error', error);
});
```

Node Demo

An Example App

Alfresco Healthcare App

- Implemented during an early week-long internal hackathon within the ADF team
- Aims to show customising an application to suit audiences in a particular vertical sector
- Uses ADF Components and JS-API
- Client-side components and Node.js scripts


Health Care App

Patients Visits List

Tags

CREATE...

	↑ First Name	Last Name	Doctor	Created On
	Will	Abson	Dr Jones	Apr 27, 2017, 3:17:09 AM
	Bob	Jones		Apr 27, 2017, 3:35:23 AM

Rows per page: 20 1-2 of 2

lastName
Abson

doctor
Dr Jones

firstName
Will

Soft-wiring Apps

- Alfresco allows us to compose core business logic and user interactions
- We call this **soft-wiring**
- Alfresco content services store all content and makes this available complemented by process interactions
- We can rapidly build new custom apps that respond to changes in the underlying platforms, using the Alfresco Angular2 components
- Option to prototype early using native UIs

Health Care App Demo

Loading Data

- Initially sample data was loaded manually, this was very painful!
- Users
- Folder structure
- Custom Content Models
- Custom Processes, Forms & App
- Initial attempt to script this using cURL
- Now using JS-API!

Health Care Data Demo

JS-API Summary

- ✓ Easy to include in your App
- ✓ Get started quickly
- ✓ Browser and Node.js support

More Information

- https://github.com/Alfresco/alfresco-js-api
- https://github.com/Alfresco/health-care-app
- https://github.com/covolution/enablecors

Speaker contacts

Email: will.abson@alfresco.com
Twitter: @wabson