Table of Contents

Using Javascript to integrate custom Cordova plugins for native mobile applications

Cordova plugins are components that provide access to some built-in features of mobile phones, (such as camera or contacts), for which there is no Javascript access. When components are integrated into the system these features become available through some special Javascript functions that the plugin makes available to the developer. Cordova plugins can only be used in native mobile applications.

AwareIM integrates some Cordova plugins out-of- the-box and provides rule actions that activate them (for example, `MOBILE PUSH` or `MOBILE CAMERA SNAP INTO`. However, there are many plugins around and it is impossible to integrate all of them into ***AwareIM***.

Still there is a way to do this by adding some custom Javascript to your application. The following section explains how to do it.

This is the high level overview of what needs to be done:

1. Study the documentation of the Cordova plugin to fully understand Javascript methods that it exposes
2. Write the Javascript that calls the appropriate Javascript function that the plugin provides
    1. Give this function the data obtained from AwareIM if necessary. For example, read the data from the database and provide this data to the function of the plugin. The useful ***AwareIM*** function that can be used here is `AwareApp.getObjectData()`
    2. Handle the return of this function if necessary – for example write the data returned by the function to the database. Useful AwareIM functions for this are `AwareApp.createOrUpdateObject()` and `AwareApp.startProcessWithInit()`
3. Define panel operations or menu items in the mobile part of your business space version (using the Configuration Tool) that would run this Javascript. You should select an operation or menu item of the “Execute Javascript” type for this.
4. Build a native mobile application for your business space version using the “Build Native Mobile Application” command in the Configuration Tool. This will create a zip file.
5. Uzip this zip file somewhere. Find the `config.xml` file in the root of the unzipped application and open it for editing.
6. Find the section in this file that lists the plugins used by the application, for example:`<plugin name=“cordova-plugin-camera” spec=“2.0.0”/>`
   Add the definition of the Cordova plugin you need to integrate – look up the documentation of the plugin for details of the plugin name and version number
7. Zip up the application again and use the PhoneGap build to create application files in the native format of the mobile phone

Let’s look at an example. We will be integrating a Cordova plugin for Contacts into the CRM mobile sample application.

The documentation of the plugin can be found here: https://github.com/apache/cordova-plugin-contacts

As we can see the plugin provides the navigator.contacts object that can be used to create contacts, find existing contact or pick a particular one. Let’s add the following functionality to the CRM application:

- From the form of a customer or from a customer list create a phone contact populated with the information from the customer record in the application
- Pick a contact from the list of phone contacts and send this contact an email that includes some information stored in the application

- Creating a contact on the phone
- Send email to the selected contact

From:
http://www.awareim.com/dokuwiki/ - **Documentation**

Permanent link:
http://www.awareim.com/dokuwiki/docs/3500/0800/0890

Last update: **2025/06/12 04:00**