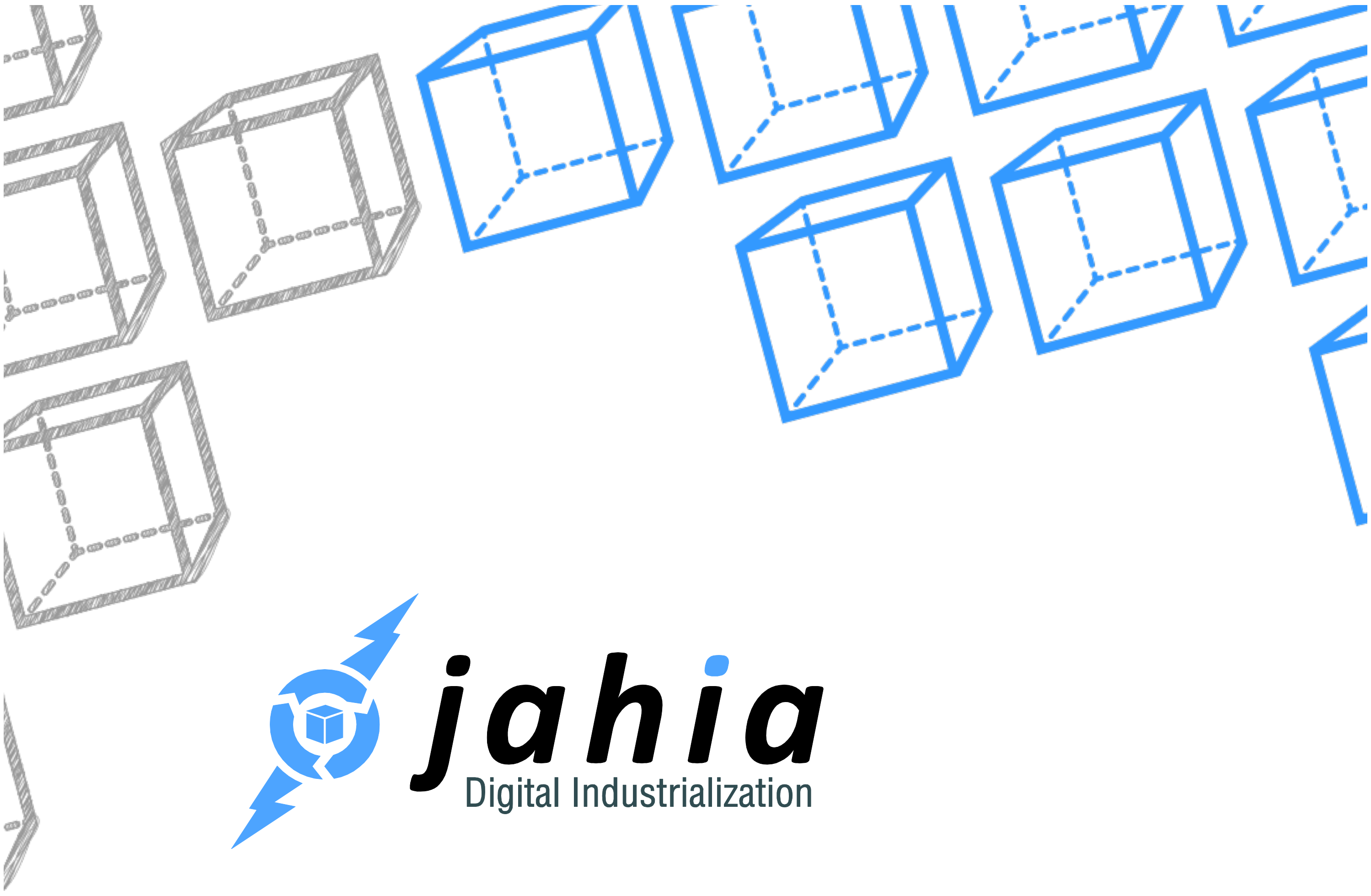


jahia
Digital Industrialization

DIGITAL FACTORY 7.0

Building native mobile apps for Digital Factory

Rooted in Open Source CMS, Jahia's Digital Industrialization paradigm is about streamlining Enterprise digital projects across channels to truly control time-to-market and TCO, project after project.

Jahia Solutions Group SA

9 route des Jeunes,
CH-1227 Les acacias
Geneva, Switzerland

http://www.jahia.com

Summary

1 Introduction 3
2 Target audience & assumptions 4
3 Native mobile applications 5
3.1 Native multi-platform frameworks 7
3.1.1 Adobe PhoneGap / Apache Cordova 8
3.1.2 Pugpig 8
3.1.3 Titanium 9
4 Additional resources 10

1 Introduction

This document presents one of the two main possibilities of how to address the mobile rendering needs for your CMS/WCM projects using Digital Factory. Our product has two technologies that can help in this regard:

- Channel-based rendering and editing
- REST API to build native clients

This guide covers the second technology. If you're interested in the first, building mobile web sites using Digital Factory, please read our companion document: Mobile web developer guide.

2 Target audience & assumptions

It this document it is assumed that the reader has a working knowledge of the following:

- Java programming
- JSP programming
- Jahia template development
- Mobile development

If you are not familiar with one of the above-mentioned fields, we recommend you get at least familiar with them. For Digital Factory template development, please check our documentation on http://www.jahia.com.

The first chapters will be more focus on presenting some of the concepts behind mobile application development using the Digital Factory platform, but the next ones will go into technical details as how to implement such solutions. A working knowledge of Digital Factory development and integration is therefore strongly recommended.

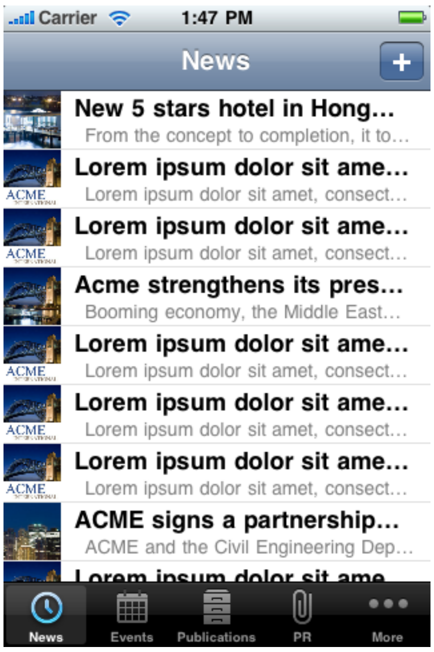
3 Native mobile applications

An alternative to building web sites that target mobile devices is developing a native mobile application. This is usually a tradeoff since building a mobile application means that it will be proprietary to the device's operating system. So the development of a native mobile application will not be reusable when targeting another device. But building a native application usually makes for a better user experience, as the interactions will be smoother and less dependent on network availability.

A native application must use some kind of protocol to communicate with a server, and in the case of integrating with Digital Factory, it is of course possible to request HTML renderings to display directly on the mobile screen using a web view. But if the data needs to be re-formatted or presented in different views in the native app, or if network traffic must be reduced to an absolute minimum, it is best to use Digital Factory's REST API.

Digital Factory REST API makes it possible to retrieve, create, edit and delete content simply by using basic HTTP requests that retrieve JSON content and use HTTP POST to perform content actions. This seemingly simple protocol is actually very powerful and most content actions you perform through Digital Factory's UI are also available through the REST API. And as it is possible to develop custom actions, any missing functionality may be implemented by developing custom actions packaged in custom modules.

An interesting example application illustrating the usefulness of the REST API is a prototype of a native iPhone/iPad application that was designed by Digital Factory. As this is a prototype that is not available publicly, if you are interested in learning more about it, please contact us at support@jahia.com. The native application connects to Digital Factory on startup, retrieves a configuration file that contains different screen properties, notably queries that will be executed with the REST find servlet. The results are generated using JSON output. The native application can then display the content anyway it chooses. Having the content in JSON format makes it easy to adapt to multiple screen sizes, as the native prototype is a dual iPhone and iPad application. The application is also capable of creating new content by using POST HTTP requests to create new content objects. All this is quite optimal and integrated with Digital Factory 's authentication as it also uses the login and logout servlets.



Carrier 1:47 PM

News

New 5 stars hotel in Hong…
From the concept to completion, it to…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Acme strengthens its pres…
Booming economy, the Middle East…

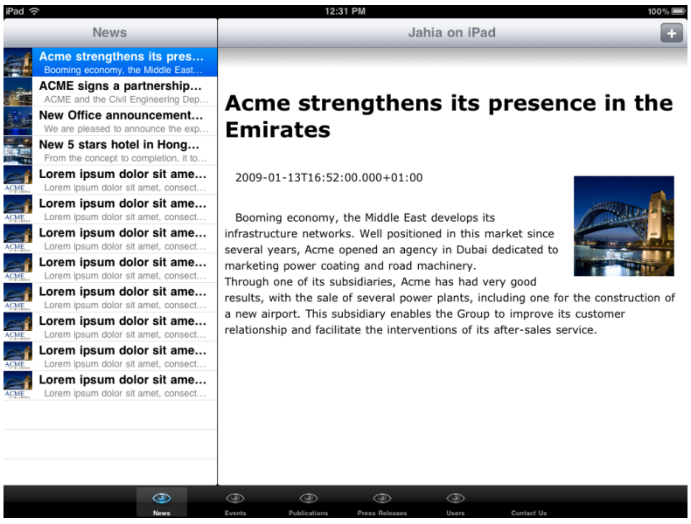
Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

ACME signs a partnership…
ACME and the Civil Engineering Dep…

As you can see in the above example, the native application displays the ACME demo content with native device rendering. This makes for a strong mobile user experience while having exactly the same content base as the regular web site. It is also possible to cache the data on the native client for offline browsing. The prototype demo application is also a native iPad application, as may be seen in the screenshot below:



iPad 12:31 PM 100%

News

Acme strengthens its pres…
Booming economy, the Middle East…

ACME signs a partnership…
ACME and the Civil Engineering Dep…

New Office announcement…
We are pleased to announce the exp…

New 5 stars hotel in Hong…
From the concept to completion, it to…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Lorem ipsum dolor sit ame…
Lorem ipsum dolor sit amet, consect…

Jahia on iPad

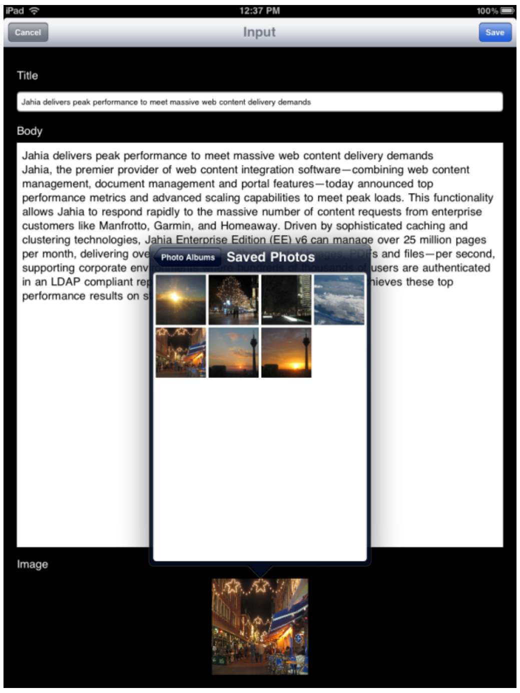
Acme strengthens its presence in the Emirates

2009-01-13T16:52:00.000+01:00

Booming economy, the Middle East develops its infrastructure networks. Well positioned in this market since several years, Acme opened an agency in Dubai dedicated to marketing power coating and road machinery.
Through one of its subsidiaries, Acme has had very good results, with the sale of several power plants, including one for the construction of a new airport. This subsidiary enables the Group to improve its customer relationship and facilitate the interventions of its after-sales service.

The iPad version displays a little differently, making good usage of the extra screen space, while retaining access to the same content objects as the iPhone and the public web site.

It also features an example of native data input, including text input as well as upload of binaries such as pictures. In the example below the user is selecting a picture from the picture gallery after entering a title and an article body:



Title

Jahia delivers peak performance to meet massive web content delivery demands

Body

Jahia delivers peak performance to meet massive web content delivery demands
Jahia, the premier provider of web content integration software—combining web content management, document management and portal features—today announced top performance metrics and advanced scaling capabilities to meet peak loads. This functionality allows Jahia to respond rapidly to the massive number of content requests from enterprise customers like Manfrotto, Garmin, and Homeaway. Driven by sophisticated caching and clustering technologies, Jahia Enterprise Edition (EE) v6 can manage over 25 million pages per month, delivering ove … s and files—per second, supporting corporate env … users are authenticated in an LDAP compliant rep … hieves these top performance results on s

Image

In this native prototype we have only scratched the surface of what is possible with REST API. Please refer to Digital Factory's documentation available on http://www.jahia.com/documentation-and-downloads/developers-techwiki to learn more about this and other technologies that will help you achieve your multi-channel objectives.

3.1 Native multi-platform frameworks

The main problem when building native a mobile application is that each device has it's own proprietary SDK, and these are often not compatible. For example the iOS SDK requires programming in the Objective C language, while the Android and Blackberry SDKs use Java as the main programming language. The provided libraries used in each SDK are quite different and incompatible, so it makes porting code from one platform to another very difficult, and often implies building or purchasing custom compatibility layers to obtain abstractions that help code portability. There is some common ground on the 3D side though, because often OpenGL is used as an underlying 3D framework, but this technology is also very low level and requires a lot of effort to build basic 3D applications.

Because of these porting difficulties, multi-platform frameworks have emerged to help code re-use across platforms, and we will now illustrate these with a few examples that might prove interesting to integrate in native mobile projects.

3.1.1 Adobe PhoneGap / Apache Cordova

One of the best-know native multi-platform frameworks is Adobe PhoneGap. The main concept being this framework is quite simple: to use HTML and Javascript to build native mobile applications. This is possible because the framework simply provides minimal native code to expose a built-in web browser, and exposes as Javascript objects native device functions such as camera control, accelerometer, address book access, and so on. So in effect you can build a native mobile application by developing HTML and Javascript, and then generate each native mobile application using the provided native code from the PhoneGap framework. One important thing to mention is that the PhoneGap source code has been entirely contributed to the Apache Foundation under the name “Apache Cordova”, and therefore this platform is available under the Apache License, which is 100% business friendly.

Also, it is possible to combine PhoneGap with jQuery Mobile, making it very easy to build native mobile applications that look and feel great using a lot less development effort. If you’re interested in learning more, we have a webinar recording available on our website.

3.1.2 Pugpig

Pugpig is another alternative to building mobile cross-platform solutions, aimed more precisely at the publishing industry. Using Pugpig you can use HTML 5 to easily build magazines or other digital packages to publish content on the go. Digital Factory is capable of integrating with Pugpig by generating and packaging digital bundles that can then be compiled into a native application. At the same time, it is also possible to use Digital Factory’s REST API to access content dynamically from a digital bundle distributed in a Pugpig native mobile application. Again, we have a webinar recording available if you are interested in learning more about this type of framework integration.

3.1.3 Titanium

The Titanium Mobile Development platform takes a slightly different approach to cross-platform native mobile development. Instead of using HTML and Javascript as a development platform, it actually uses Javascript to directly build native UIs on each platform. The way this works is that they have built a Javascript interpreter into each native code layer, and then provide Javascript abstraction libraries to build UIs using the platform's native UI elements. So there is no direct usage of the built-in browser technology, and it is possible to build applications that really look and feel like the real thing on each platform, although they will not be as fast as a real native application since they are still running in an interpreter. It is also possible to access native API using Titanium's Javascript engine, making it possible to integrate very tightly with a specific mobile platform if desired.

4 Additional resources

We present in this section some references we have presented in this document as well as additional reading material that may be interesting to learn more about mobile technologies.

- Digital Factory's Mobile web developer guide, http://www.jahia.com
- Wireless Universal Resource File, WURFL, http://wurfl.sourceforge.net/
- Apache Mobile Filter, http://www.idelfuschini.it/it/apache-mobile-filter-v2x.html
- Apache Mobile Filter sample code, http://www.idelfuschini.it/it/sample-code.html
- Apple iOS developer center, http://developer.apple.com/ios
- Google Android developer center, http://developer.android.com/index.html
- Blackberry developer center, https://developer.blackberry.com
- CSS media queries, http://www.w3.org/TR/css3-mediaqueries/
- Adobe Phone Gap, http://phonegap.com
- Apache Cordova, http://incubator.apache.org/cordova/
- Pugpig, http://pugpig.com
- Titanium, http://www.appcelerator.com
- jQuery Mobile, http://jquerymobile.com
- Bootstrap, http://twitter.github.com/bootstrap/

jahia
Digital Industrialization

Jahia Solutions Group SA

9 route des Jeunes,
CH-1227 Les acacias
Geneva, Switzerland

http://www.jahia.com