Alfresco SDK

Past, present and future

Ole Hejlskov

Product Manager, Developer Platform & APIs
ole.hejlskov@alfresco.com
@OleHejlskov

Alfresco

Ole Hejlskov

Product Manager

ole.hejlskov@alfresco.com

@OleHejlskov

- Some of you may know me as **ohej**
- Works at Alfresco as Developer Evangelist
- **Transitioning into product management**
  - **I'm responsible for Developer Platform & API's**
- 6 years of experience working with Alfresco
- Co-founder of Order of The Bee
- Community contributor
  - SDK contributions
  - Aikau tutorials
  - Almost always online in #Alfresco @FreeNode
- Open Source enthusiast

Agenda

- About me
- The Good, The Bad and The Ugly
- The Past, Present and Future
- Demo
- Questions

The Good

- **Alfresco SDK has gained traction**
  - Easy to get started
  - Example code and tests
- **Hot reloading**
  - SDK 2.0 and 2.1 really improves productivity
  - More and more fixes in product rather than SDK
- **Integrates easily with IDEs**
  - Combined with hot reloading it gives the Save'n'refresh feel
  - Debugging is easier
- **Easy to get help**
  - If you are facing an issue with your code, having it all in an SDK project it is easy to send the whole project and have get help in the community
  - **This also applies to Developer Support!**
- **A lot of good resources**
  - SDK Video tutorial *(a bit old, but still valid)*
  - Revamped documentation

The Bad

- **Alfresco SDK was created before Alfresco was fully "mavenized"**
  - This means that a lot of hacks was made in the SDK
    - H2 support ended in limbo
  - Solr artifacts changed in 4.2.f, 5.0.c and 5.0.d
  - Solr SSL / Repo SSL hacks
    - Replacer plugin to comment out sections in web.xml
- Complex compatibility matrix

Compatibility matrix

Alfresco SDK has several versions and compatibility with Alfresco versions varies.

It is recommended you use the latest version of the Alfresco SDK where possible.

The following table shows compatibility between Alfresco SDK and versions of Alfresco.

Alfresco version	Maven Alfresco Lifecycle (deprecated)	Maven SDK 1.0.x (deprecated)	Maven SDK 1.1.x	Alfresco SDK 2.0.x	Alfresco SDK 2.1.x	Alfresco SDK 2.2.x
3.2.2 - 4.1.1.x	Compatible (but not supported)	Not available	Not available	Not available	Not available	Not available
4.1.x (x >= 2)	Not available	Compatible (but not supported)	Not available (SDK 1.1.0 does not work with Alfresco 4.1.2-4.1.5 using **Solr Search Subsystem**. It is possible to use Alfresco 4.1.6 and greater, or use Lucene Search Subsystem)	Not available	Not available	Not available
4.2.x	Not available	Not available	Compatible and supported	Not available	Not available	Not available
5.0 and 5.0.c	Not available	Not available	Not available	Compatible and supported	Not available	Not available
5.0.1+ and 5.0.d+	Not available	Not available	Not available	Compatible and supported	Compatible and supported	Not available
5.1+ and 5.1.d+	Not available	Not available	Not available	Not available	Not available	Compatible and supported

Note that Alfresco 4.1.x requires Java 6, Alfresco 4.2.x and Alfresco 5.0 require Java 7. Alfresco 5.0.1 and 5.0.d requires Java 7 or 8. Alfresco 5.1.0 and 5.1.d requires Java 7 or 8. Note also that Alfresco SDK works only on Linux, Windows or Mac.

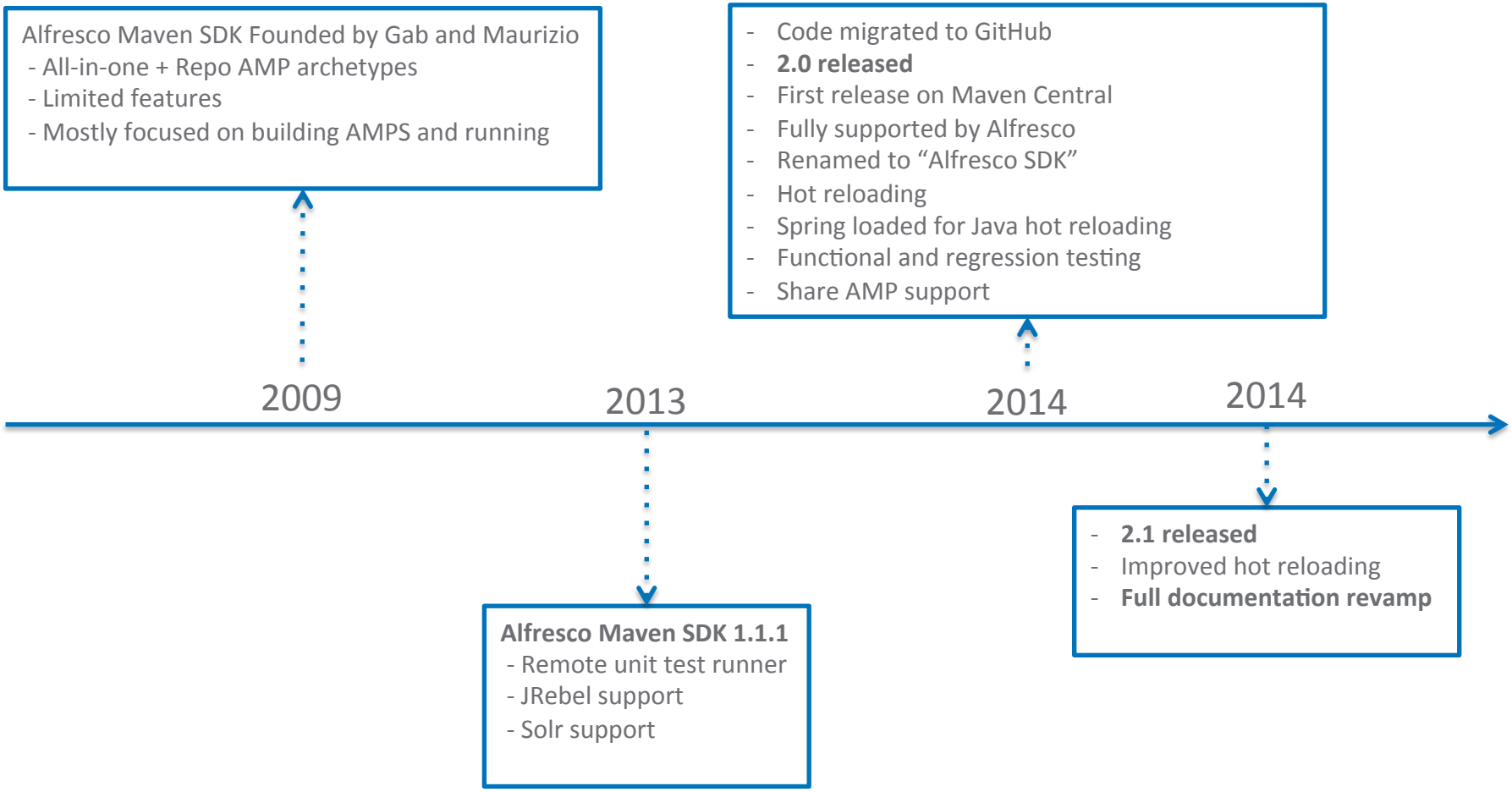
- **Alfresco SDK was created before Alfresco was fully "mavenized"**
  - This means that a lot of hacks was made in the SDK
    - H2 support ended in limbo
  - Solr artifacts changed in 4.2.f, 5.0.c and 5.0.d
  - Solr SSL / Repo SSL hacks
    - Replacer plugin to comment out sections in web.xml
- Complex compatibility matrix
- Java 7 vs Java 8
  - SDK 2.1 was Java8 only
  - SDK 2.1.1 reverted this so it once again works with Java7
  - SDK 2.2 is compatible with Java7 and Java8, but 5.1 only works with Java8
- SDK 2.2 is only compatible with Alfresco 5.1 and onwards

The Ugly

- **AMPs + Custom directory layout + hot reloading == Messy classpath**
  - alfresco-maven-plugin installs AMP into the WAR
  - Also copies everything to target/classes + target/test-classes
    - Same resources present in three locations, which one wins?
      - Resources are loaded twice (at least)
    - Will every IDE respect all the <resources/> instructions to make sure files are put in the right place?
      - **YES** – but it takes ~150 lines of Maven configuration
- **Archetype changes between almost every release**
- **Static copy of H2 dialect DB scripts causes incompatibility**
- **Upgrading a SDK project is a major pain**

The Past, Present and Future

The Past


Alfresco Maven SDK Founded by Gab and Maurizio
- All-in-one + Repo AMP archetypes
- Limited features
- Mostly focused on building AMPS and running
2009
2013
Alfresco Maven SDK 1.1.1
- Remote unit test runner
- JRebel support
- Solr support
- Code migrated to GitHub
- 2.0 released
- First release on Maven Central
- Fully supported by Alfresco
- Renamed to "Alfresco SDK"
- Hot reloading
- Spring loaded for Java hot reloading
- Functional and regression testing
- Share AMP support
2014
2014
- 2.1 released
- Improved hot reloading
- Full documentation revamp

The Present

- **2.2 released**
  - Number of bug fixes
  - Dedicated H2 artifact ***(Thanks Samuel!!)***
  - Remove Java8 dependency
    - However 5.1 is only supported on Java8
  - **ONLY Compatible with 5.1**
  - **No spring-loaded**
    - Security issue in spring-loaded causes trouble

- 2.1.1 not compatible with 5.1
  - Works if you manually patch with H2 scripts or use an external DB

Features of the Alfresco SDK 2.2 Today

- All-in-one, Repository AMP and Share AMP archetypes
- Hot reloading
  - ~~Java classes (spring loaded)~~
  - Static resources (client side Javascript and CSS)
  - Server side javascript
  - ~~Spring Beans~~
    - *Can be added by using JRebel, but we have seen a number of issues regarding behaviours being loaded multiple times*
- Dynamic “refresh web scripts” when compiling/process-resources
- IDE Integration
  - Eclipse and IntelliJ IDEA works mostly out of the box

- **2.1.1 is not compatible with 5.1.x**
- **2.2 is only compatible with 5.1 but not 5.0**
- 5.1 introduced a new table for authorization
  - The SDK contains a copy of the PostgreSQL DB scripts that H2 uses
    - The H2 scripts are now out dated
    - This issue also affects 4.2.5 and SDK 1.1.1 projects
  - **The solution was to create an H2 artifact that is released along with the repository – including community edition**
    - We now have this from 5.0.d and onwards
- SDK 2.1.1 ***can*** be modified to work with 5.1
  - Working example can be found here

The Future

Let’s step back for a minute..

What is the problem we are trying to solve?

"As a developer I need simple, agile, and future-proof tools to extend Alfresco so that I can meet the needs of my evolving business"

I expect to achieve this with

Modern, Industry-Standard Conventions and Tooling,
Comprehensive and Reliable Public APIs and
Supported Extension Mechanisms
That Work Across Versions of the Product

Requirements

- Lifecycle management
- Tooling
- Hot reloading
- Seamless integration with IDEs
- Based on standards and conventions
- Small footprint on projects
- Easy to upgrade
- Compatibility across versions
- **Users should not have to be a Maven expert to get the SDK running**

It all starts with a JAR

AMP vs JAR

Feature	AMP	JAR
3rd party dependencies	Yes	No
Control module load order	Yes	No
Isolated and separated class path	No	Yes
Transient dependency management	No	Yes
Works for both repo and share	Yes	Yes
Invasive/Complex Maven setup	Yes	No
Seamless integration with IDEs	No	Yes
Easy to create cross project dependencies	No	Yes

JAR Support in 5.1

- **JAR Support in 5.1**
  - ${ALF_HOME}/modules/[platform|share]
  - Simple module, no 3rd party dependencies
  - Include module.properties
  - Put static assets in META-INF/resources
    - **Servlet3 specs takes care of this**
  - Put web fragments in META-INF/web-fragment.xml
    - **Allows additive changes to web.xml**
  - Look at Alfresco SDK Samples in GitHub for references

- **5.1 still support AMPs – no change**
  - Install with MMT as usual
  - Supports 3rd party dependencies

JAR Support in 5.1

- **JAR Support in 5.1**
  - ${ALF_HOME}/modules/[platform|share]
  - Simple module, no 3rd party dependencies
  - Include module.properties
  - Put static assets in META-INF/resources
    - Servlet3 specs takes care of this
  - Put web fragments in META-INF/web-fragment.xml
    - Allows additive changes to web.xml
  - Look at Alfresco SDK Samples in GitHub for references

- **5.1 still support AMPs – no change**
  - Install with MMT as usual
  - Supports 3rd party dependencies

Neither AMPs or JARs are module systems!

Why not just go OSGi?

- OGSi would be a major effort and investment
- Some efforts has been tried in a community project
  - Extremely invasive
- Not all our dependencies are available
- The foundation for OSGi is a JAR

.. But Java 9 has this new cool thing

- It sure does, and we're following it closely… but it is simply not there yet

Alfresco SDK 3.0 is under development

- **JAR Support is a key feature**
  - Makes IDE integration effortless – *"it's a JAR, it just works"*
  - Standard directory layout – no more src/main/amp/*
- **AMP as an optional assembly**
- **Consolidate "runner" logic into a plugin**
  - mvn alfresco:run
  - Can run repo, solr and share – or each individually
- **Optional parent pom**
- **Testing** *(Unit, integration and functional testing)*
- **Hot reloading**
- **Integration with Alfresco SPK**
- **Yeoman generator – Thanks Bindu!!!!!**

Demo

JARs are just the first step

- **We are actively investigating a new module/plugin system for Alfresco**
- OSGi does not seem to be the answer, but it has not been fully ruled out
- By reverting to a JAR with AMPs as an optional assembly it becomes easy to support a new format while still ensuring backwards compatibility
- We do not currently have a timeline for the new module system

Questions?

Thanks!

</presentation>

@alfresco