

Starting Development with the Alfresco

Richard Esplin
Product Manager
richard.esplin@alfresco.com
June 2016

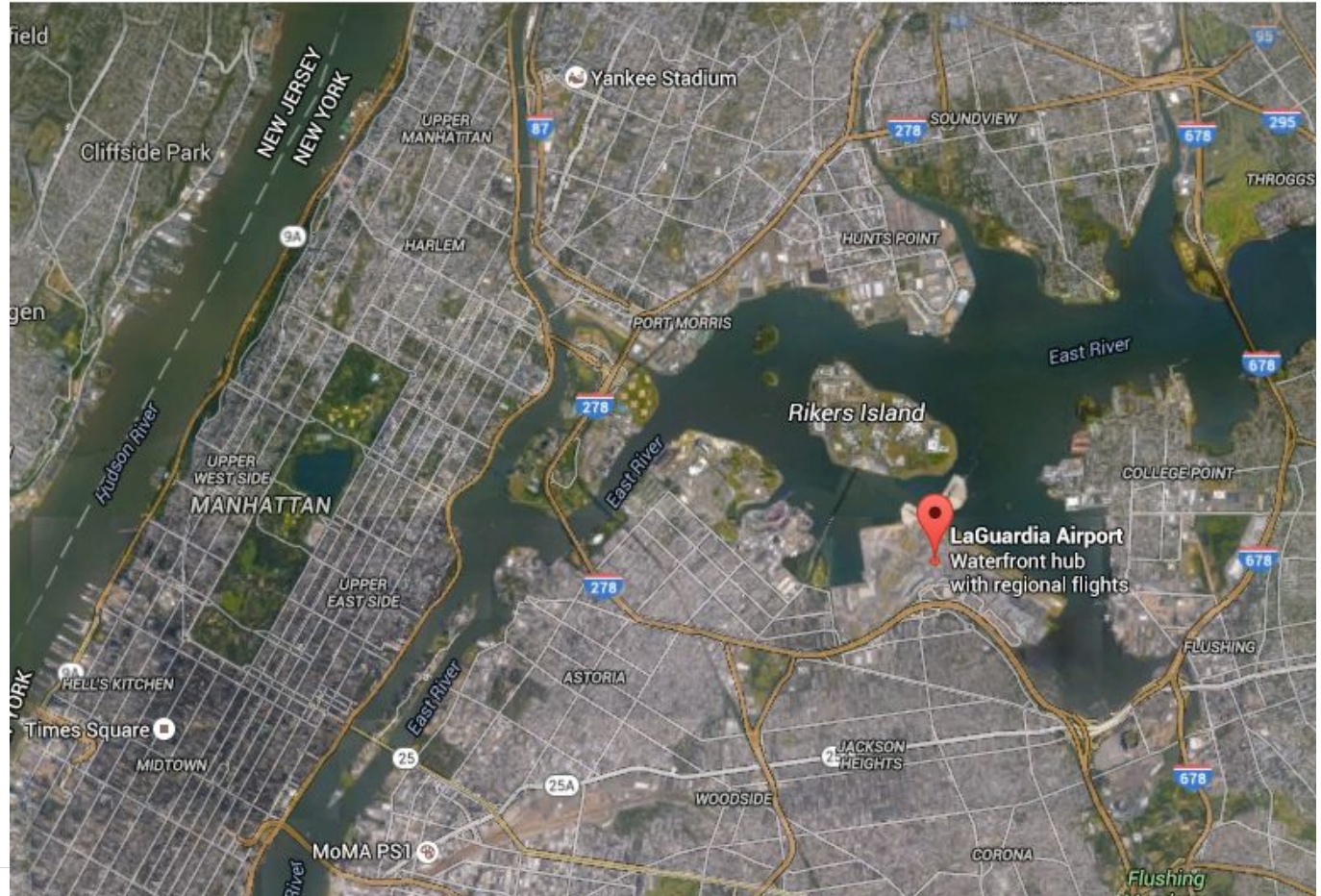




Our Goal for Today



Our Goal for Today



docs.alfresco.com

Alfresco Documentation

Get started with Alfresco

[What's new in Alfresco One?](#)

[Watch and learn about Alfresco One](#)

Alfresco One

[Alfresco One 5.0.2](#)

[Alfresco in the Cloud](#)

[Alfresco Analytics 1.1.0](#)

[Alfresco Media Management 1.0.1](#)

[Alfresco Outlook Integration 2.0.1](#)

[Alfresco Records Management 2.3](#)

[Alfresco SDK 2.1.1](#)

Alfresco Activiti

[Alfresco Activiti](#)

Alfresco Community

For older versions of Community Edition, see the Enterprise documentation that applies to your version.

[Alfresco Community Edition 5.1](#)

[Alfresco Community Edition 5.0](#)

[Alfresco Records Management Community 2.3.c](#)

Alfresco Mobile

[Alfresco Mobile for Android](#)

Get the full
documentation
library in PDF

The Alfresco Platform

Alfresco delivers **enterprise content management** and **business process management** as a platform upon which you can build applications to solve business problems.

Content Apps



Web Applications



External Systems

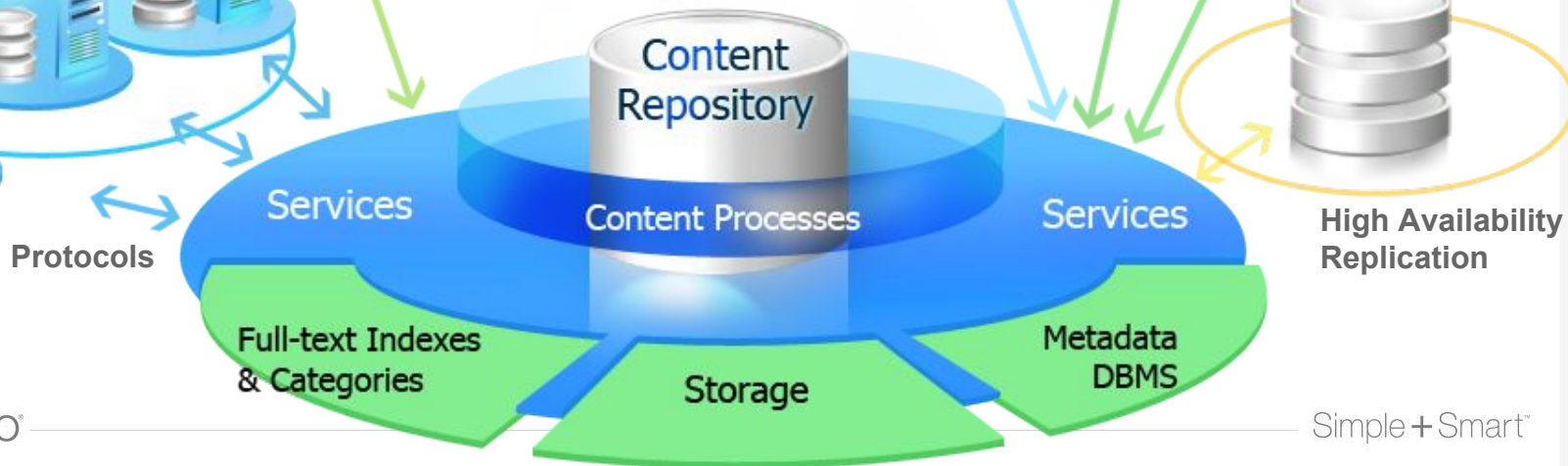


Development

Virtual File System



Public REST / CMIS APIs



docs.alfresco.com → Community / Alfresco One → Developer guide

What do you want to do?	Documentation
You would like to get an overview of the architecture of Alfresco from the developer's perspective	Architectural overview
You want to write client applications for Alfresco using a REST API	Alfresco REST API
You are going to build a Platform Extension.	Developing extensions
You are going to build a Platform Integration.	Developing integrations
You want to write small extensions to Alfresco in JavaScript and FreeMarker	Web Scripts
You want to create a custom REST API for Alfresco	Web scripts
You want develop applications for Alfresco using RAD	Alfresco SDK
You want to get developing for Alfresco using Maven	Alfresco SDK
You would like to know what APIs are available for Alfresco, and when you should use them	Overview of Alfresco APIs
You would like to know how to package your extensions for distribution	Extension packaging
You want to write new Alfresco services in Java, and need to check what APIs are supported	Alfresco Public Java API
You want to configure and customize Share	Share Extensions
You would like to develop extensions to Share	Developing Share Extensions
You would like to know about the new UI framework Aikau	Aikau
You want to develop applications for iOS.	Alfresco Mobile SDK for iOS
You want to develop applications for Android.	Alfresco Mobile SDK for Android

Approaches to Alfresco Development (1 of 3)

- Configuration
- Information Architecture
- Templates
- Model Manager
- Content Rules
- Smart Folders
- Workflows with the Activiti Designer
- Protocols
- Mobile Profiles
- Replication / Hybrid Sync

Protocol Support

- Allows you to access content from 'legacy' client apps and desktops
- Alfresco Office Services
 - Allows Office Apps to seamlessly work with Alfresco
- WebDAV
 - Makes Alfresco act like a WebDAV Server
- FTP
 - Makes Alfresco act like an FTP Server
- CIFS
 - Makes Alfresco act like an SMB.CIFS Server
- IMAP
 - Allows mail clients like Microsoft Outlook to work with Alfresco
- SMTP
 - Out-bound and In-bound e-mail support

Approaches to Alfresco Development (2 of 3)

Out-of-Process (Remote APIs)

- Access Alfresco content and services from external applications
- Docs: “Integration Points”
- Content Management Interoperability Services (CMIS)
 - CMIS libraries in Java, C, PHP, .NET, Python, Ruby
- Alfresco Public REST
- Mobile SDK (Android and iOS)

ReST API

- Allows you to access content using HTTP
- ReST = Representational State Transfer
- We use ReST to provide access to Files and Folders through CMIS
 - Content Management Interoperability Services
 - OASIS Standard
 - XML/HTTP, JSON/HTTP and SOAP bindings
 - `/api/{tenant}/{scope}/cmis/versions/{version id}/{binding}/{binding specific portion}`
 - `/cmisws/cmis?wsdl/`
- We also use ReST to provide access to Sites, Favorites, Tasks, People...
 - `/ {tenant}/ {scope}/alfresco/versions/{version id}/{entity}/{id}/{relation}`

Mobile SDK

- Allows you to build your own iOS or Android App
- Available in GitHub
 - <https://github.com/Alfresco/alfresco-ios-sdk>
 - <https://github.com/Alfresco/alfresco-android-sdk>
- Supports iOS
 - Xcode 5.1.1+
 - A collection of useful classes including Session, Services, Model, FileManager and Utils
- ... and Android
 - Java 1.7+
 - A collection of useful classes including Authentication, Services and Exceptions

docs.alfresco.com → Community / Alfresco One → Developer guide → Platform integrations → Platform integration points

Platform integration points

The Alfresco platform features a number of integration points that can be used to integrate web clients, desktop clients, enterprise services, and applications with Alfresco.

The following table lists platform integration points and links to relevant documentation.

Integration point	Description
REST Clients	Most of the remote integrations will use the REST API . By using the REST API the client code can be written in any language that can make HTTP calls and that can process XML or JSON.
Java Clients	If the remote client is written in Java, then it might make sense to wrap the REST API in a Java library for convenience and easier usage. The Apache Chemistry OpenCMIS Java library can be used for most operations, except to manage Alfresco specific content such as sites, tags, and ratings.
Aikau Clients	If you are looking at the out-of-the-box Alfresco Share web client, and there are too many changes needed to turn it into the web client that you want, then you can build a new content management UI with the Aikau framework and content widgets.

Approaches to Alfresco Development (3 of 3)

In-Process (Embedded APIs)

- Modify the Alfresco platform
- Docs: “Extension Points”
- Repository Tier
 - Java API
 - Repository JavaScript API
 - Freemarker Template API
- Share Tier
 - Spring Surf API
 - Aikau Widget API

Java API

- A way of extending the Alfresco Platform, server-side, using Java
- Java Classes have access to the Service Registry, which is an index of services that the class can use ...

```
<bean id="myContentService" class="org.alfresco.tutorial.publicapiaccess.  
service.AcmeContentServiceImpl">  
    <property name="serviceRegistry">  
        <ref bean="ServiceRegistry" />  
    </property>  
</bean>
```

- The services available include things like the `ActionService` (e.g. Add an Aspect), `FileFolderService` (e.g. Copy a File) and `WorkflowService` (e.g. Start a Workflow)
- Deployment of the extension is done as a Repository AMP extension project or a Simple JAR module

Repository JavaScript API

- Access to content and services as JavaScript objects of type `ScriptNode`
- Objects such as ...
 - `document` ... the current document node
 - `companyHome` ... the Company Home
 - ... and many more
- Services such as ...
 - Site Service
 - Authority Service
 - ... and many more
- Deployed in the classpath or in the Repository
 - `/shared/classes/alfresco/extension/templates/webscripts`
 - `/Company Home/Data Dictionary/Scripts`

Repository Freemarker Template API

- Allows control over the rendering of content (the view in an MVC pattern)
- Widely used by Alfresco itself, e.g.
 - E.g. the repository uses templates to generate emails
- A template can generate any kind of text content
 - JSON, HTML, XML ...
- Template files are store in the classpath or in the Repository itself
 - /shared/classes/alfresco/extension/templates/webscripts)
 - /Data Dictionary/Web Script Extensions
- Templates have access to `TemplateNode` objects, e.g.
 - `person` ... the currently logged-in user
- ... and methods
 - `hasAspect` ... whether an object has a particular aspect

Spring Surf API

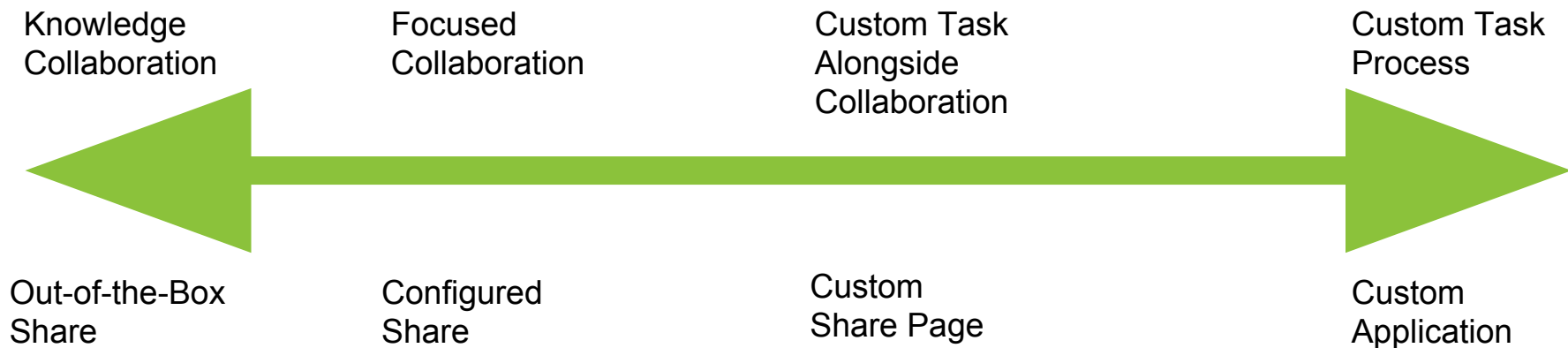
- Provides a way to build web components to extend the presentation tier
- Surf Framework
 - JavaScript API
 - FreeMarker API
- Have access to presentation tier root objects like ...
 - `user` ... the currently logged-in user
 - `page` ... the current page
 - `remote` ... a framework for executing HTTP requests
 - ... and many more
- No server restarts required when components are changed

The Aikau Widget API

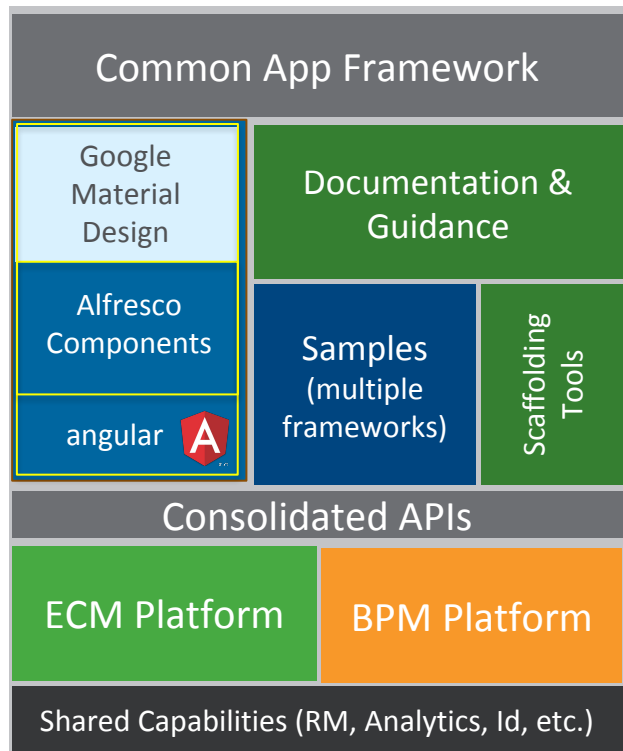
- Provides an easy way to build Web UI's that leverage Alfresco
- Widgets are written JavaScript, CSS and HTML
- Based on Dojo <https://dojotoolkit.org/>
 - but not restricted to Dojo
 - E.g. The Share Calendar uses YUI, JQuery and Dojo
- Aikau Widgets have access to helper services (e.g. `DocumentService`)
- Portions of Share are implemented as Aikau Widgets (e.g. Filtered Search Page, Site Management Page and the Analytics Widgets)
- No server restart is required when components are changed
- Aikau Source Code is available on GitHub <https://github.com/Alfresco/Aikau>
- ... and a Tutorial is available on Github <https://github.com/Alfresco/Aikau/blob/master/tutorial/chapters/About.md>

Key Decision: Customize Share?

- Share is a powerful collaboration interface
- Not all features are needed by every user
- Some use cases require a more focused interface



Current Focus on Time to Value: Unified Application Framework



Features:

- Used for new custom applications
- Reusable libraries for Angular 2 and Web Components
- Framework independent guidance, API's, samples, and tooling
- Design for mobile-first web apps

Aikau Continues:

- Share and Records Management continues to be extended with Aikau
- Leverages the lessons from Aikau
- Aikau will transition toward Angular.js interoperability

docs.alfresco.com

- Community / Alfresco One
- Developer guide
- Platform extensions
- Platform extension points



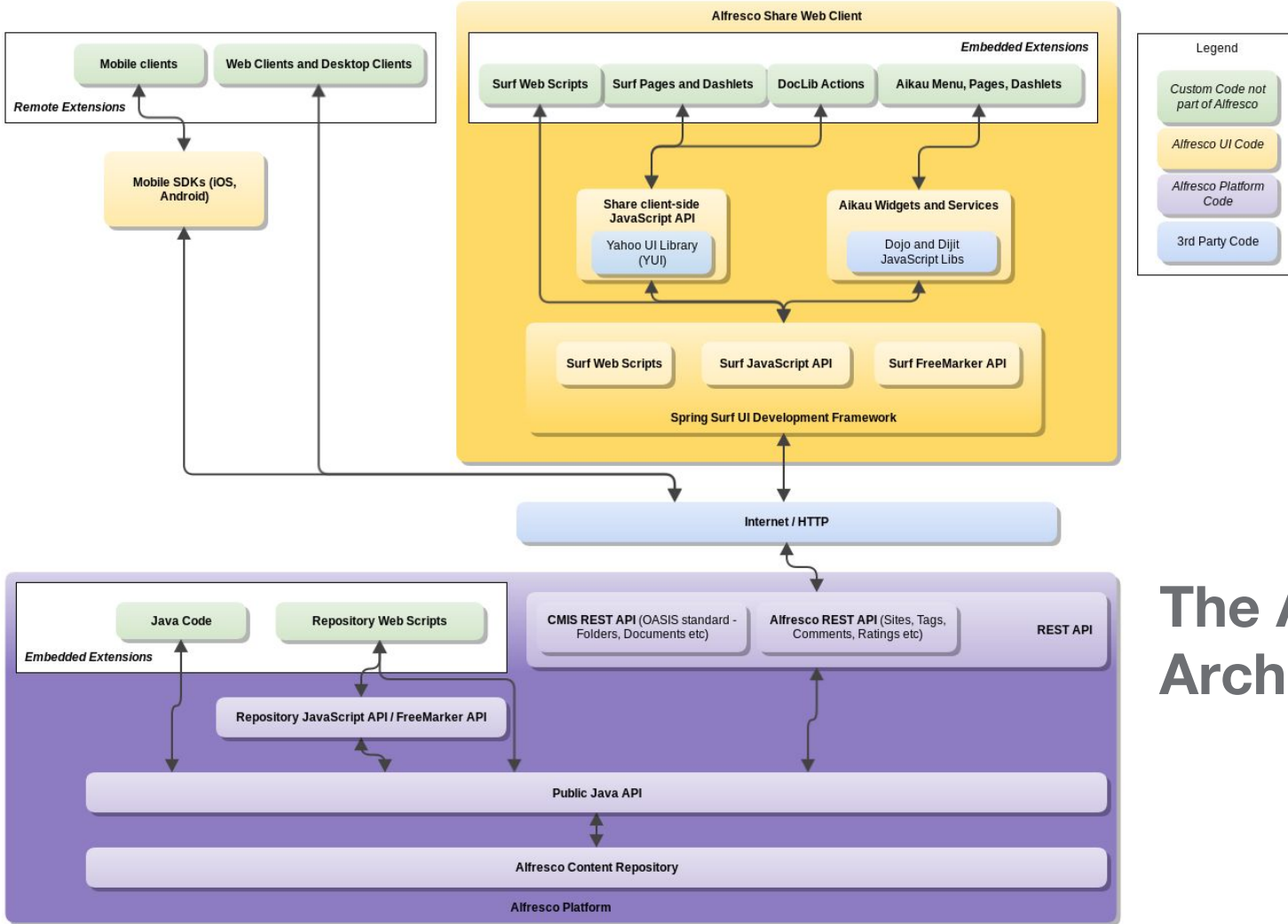
Platform extension points

The Alfresco platform features a number of extension points that can be used to create customizations. The table of supported extension points includes links to more information.

The following table lists platform extension points and links to relevant documentation:


Extension point	Description	Support Status
Content Model	Content modeling provides a foundation for structuring and working with content. It is used to create a domain specific model that can be used to classify content and refine the search capabilities.	Full Support
Actions	Actions are Spring beans that act upon a content node. You develop actions using Java and register them with the repository through a Spring configuration file. Actions provide a place to locate reusable business logic.	Full Support
Workflow	Alfresco One embeds the Activiti Workflow Engine as standard. You can create custom business workflows to manage your content and processes.	Full Support
Web scripts	Web scripts provide the ability to create custom REST APIs. A web script is implemented in XML, JavaScript, and FreeMarker. Java can also be used to implement a web script if the business logic requires it. If written in JavaScript and FreeMarker it is possible to write new extensions that do not require a server restart to take effect.	Full Support
JavaScript root objects	The JavaScript root object collection provides a ready made set of objects you can access from your web scripts. These objects provide access to the repository content. It is also possible to extend the platform with new custom JavaScript root objects that can be used in for example Web Scripts controllers.	Full Support
FreeMarker root objects	The FreeMarker root objects provide a collection of ready-made objects that you can work with from your FreeMarker code. These objects provides access to the Repository content. It is also possible to extend the platform with new custom FreeMarker root objects that can be used in for example Web Scripts templates.	Full Support
Behaviors / policies	Behaviors are logic that is tightly coupled to a repository event, such as adding content. Examples of out-of-the-box mechanisms that employ behaviors are versioning and auditing. Custom behaviors can be implemented to support features such as automatically adding a unique ID property to a content node when added to the repository, or automatically applying metadata attached to a folder to content stored in the folder.	Full Support
Bootstrap content	Most content management solutions require some content to be available before the system is going live. This can be users, groups, files and folders, sites, and so on. This content can be imported into the repository using a bootstrapping procedure.	Full Support
Permissions (Custom Roles)	Permissions and their groupings are defined in XML configuration files. The default files are found in the distribution configuration directory as <code>permissionDefinitions.xml</code> and	Full Support

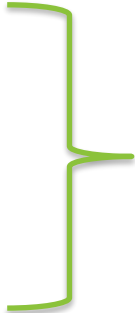
Let's Review



The Alfresco API Architecture

The Alfresco API Set

- Remote
 - ReST API
 - Protocols
 - Mobile SDK

Access Alfresco Content Services from Apps
- Embedded
 - Java API
 - Repository JavaScript API
 - Aikau Widget API
 - Spring Surf API
 - Freemarker Template API

Extend and Customize Alfresco

The Web Application Analogy

- Database → Content Repository
 - Schemas → Content Models
 - Stored Procedures → Web Scripts
- CMS Application → Share
 - Backend Framework → Surf
 - Front-end Framework → Aikau (legacy: YUI)

Every analogy has its limits.

The Web Application Analogy (Future Option)

- Database → Content Repository
 - Schemas → Content Models
 - Stored Procedures → Web Scripts
- CMS Application → Custom with Alfresco widgets
 - Backend Framework → Node.js (and others)
 - Front-end Framework → Angular 2

Every analogy has its limits.

A Few More Details

- API Lifecycle and Support
 - Remote API - Independently versioned and backward compatible
 - Embedded API – These don't change within a major version (e.g. 5.x)
 - Automated compliance testing with a Backwards Compatibility Kit (BCK)
 - Runs as part of the build to identify cases where API changes the API signature in a way that breaks compatibility
- Authentication
 - Remote
 - On-Premise
 - Basic Authentication and Alfresco tickets
 - Cloud
 - OAuth2
 - Embedded
 - Relies on the logged-in user authentication

A Closer Look at the SDK

Key Tools for Building Alfresco Extensions

- Maven (artifacts.alfresco.com)
- AMPs and JARs
- Web Scripts
- Actions (Repo & Share)
- Behaviors
- Transformers
- Extractors
- Scheduled Jobs
- Selenium Web Driver



SDK Architecture

- Maven archetypes defined in pom files
 - Repo AMP, Share AMP, All-in-One (AIO)
- Includes config, IDE config, and examples
- Maven plugins for loading and reloading Alfresco
- H2 database
- Apache Tomcat
- Alfresco Community Edition artifacts
- Spring Loaded JAR (put in the project run definition)
- Git repository of samples

Starting with the SDK (simplified)

- Install Maven
- Install Eclipse IDE for Java EE Developers
- Download from GitHub: [spring-projects/spring-loaded](https://github.com/spring-projects/spring-loaded)
- Create a new Maven project in Eclipse
- Select an Alfresco archetype
 - Add the Maven Central catalog
<http://repo1.maven.org/maven2/archetype-catalog.xml>
- Build an in-process extension in your new project
- Run it, change it, and run it without reloading (usually)

Key Maven Commands

`mvn {goal} -P{profile}`

`mvn integration-test -Pamp-to-war`

`mvn package`

`mvn install -Prun`

`mvn install -Pfunctional`

`mvn clean -Ppurge`

`mvn eclipse:eclipse`

Cautions

Maven downloads *lots* of stuff

Repo AMP run configuration needs extra memory
-Xms1024m -Xmx2048m

Versioning ([see compatibility matrix](#))

Upgrading ([see documentation](#))

SDK 2.2.0: Minimal updates to support Alfresco 5.1

- All-In-One AMP module names and ID now includes AIO name to make them unique
- Share AMP files prefixed to avoid name clashes when multiple Share AMPs are applied
- Aligned Module version with Artifact version - 5.0 and 5.1 supports SNAPSHOT versions in module.properties
- Bug fixes
- Updates to the documentation and samples

Limitations in SDK 2.2.0

- Spring Loaded Java agent only works with Share AMP
- Still adding samples and tests for all Extension Points
- No Maven archetype for simple JAR modules
 - Example in alfresco-sdk-samples →
Simple-module
- Only supports Alfresco 5.1

Learning More (1 of 3)

Official SDK Documentation

docs.alfresco.com → [Community / Alfresco One](#)
→ [Developer Guide](#) → [Alfresco SDK 2.2.0](#)

SDK Project (see the tutorial video in the README.md)

github.com/Alfresco/alfresco-sdk

SDK Samples

github.com/Alfresco/alfresco-sdk-samples

Learning More (2 of 3)

Alfresco Tech Talk Live #86: Alfresco SDK 2.1

youtube.com/alfresco101 → [Playlists](#) → [Tech Talk Live](#)

ECM Architect SDK Tutorial

<http://ecmarchitect.com/alfresco-developer-series-tutorials/maven-sdk/tutorial/tutorial.html>

Alfresco Developer Blog

blogs.alfresco.com → [Staff Blogs](#) → [Alfresco Developers](#)

Learning More (3 of 3)

Aikau Sandpit

<https://aikau-sandpit.alfresco.com/aikau-sandpit/page/na/ws/home>

API Explorer

<https://api-explorer.alfresco.com>

Alfresco Developer Forums

forums.alfresco.com

Alfresco Training (consider the ACE certification)

university.alfresco.com → [Training by Role](#) → [Developer](#)

Thank you to Greg Melahn for letting me use many of his slides.

These slides can be redistributed under the Creative Commons Attribution No Derivatives license.