



Adobe Summit 2015 Lab 712:

Building Mobile Apps: A PhoneGap Enterprise Introduction for Developers

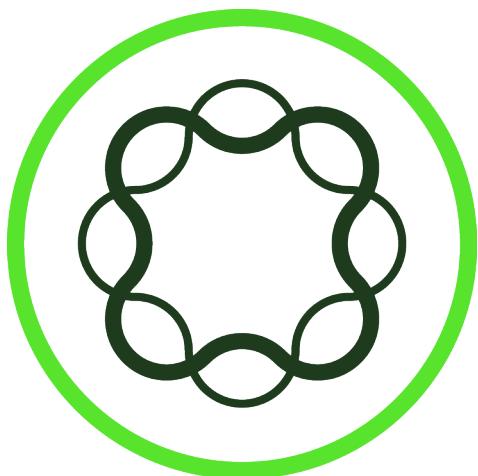




Table of Contents

INTRODUCTION	5
MODULE 1 – AEM & PHONEGAP ENTERPRISE INTRODUCTION	6
LESSON 1- AEM BASICS	6
OVERVIEW	6
OBJECTIVES	6
EXERCISE 1 – STARTING AEM	6
EXERCISE 2 – LOGGING IN TO AEM AUTHOR	7
EXERCISE 3 – REVIEW THE AEM AUTHOR UI	8
EXERCISE 4 – (OPTIONAL) REVIEW THE AEM PUBLISH INSTANCE’S GEOMETRIXX OUTDOORS SITE	8
OUTCOME	8
LESSON 2 - AEM APPS BASICS	9
OVERVIEW	9
OBJECTIVES	9
EXERCISE 1 – OPENING AND VIEWING AN EXISTING MOBILE APP IN AEM	9
EXERCISE 2 – EDIT THE APPLICATION NAME AND DESCRIPTION	10
EXERCISE 3 – (OPTIONAL) EXPLORE THE OTHER APP DETAILS AND TILES	11
OUTCOME	11
MODULE 2: PHONEGAP INTRODUCTION	12
LESSON 1 - PHONEGAP BASICS	12
OVERVIEW	12
OBJECTIVES	12
EXERCISE 1 – PHONEGAP APPLICATIONS	12
LESSON 2 - PHONEGAP COMMAND LINE	12
OVERVIEW	12
OBJECTIVES	12
EXERCISE 1 - BUILDING A MOBILE APPLICATION WITH PHONEGAP CLI	12
EXERCISE 2 - RUN A BASIC PHONEGAP APP	13
EXERCISE 3 (OPTIONAL) – RUN A BASIC PHONEGAP APP VIA XCODE	14
OUTCOME	15
MODULE 3: APP DEVELOPMENT	16
LESSON 1 – PROJECT STRUCTURE	16
OVERVIEW	16
OBJECTIVES	16
EXERCISE 1 – PROJECT OVERVIEW	16
LESSON 2 – COMPONENT DEVELOPMENT	16
OVERVIEW	16
OBJECTIVES	16
EXERCISE 1 – CREATE CAMERA COMPONENT	17



EXERCISE 2 – ADD JAVASCRIPT TO CAMERA COMPONENT	17
EXERCISE 3 – ADD HTML TO CAMERA COMPONENT	17
LESSON 3 – PROJECT INSTALLATION	17
OVERVIEW	17
OBJECTIVES	18
EXERCISE 1 – INSTALLING A PROJECT TO AEM	18
OUTCOME	18
<u>MODULE 4: MANAGING APP CONTENT</u>	<u>19</u>
LESSON 1 – AEM AUTHORIZING BASICS	19
OVERVIEW	19
OBJECTIVES	19
EXERCISE 1 – CREATING A PAGE IN AEM	19
EXERCISE 2 - EDIT A PAGE IN AEM	21
EXERCISE 3- (OPTIONAL) PREVIEW IN MOBILE EMULATOR	24
OUTCOME	24
<u>MODULE 5: REVIEWING APP CONTENT</u>	<u>25</u>
LESSON 1 – PHONEGAP CLI	25
OVERVIEW	25
OBJECTIVES	25
EXERCISE 1 – DOWNLOAD CLI ASSETS	25
EXERCISE 2 – BUILDING VIA PHONEGAP CLI	25
LESSON 2 – PHONEGAP ENTERPRISE VIEWER	26
OVERVIEW	26
OBJECTIVES	26
EXERCISE 1 – LAUNCH VIEWER	26
EXERCISE 2 – REFRESHING AN APP	26
EXERCISE 3 (OPTIONAL) – STAGING AN APP	27
OUTCOME	28
<u>MODULE 6: ADOBE MOBILE SERVICES ANALYTICS</u>	<u>29</u>
LESSON 1 – ADD ANALYTICS TRACKING	29
OVERVIEW	29
OBJECTIVES	29
EXERCISE 1 – ADD TRACKING CAPABILITY TO YOUR APP	29
EXERCISE 2 – ADD TRACKING TO A COMPONENT	29
OUTCOME	30
LESSON 2 – ADD ANALYTICS REPORTING	30
OVERVIEW	30
OBJECTIVES	30
EXERCISE 1 – ADD AN ADOBE MOBILE SERVICES CONFIGURATION	30
EXERCISE 2 – ADD AN ADOBE MOBILE SERVICES CONFIGURATION TO YOUR APP	30
OUTCOME	31



LESSON 3 – TEST ANALYTICS REPORTING	31
OVERVIEW	31
OBJECTIVES	31
EXERCISE 1 – LAUNCH BLOODHOUND	31
EXERCISE 2 – TEST THE APPLICATION	32
OUTCOME	32
APPENDIX	33
PHONEGAP	33
APPENDIX A – INSTALLING NODE	33
APPENDIX B – INSTALLING PHONEGAP CLI	33
APPENDIX C – PHONEGAP BUILD	33



Introduction

Adobe Experience Manager (AEM) Apps and PhoneGap Enterprise bring together the simplicity and ease-of-use of the AEM authoring environment with the power and portability of PhoneGap mobile applications. Using AEM Apps you can create, author, and update your applications within AEM. With PhoneGap Enterprise you can choose from a range of pre-built app components, build your app across multiple mobile platforms, and test your apps within the PhoneGap Enterprise mobile application.

In this lab you will be introduced to Adobe Experience Manager and PhoneGap, learn how to quickly and easily build your own mobile application, how to test it in simulators for a range of devices, and how to track app usage.



Module 1 – AEM & PhoneGap Enterprise Introduction

This module will provide a general introduction to Adobe Experience Manager and PhoneGap Enterprise

Lesson 1- AEM Basics

Overview

Adobe® Experience Manager helps you organize, create, and manage the delivery of creative assets and other content across your digital marketing channels, including web, mobile, email, communities, and video.

Objectives

- Learn how to start AEM Author and Publish Servers
- Learn how to log in to AEM
- Learn the key parts of the easy-to-use AEM interface

Exercise 1 – Starting AEM

1. Find the AEM folder on the desktop (named **AEM-Apps-Lab**)
2. Open up the author folder and double-click the quickstart jar (named **aem-author-4502.jar**) to start the AEM author server
 - a. When the AEM author server has started your browser window/tab will automatically open to <http://localhost:4502/> (this may take a few minutes)



3. Open up the publish folder and double-click the quickstart jar (named **aem-publish-4503.jar**) to start the AEM publish server
 - a. When the AEM publish server has started your browser window/tab will automatically open to <http://localhost:4503/> (this may take a few minutes)

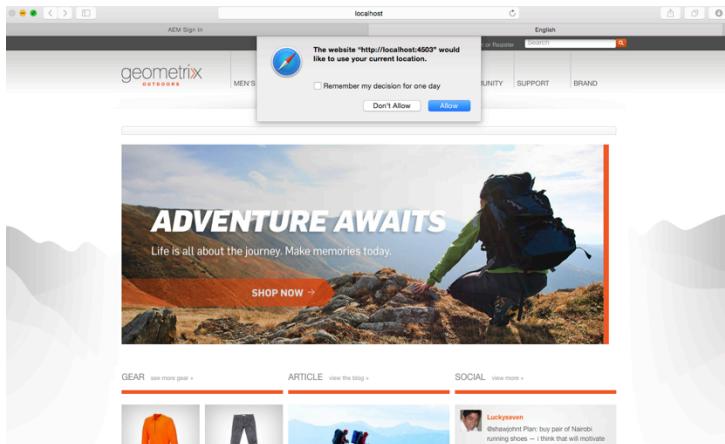




4. Wait for the author server to start (Login screen will be shown)



5. Wait for the publish server to start (Geometrixx Outdoors Site will be shown)
 - If the browser asks to user your current location click '**Allow**' to dismiss the dialog



Exercise 2 – Logging in to AEM Author

1. Go to the browser window/tab that shows the AEM Author login (<http://localhost:4502/>)
2. Enter the user name 'admin'
3. Enter the password 'admin'
4. Click '**Sign In**'



5. Click '**Cancel**' when prompted by the 'Analytics and Targeting configuration' dialog



Exercise 3 – Review the AEM Author UI

The screenshot shows the AEM Author UI with various UI components highlighted by yellow ovals:

- Header Bar & Breadcrumb**: Located at the top right, it includes the Adobe logo, user profile, and navigation links.
- Action Bar**: Located below the Header Bar, it contains buttons for Timeline, Select, Create, and other actions.
- Rail**: The left sidebar, highlighted with a green oval, contains a list of core product areas and sub-areas.
- Content Pane**: The main area displaying the project card, highlighted with a yellow oval.

1. Navigate around the landing page to get an idea of the content and layout.
2. Review Header Bar
 - Marketing Cloud link
 - Rail collapse icon
 - Bread crumb
 - Help
 - Notifications
 - User settings
3. Review Rail menu options
 - Core product areas and sub areas
 - Can be collapsed
4. Review Bread Crumb
 - Trail of path taken
 - Path is linked to allow fast
5. Review Action Bar
 - Actions on selected items from Content Pane
6. Review Content Pane
 - Display content, views can vary and include; card view, list view, details view etc.

Exercise 4 – (Optional) Review the AEM Publish instance's Geometrixx Outdoors Site

1. Go to the browser window/tab that shows the Geometrixx Outdoors Site (<http://localhost:4503/>)
2. Navigate through the site and explore the sample web site

Outcome

Congratulations! You are now familiar with the basics of Adobe Experience Manager.



Lesson 2 - AEM Apps Basics

Overview

AEM Apps enables businesses to deliver more effective apps by bridging the gap between marketers and developers. Mobile apps can be developed to reach the broadest audience with Adobe PhoneGap technology and plugins. Marketers can quickly update mobile apps with engaging content and features to continuously drive engagement to meet business goals.

Objectives

- Explore the AEM Apps Dashboard
- Edit application details

Exercise 1 – Opening and viewing an existing mobile app in AEM

1. Open the AEM authoring interface at <http://localhost:4502> in your browser
2. Navigate to the Apps section from the side rail and view the list of mobile apps
3. Click the Geometrixx app card to open its dashboard
4. Explore the Apps Dashboard
 - a. Details Tile
 - Manage app metadata and assets
 - b. Metrics Tile
 - Report and manage Adobe Analytics integration
 - c. Content Tile
 - Manage your app content
 - d. Content Release Tile
 - Manage your app updates
 - e. Push Notifications Tile
 - Manage app push integration and send push notifications
 - f. PhoneGap Build Tile
 - Manage PhoneGap Build integration and perform remote builds

The screenshot shows the AEM Marketing Cloud interface. The left sidebar is the Experience Manager navigation bar with various sections like Projects, Sites, Apps, Publications, Forms, Assets, Personalization, Communities, Commerce, and Tools. The 'Apps' section is currently selected. On the right, there's a detailed view of the 'Geometrixx Mobile App'. The main card displays the app's title 'Geometrixx AEM Team', version '1.0', compatibility 'PhoneGap 3.6.3', and update URL 'http://localhost:4503/'. It also lists platforms (iOS, Android, Web) and a description: 'Sample to illustrate building a single page app using AEM'. Below this are sections for 'Metrics' (warning: 'Please select a Mobile Service configuration to view reports'), 'Content' (listing 'Geometrixx Mobile App - English' with 92 pages), 'Content Sync Packages' (listing 'Geometrixx Mobile App' and 'Geometrixx Mobile App - English' both pending), and 'Push Notifications' and 'PhoneGap Build' sections.



Exercise 2 – Edit the application name and description

1. Drill into the Details Tile by clicking at the bottom right of the tile
2. Click on **Edit** in action bar
3. Scroll down to **Common Metadata** section
4. Change the name of the app
5. Change the description of the app

The screenshot shows the 'Details' page for an app named 'Geometrixx Summit Lab'. The 'Description' field contains the text: 'Sample to illustrate building a single page app using AEM for Summit 2015'. This field is currently selected, indicated by a blue border around its input box.

Name	Author
Geometrixx Summit Lab	AEM Team

Description	Author Email	Author URL
Sample to illustrate building a single page app using AEM for Summit 2015		http://adobe.com

App Version	Privacy URL
1.0.0	

PhoneGap Version	Content Update Server URL	Starting Page
3.6.3	http://localhost:4503/	content/phonegap/geometrixx-outdoors/en.html

6. Click on **Done** in the action bar
7. Notes the changes in the app details

The screenshot shows the 'Details' page for the 'Geometrixx Summit Lab' app. The 'Description' field now displays the updated text: 'Sample to illustrate building a single page app using AEM for Summit 2015'. The 'App ID' is listed as 'info.geometrixx.outdoorsapp'.

App ID:	info.geometrixx.outdoorsapp
App Version:	1.0.0
PhoneGap Version:	3.6.3
Update URL:	http://localhost:4503/

8. Click back < from the bread crumb in header
9. Note the changes in the Details tile as seen in the app dashboard



Adobe Marketing Cloud

Experience Manager

Projects

Sites

Apps

Publications

Forms

Assets

Personalization >

Communities >

Geometrixx Mobile App

Details

Geometrixx Summit Lab
AEM Team

Version: 1.00
Compatibility: PhoneGap 3.6.3
Update URL: http://localhost:4503/
Platforms:

DESCRIPTION

Sample to illustrate building a single page app using AEM for Summit 2015

Metrics

Please select a Mobile Service configuration to view reports

Content

TITLE	PAGES	MODIFIED
Geometrixx Mobile App - English	92	Administrator 5 months ago

Exercise 3 – (Optional) Explore the other app details and tiles

Outcome

Congratulations! You now know how the Apps Dashboard can be used to manage an app's entire life cycle.



Module 2: PhoneGap Introduction

Lesson 1 - PhoneGap Basics

Overview

PhoneGap is a framework that allows you to create mobile apps using standardized web APIs for the platforms you care about. It is based upon the Apache Cordova open source project.

Objectives

- Understand what PhoneGap offers to mobile application developers
- Become familiar with the structure of a PhoneGap app

Exercise 1 – PhoneGap Applications

1. Visit the PhoneGap developer website at <http://docs.phonegap.com/> and identify the plugin APIs that might facilitate a unique and compelling mobile experience for you.
2. Take a look at the individual API plugin pages at http://docs.phonegap.com/en/4.0.0/cordova_plugins_pluginapis.md.html and see if you can understand how the plugins work.
3. (Optional) Visit the template PhoneGap application at <https://github.com/phonegap/phonegap-start> and look at the key constituent parts:
 - a. phonegap-start / www / index.html
 - b. phonegap-start / www / config.xml
 - c. phonegap-start / www / js / index.js

Lesson 2 - PhoneGap Command Line

Overview

The PhoneGap command-line interface is a tool that allows you to create new projects, build them on different platforms, either locally or on a remote server, and run them within an emulator or device. You can also use the CLI to initialize project code, after which you use various platforms' SDKs to develop them further.

Objectives

- Understand the PhoneGap CLI environment
- Create an app from scratch with PhoneGap CLI
- Test the application in the simulator

(Building locally requires having native SDKs installed for targeted platforms.)

Exercise 1 - Building a mobile application with PhoneGap CLI

In this exercise you will create a PhoneGap application for iOS.

1. Create a basic app by typing the following into the Terminal:
`phonegap create my-lab-app
cd my-lab-app`



2. Inspect the created files in Finder. In the Terminal type `open .` (open followed by a space and a period) to view them in Finder.

my-lab-app				
Name	Date Modified	Size	Kind	
► .cordova	Today, 3:03 PM	--	Folder	
► .DS_Store	Today, 3:04 PM	6 KB	Document	
► config.xml	Today, 3:03 PM	5 KB	XML text	
► hooks	Today, 3:03 PM	--	Folder	
► platforms	Today, 3:03 PM	--	Folder	
► plugins	Today, 3:03 PM	--	Folder	
▼ www	Today, 3:03 PM	--	Folder	
► css	Today, 3:03 PM	--	Folder	
icon.png	Today, 3:03 PM	11 KB	PNG image	
► img	Today, 3:03 PM	--	Folder	
index.html	Today, 3:03 PM	2 KB	HTML text	
► js	Today, 3:03 PM	--	Folder	
► res	Today, 3:03 PM	--	Folder	
► spec	Today, 3:03 PM	--	Folder	
spec.html	Today, 3:03 PM	3 KB	HTML text	

Exercise 2 - Run a basic PhoneGap app

1. Build and run the app just created by typing the following in the Terminal:
`phonegap run ios`



2. Close the app by pressing SHIFT+CMD+H (or Menu: Hardware>Home)
3. Close the simulator by pressing CMD+Q (or Menu: iOS Simulator>Quit iOS Simulator)
4. Edit the app content:



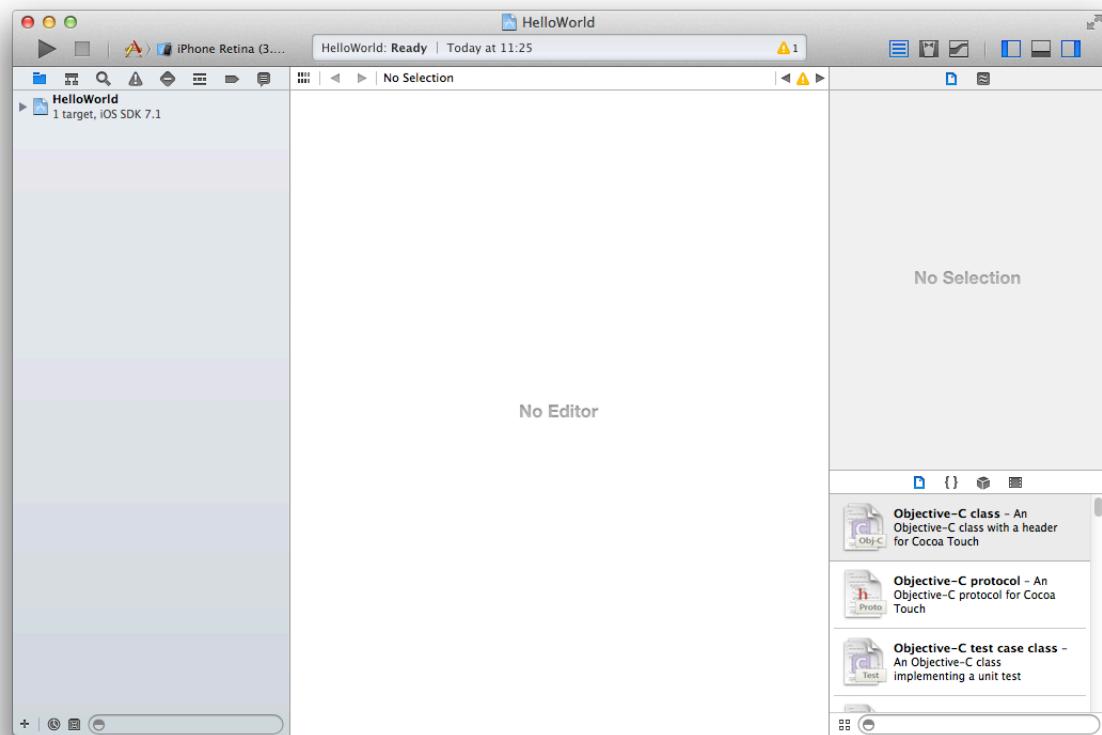
- a. In the Finder, find the file **www/index.html** (in **my-lab-app**) and CTRL-click on it, then select Open With > Xcode
- b. Change 'Connecting to Device' to 'Hello <YOURNAME>. Connecting to Device'
- c. Change 'Device is Ready' to 'Hello <YOURNAME>. Device is Ready'
- d. Save the document by typing CMD-S (or Menu: File > Save)
- e. Run the app to view your changes by typing in the Terminal:
`phonegap run ios`
- f. Confirm the app now shows your name.

```
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="format-detection" content="telephone=no" />
    <!-- WARNING: for iOS 7, remove the width=device-width and height=device-height attributes. See https://issues.apache.org/jira/browse/CB-4323
    -->
    <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height,
      target-densitydpi=device-dpi" />
    <link rel="stylesheet" type="text/css" href="css/index.css" />
    <title>Hello World</title>
  </head>
  <body>
    <div class="app">
      <h1>PhoneGap</h1>
      <div id="deviceready" class="blink">
        <p class="event listening">Hello LAB! Connecting to Device</p>
        <p class="event received">Hello LAB! Device is Ready</p>
      </div>
    </div>
    <script type="text/javascript" src="phonegap.js"></script>
    <script type="text/javascript" src="js/index.js"></script>
    <script type="text/javascript">
      app.initialize();
    </script>
  </body>
</html>
```

5. (Advanced) Change simulated device:
 - a. Menu: Hardware>Device> ...
 - b. From the home screen, drag right to left with the mouse and click the app icon to re-open it
6. (Advanced) Debug in browser:
 - a. Open the Safari web browser
 - b. Preferences > Advanced > tick "Show Develop menu in menu bar"
 - c. Menu: Develop > iPhone Simulator > index.html
7. (Advanced) Create some additional content:
 - a. Create an additional HTML page 'my-lab-app/www/page1.html'
 - b. Open my-lab-app/www/index.html and create an html hyperlink to your new page
 - c. Run the app to view your changes:
`phonegap run ios`

Exercise 3 (optional) – Run a basic PhoneGap app via Xcode

1. Find the generated Xcode project **my-lab-app/platforms/ios/HelloWorld.xcodeproj** and open it by double-clicking it.
2. Click the play icon in the top-left corner to build and run the app in the simulator.
 - a. If prompted to "Enable Developer Mode on this Mac", select "Enable" and enter the provided password.



Outcome

Congratulations! You have built your first mobile application – without writing a single line of code. You should now be familiar with the basics of PhoneGap and PhoneGap Build:

- understanding the access to native device functionality that PhoneGap provides
- familiarity with the different native APIs that are available
- knowing that PhoneGap apps are comprised of HTML, JavaScript, CSS, and some configuration
- knowing that PhoneGap apps do not require any native code (such as Objective C or Java) to be written



Module 3: App Development

Lesson 1 – Project Structure

Overview

Creating a new PhoneGap Enterprise App starts with a well-defined project structure that not only simplifies installation to AEM but also enables web developers to continue using the same tools they are familiar with.

Objectives

- Explore an app's project structure

Exercise 1 – Project Overview

The project used by this lab includes a folder structure for developing a typical AEM application and installing resources to the server. This allows the developer to work with the tools they are accustomed to yet still provide a simple means to deploy changes to AEM through packages.

Note: The source for this project can be found at the following GitHub repository:

<https://github.com/Adobe-Marketing-Cloud-Apps/summit-developer-lab>

1. Open lab project in Finder
 - a. cd summit-developer-lab
 - b. open .
2. Navigate to /content/src/main/content/jcr_root
3. Explore the directory structure that will be installed to AEM
 - a. **/apps**: location for components and templates used by app
 - b. **/etc/designs/phonegap/summit/developer-lab**: the design used by the app (ie. Styles)
 - c. **/content/dam**: assets used by the app (ie. Images)
 - d. **/content/phonegap/summit-developer-lab**: all of the page content for the app
4. What component does the page template for the app use?
5. What clientlibs is the app's design dependent on?

Lesson 2 – Component Development

Overview

Components are an integral part to the AEM authoring experience. They are the building blocks that developers provide to authors for assembling mobile app pages.

Objectives

- Create a new component using device APIs



Exercise 1 – Create Camera Component

1. Copy **lessons/module3/phonegap-camera** to **/content/src/main/content/jcr_root/apps/summit-developer-lab/components**
2. Open SublimeText editor
3. Select Project > Open Project... menu
4. Go to **~/L712/summit-developer-lab** and select **summit-developer-lab.sublime-project**
5. Open **.content.xml** of the phonegap-camera component and change the title and description of your component
 - a. **jcr:title="Camera"**
 - b. **jcr:Description="PhoneGap Camera Component"**

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <jcr:root xmlns:cq="http://www.day.com/jcr/cq/1.0"
   xmlns:jcr="http://www.jcp.org/jcr/1.0"
3   |   jcr:description="PhoneGap Camera Component"
4   |   jcr:primaryType="cq:Component"
5   |   jcr:title="Camera"
6   |   componentGroup="PhoneGap"/>
7
```

6. Notice the group this component belongs to

Exercise 2 – Add JavaScript to Camera Component

1. Open **clientlibs/phonegap-camera.js** in SublimeText
2. Add JavaScript that will be used by the camera component under the *CameraCtrl* controller
 - a. Copy JavaScript from **'lessons/module3/exercise2/README.md'**
3. Add **'phonegapCamera'** module to **'/content/src/main/content/jcr_root/apps/summit-developer-lab/components/ng-phonegap-page/angular-module-list.js.jsp'**

Exercise 3 – Add HTML to Camera Component

1. Open **phonegap-camera.jsp** in SublimeText
2. Add ng-controller attribute to top level DIV
 - a. **ng-controller="CameraCtrl"**
3. Add a click handler for **takeAPicture()** to the first **<A>** element
 - a. **ng-click=" takeAPicture() "**
4. Add a click handler for **browseForAPicture()** to the second **<A>** element
 - a. **ng-click=" takeAPicture() "**
5. See: **'lessons/module3/exercise3/README.md'** for final HTML

Lesson 3 – Project Installation

Overview

This project uses Maven for building and contains helpful profiles and properties to build and deploy your project to AEM.



Objectives

- Installing an app to AEM

Exercise 1 – Installing a Project to AEM

1. Go to root folder of the lab project
 - a. Open Terminal
 - b. cd ~/lab712/summit-developer-lab
2. Run mvn -PautoInstallPackage clean install to build the content package and install to an AEM instance
3. In a browser navigate to <http://localhost:4502>
4. Log in to the AEM author instance using `admin:admin`
5. Click on **Apps** from the left navigation rail
6. Locate the project you just installed and select it
7. Explore the **Apps Dashboard** for the project you installed

Developer Lab

Modified By Administrator 4 months ago



Outcome

Congratulations! You have just completed an important step on the road to building a hybrid app managed by AEM Apps. You should now be familiar with the following development steps:

- AEM content package structure
- Creating a component that uses PhoneGap APIs
- Using Maven to build and install content packages



Module 4: Managing App Content

Lesson 1 – AEM Authoring Basics

Overview

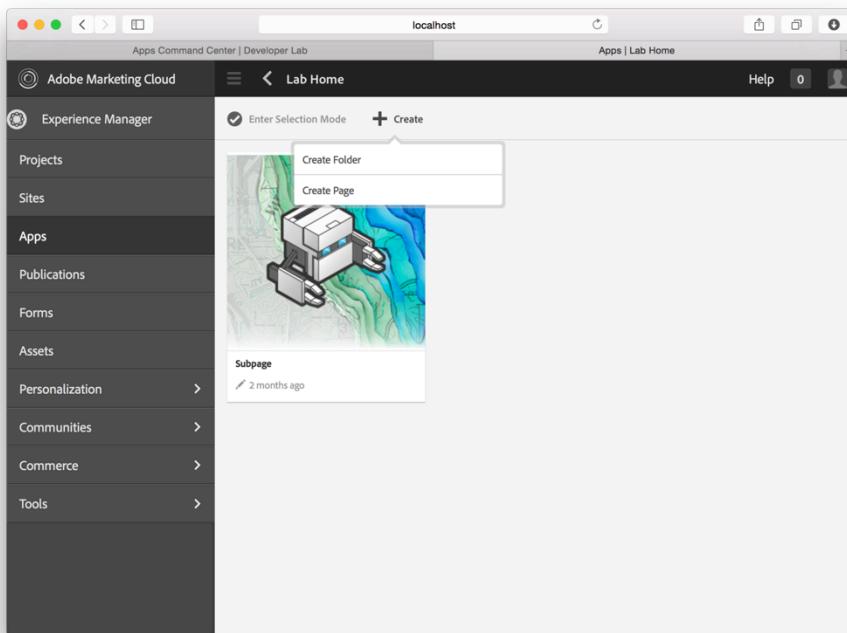
Adobe® Experience Manager helps you organize, create, and manage the delivery of creative assets and other content across your digital marketing channels, including web, mobile, email, communities, and video.

Objectives

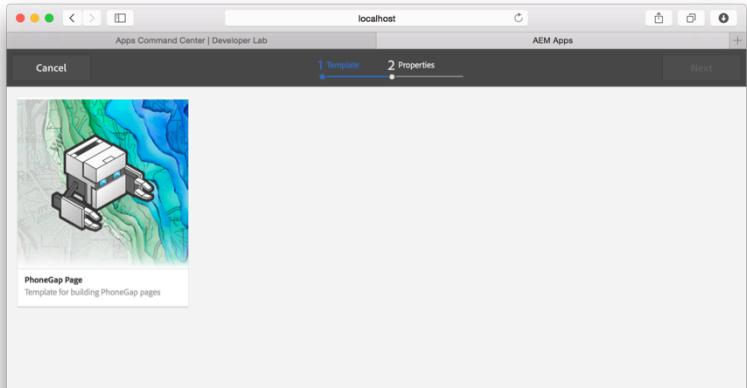
- Learn how to create a page using templates
- Learn how to edit a page using components and assets
- Learn how to preview your page
- Learn the key parts of the easy-to-use AEM authoring interface

Exercise 1 – Creating a page in AEM

1. Go to the Developer Lab App command center on the author instance (<http://localhost:4502/libs/mobileapps/admin/content/dashboard.html/content/phonegap/summit-developer-lab/shell>)
 - a. Click on “Apps”
 - b. Click on “Developer Lab”
 - c. Click on “English” from Content Tile
2. Click on “Lab Home” to drill down to child pages
3. Create a new page by selecting “Create” from the top menu, then “Create Page”:

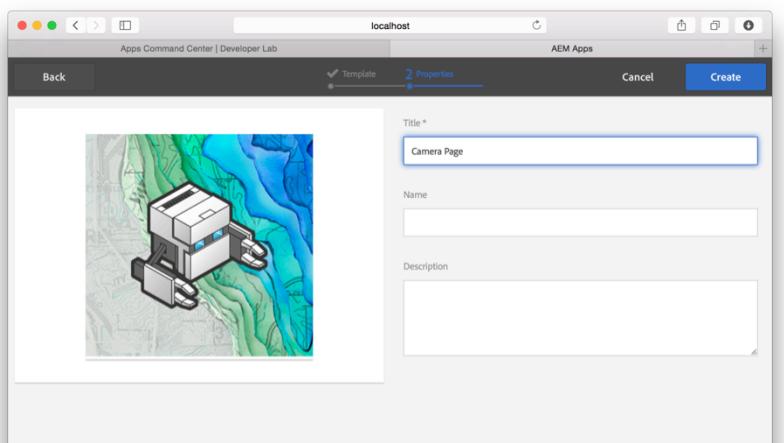


4. Select the “PhoneGap Page” template and click on “Next”



5. Enter a title for your page and click on “Create”

a. eg. **Camera Page**



6. Click Done

Note:

Clicking “Open” will open the newly created page in Edit mode

Clicking “Done” will open the folder/page the page was created in and the card layout will display a new card for the new page.



Exercise 2 - Edit a page in AEM

1. Edit the page from the previous exercise by moving mouse over card to expose the edit icon and click the edit icon

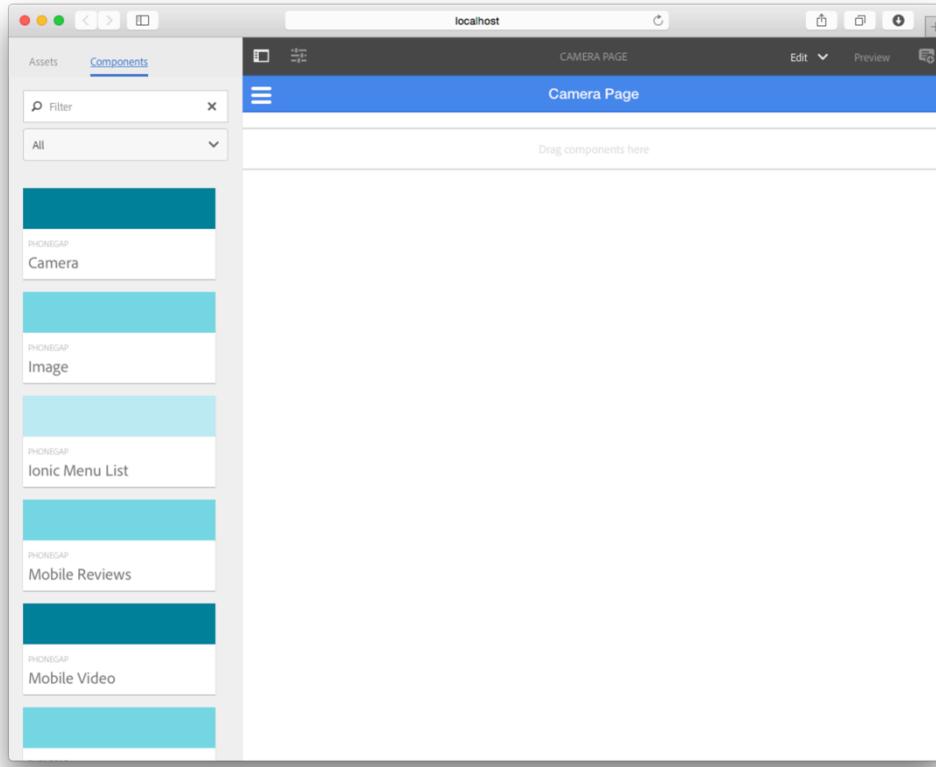
2. Add text component:

- a. Toggle the side panel open by clicking on the following icon.





- b. Select the Components tab



- c. Scroll through the list of components until you find the text component, then drag it onto the page.

Tip: you can filter to the component you want by typing “text” in the filter, for example.

- d. Add some content by clicking slowly twice in the “Text” component box and start typing.

3. Add image component:

- Scroll through the list until you find the image component, then drag it onto the page.
- Select the Assets tab
- Scroll through the list of assets until you find an image you like, then drag it onto the image component added earlier
- Crop the image

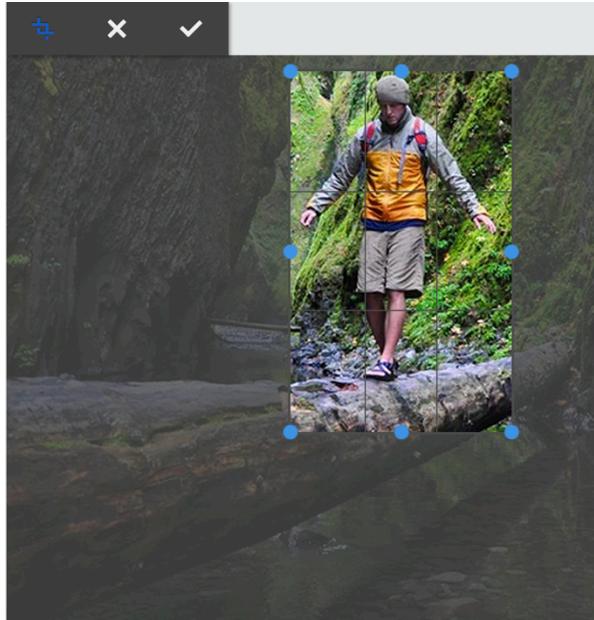
- Select the image



- Click the edit icon Click the crop icon



iii. Crop the image



iv. Click the check box to close the crop

v. Click the check box to close the image edit

4. Add camera component

- Add the camera component you developed in the previous module

The screenshot shows a web application titled "Camera Page" running on "localhost". At the top, there's a toolbar with standard browser controls like back, forward, and refresh, along with "Edit" and "Preview" buttons. Below the toolbar, the main content area has a blue header bar with the title "Camera Page". In the main body, there's a "Sample Text" section containing a small image of a person walking in a forest. Below this is a "Camera" section with the sub-instruction "Take a picture". At the bottom of the page, there are two buttons: "[camera] Take a picture" and "[photo] Browse gallery". A large, empty rectangular area at the very bottom is labeled "Drag components here".



5. Test the page in Preview mode by clicking on the preview button.



6. Close the browser tab to return to the Command Center

Exercise 3- (Optional) Preview in Mobile Emulator

1. Return to editing a page
2. Return to Preview mode.
3. Click the Emulator icon



4. Explore the emulator capabilities to view your content emulating different screen sizes.

Outcome

Congratulations! You are now familiar with the basics of Adobe Experience Manager from a content owner perspective:

- navigating the AEM authoring user interface
- creating page based on templates
- editing pages using components and assets
- previewing pages



Module 5: Reviewing App Content

This module will highlight the options available to review your app content outside of AEM. It will allow you to review your content via the user's app experience rather than the authored browser experience.

Lesson 1 – PhoneGap CLI

Overview

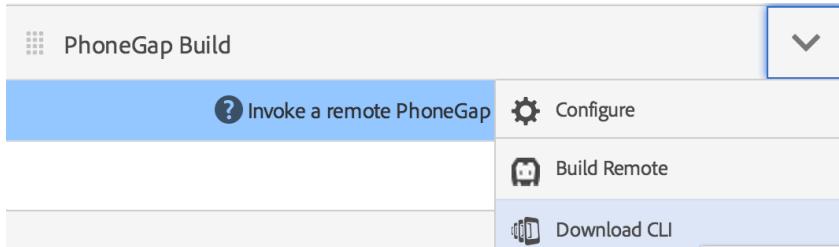
AEM Apps provides the ability to download a fully rendered app payload that contains the correct directory structure and required content for building a mobile app using the PhoneGap command line interface introduced in Module 2.

Objectives

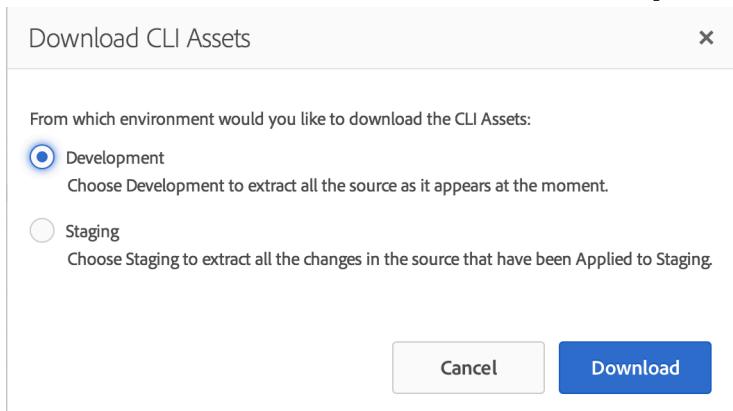
- Download an app as a PhoneGap app payload
- Build your app via PhoneGap CLI
- Run in Simulator

Exercise 1 – Download CLI Assets

1. Navigate back to the **Apps Dashboard** for your app
 - a. <http://localhost:4502/libs/mobileapps/admin/content/dashboard.html/content/phonegap/summert-developer-lab/shell>
2. Click **Download CLI** from PhoneGap Build tiles action menu



3. Click Download and choose to download the Development application assets



4. A zip file of the app should have been downloaded.
5. Open Finder to the Downloads folder to confirm the file was downloaded

Exercise 2 – Building via PhoneGap CLI

1. Double click the downloaded zip to extract it to file system



2. Open Terminal
3. In the Terminal, run the app after changing to the app directory:
cd ~/Downloads/developer-lab*
phonegap run ios

Lesson 2 – PhoneGap Enterprise Viewer

Overview

The PhoneGap Enterprise viewer provides the ability for both developers and marketers to easily review their mobile enterprise apps. The tight integration with Adobe Experience Manager also allows for all of an app's metadata to be viewed. The result is a fast and easy way for PhoneGap Enterprise apps to be tested and used by others at any given stage of development across multiple devices.

Objectives

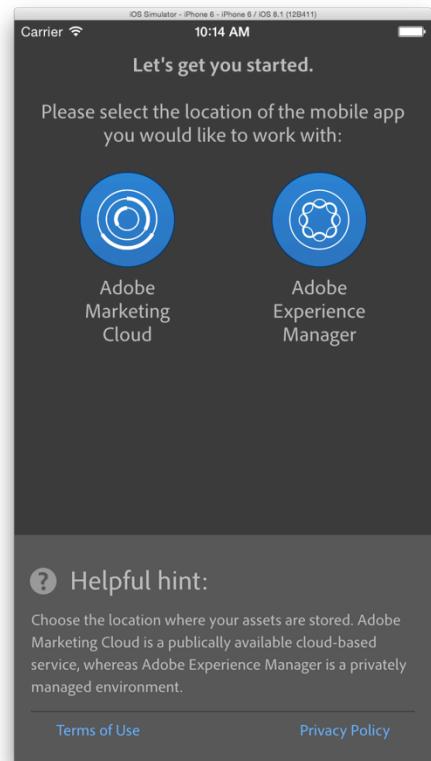
- Download and install the PhoneGap Enterprise viewer app
- View your list of apps
- Deploy a variety of apps to viewer

The PhoneGap Enterprise app is available in the Apple App Store and Google Play Store.

Exercise 1 – Launch Viewer

1. Double click **launch-sim.command** in the root of the lab directory
2. Launch **PhoneGap Enterprise** app

3. Select **Adobe Experience Manager** option
4. Connect to your local AEM instance
 - a. Server address: localhost:4502
 - b. Username: admin
 - c. Password: admin
5. View list of apps
6. Select **Summit Lab** from Development list
7. View meta data for app
8. Select **Open** to download and deploy the app
9. Click and hold for 5 seconds to return back to the PhoneGap Enterprise app list

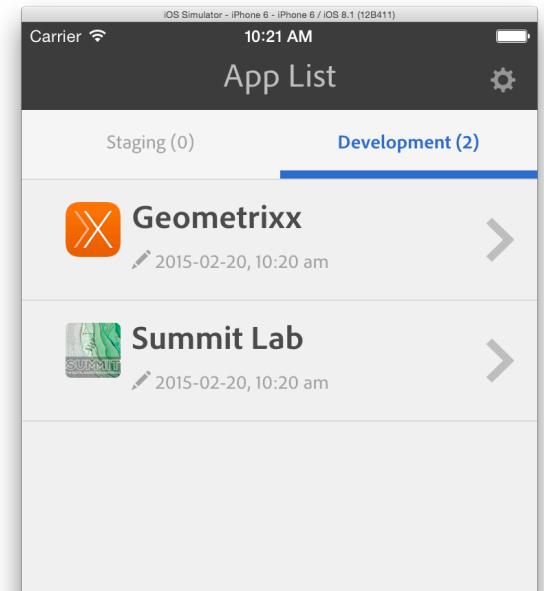


Exercise 2 – Refreshing an App

1. Return to the **Apps Dashboard**



2. Change the App Name and Description by editing the App Details
3. Edit a page in the app
 - a. eg. add or remove some text
4. Go back to the PhoneGap Enterprise app
5. Pull down on the development tab to refresh the app list
6. Open the specified app again
7. Notice the meta data has been updated
8. Tap on **Open** again to download and deploy the app
9. Notice the page content in the app has also changed



Exercise 3 (Optional) – Staging an App

In this exercise the instructor will stage the updated content for publishing in AEM.

1. Return to the **Apps Dashboard** once again
2. From the **Content Release** tile click on **English**
3. From the action menu select Stage Update

Content Update

Content Sync Config: developer-lab Content /content/phonegap/summit-developer-lab/en/jcr:content/pge-app/app-config

Status: Pending

Last Published: Never

No updates found. (Hint: Perhaps there are changes that have yet to be included in an update.)

Stage Update

4. Enter a title for the update and an optional description and click done

Create Update

Update Title *

Update Description

Cancel Done

5. Click Close on update creation confirmation
6. Go back to the PhoneGap Enterprise app
7. Return to the app list using the click and hold gesture
8. Select the **Staging** tab
 - a. Notice the app also appears on this tab since content has now been staged



Outcome

Congratulations! You are now familiar with reviewing PhoneGap Enterprise apps using the PhoneGap CLI and PhoneGap Enterprise Viewer.



Module 6: Adobe Mobile Services Analytics

Adobe Mobile Services provides easy to use, integrated app analytics and targeting capabilities to help mobile teams understand and improve user engagement. Higher engagement leads to improved consumption and monetization of content through the mobile channel. App-specific reports and visualizations help you understand how frequently people use your app, what paths they take through the app, and if they convert effectively to drive monetization. Furthermore, the integrated workflow between analytics and optimization capabilities makes it easier than ever to take action.

Lesson 1 – Add Analytics Tracking

Overview

Instrumenting an app to send analytic data involves adding the Adobe Mobile Services plugin to your app and then instrumenting the app to use the plugin API.

Objectives

- Add Adobe Mobile Services to your PhoneGap application via their PhoneGap plugin
- Add some analytic tracking to a PhoneGap component

Exercise 1 – Add Tracking capability to your app

The developer lab has already completed the following steps required to add tracking capabilities to an app.

1. Add the ADB mobile plugin

- a. See: content/src/main/content/jcr_root/content/phonegap/summit-developer-lab/shell/jcr_content/pge-app/app-content/phonegap/www/config.xml

```
<feature name="ADBMobile">
    <param name="id" value="https://github.com/Adobe-Marketing-Cloud/mobile-services"/>
</feature>
```

2. Copy AMS config to native project

- a. A cordova **after_prepare** hook can be used to copy the AMS config to the the native project
- b. See: content/src/main/content/jcr_root/content/phonegap/summit-developer-lab/shell/jcr_content/pge-app/app-content/phonegap/scripts/copy_AMS_config.js

3. Add the ADB mobile content sync config

- a. Used to include the uploaded mobile config with the app assets package
- b. See: content/src/main/content/jcr_root/content/phonegap/summit-developer-lab/shell/jcr_content/pge-app/app-config-dev/ams-ADBMobileConfig

Exercise 2 – Add Tracking to a Component

1. Open phonegap-camera.js created during Module 3
2. Add instrumentation to track camera actions



```
if (window.ADB) {  
    ADB.trackAction('browseForAPicture', {});  
}
```

See: `summit-developer-lab/lessons/module6/phonegap-camera.js`

3. Install all project changes to AEM

- mvn -PautoInstallPackage clean install

Outcome

Congratulations! You are now familiar with Adobe Mobile Services and know how to add tracking capabilities to an app.

Lesson 2 – Add Analytics Reporting

Overview

Add AMS analytics reporting to your app.

Objectives

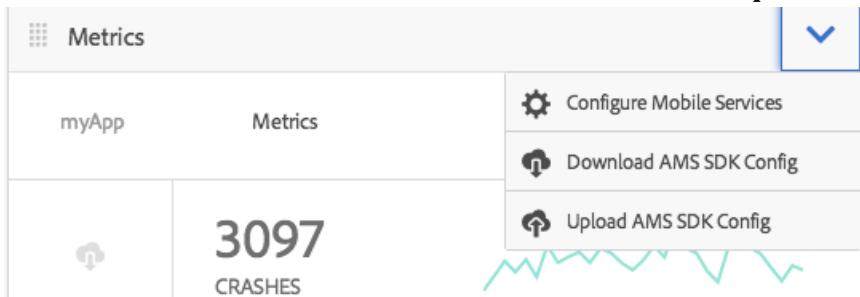
- Assign an Adobe Analytics configuration to your app

Exercise 1 – Add an Adobe Mobile Services configuration

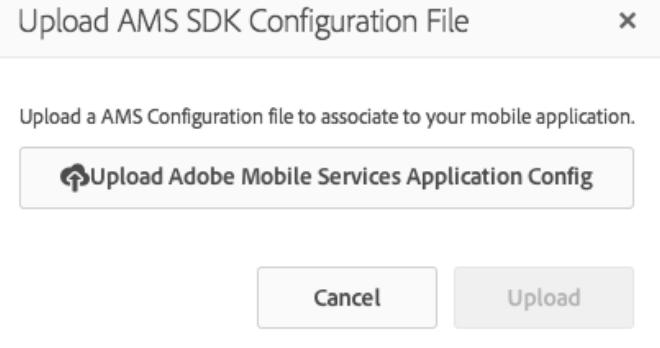
An AMS Configuration, Summit2015AppsLabConfig, is pre-configured in AEM for your use and has already been associated with the developer lab app.

Exercise 2 – Add an Adobe Mobile Services configuration to your app

- Navigate to the **Apps Dashboard** for your app
- Confirm the Metrics tile starts to load the latest metrics
- Click the down arrow icon from the Metrics tile and select **Upload AMS SDK Config**



- Click Upload Adobe Mobile Services Application Config



5. Browse to the provided ADBMobileConfig.json at `summit-developer-lab/lessons/module6` and click Open
6. This ADBMobileConfig.json will now be associated to your AMS Configuration

Outcome

Congratulations, you have now added AMS analytics reporting to your app.

Lesson 3 – Test Analytics Reporting

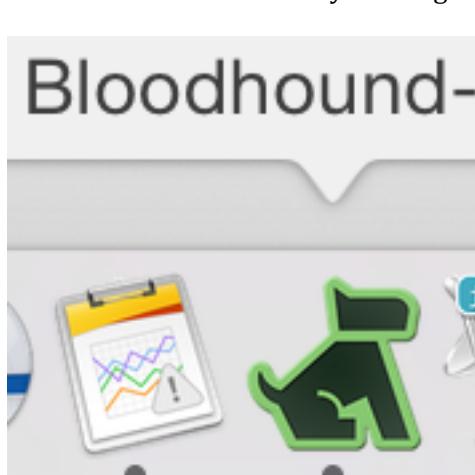
Overview

Test that an app has been instrumented correctly for tracking analytics.

Objectives

- Discover Bloodhound as an app analytic testing tool
- Test your app's lifecycle tracking
- Test your app's action tracking

Exercise 1 – Launch Bloodhound



1. Launch Bloodhound by clicking Bloodhound icon from the dock at the bottom of your screen

2. Confirm app opened and not tracking hits have been logged



A screenshot of the Adobe Bloodhound application. The window title is "Adobe Bloodhound" and the URL is "http://jfaits-mbp:50000/pac". The interface has a dark sidebar on the left with "View", "Test", and "Setup" options. The main area shows a table titled "Hits" with columns "Hit" and "Time", which is currently empty. To the right of the table are two expandable sections: "Hit Details" and "Additional Data", both currently collapsed. Below these is a section titled "Context Data" with a table showing "Key" and "Value" columns, also currently empty.

Exercise 2 – Test the application

1. Download a development payload package from the **AEM Apps Dashboard**
2. Unzip the downloaded package
3. Run the app in the iOS simulator (`phonegap run ios`)
4. Note lifecycle events in Bloodhound
5. Navigate to a variety of pages in the app
6. Return to Bloodhound and notice actions and events

Outcome

You are now able to test an instrumented app locally to ensure actions are being tracked correctly.



Appendix

PhoneGap

Appendix A – Installing Node

The PhoneGap CLI is built upon Node. Node is a platform built upon the Chrome browser's JavaScript engine, and it is used for building fast and scalable network applications.

You can test to see if Node is available by opening a Terminal and typing: `node -v`

1. Go to <http://nodejs.org/> and click “INSTALL”
2. When the package has finished downloading, click on the Downloads icon to the left of the trashcan, then click on the cardboard box icon.
3. In the installer, click “Continue”, “Continue”, “Agree”.
4. Click “Install for all users of this computer” and then click “Continue”
5. Click “Install”
6. Enter the administrator password provided
7. Click “Close”

Appendix B – Installing PhoneGap CLI

You can test to see if PhoneGap is available by opening a and typing: `phonegap -v`

1. Open Terminal by pressing CMD-spacebar, typing “Terminal”, then hitting return.
2. Install PhoneGap by typing:
`sudo npm install -g phonegap`
... and hit return.
3. Confirm the install by typing:
`phonegap -version`
... and hit return. You should see output like “4.2.0-0.23.0”.

Appendix C – PhoneGap Build

PhoneGap Build takes the pain out of compiling PhoneGap apps. Get app-store ready apps without the headache of maintaining native SDKs. The PhoneGap Build service does the work for you by compiling in the cloud. You can use PhoneGap Build to compile for iOS, Android, and Windows Phone.

Building remotely using the cloud-based PhoneGap Build service required having a PhoneGap build account.

Logging in to PhoneGap Build

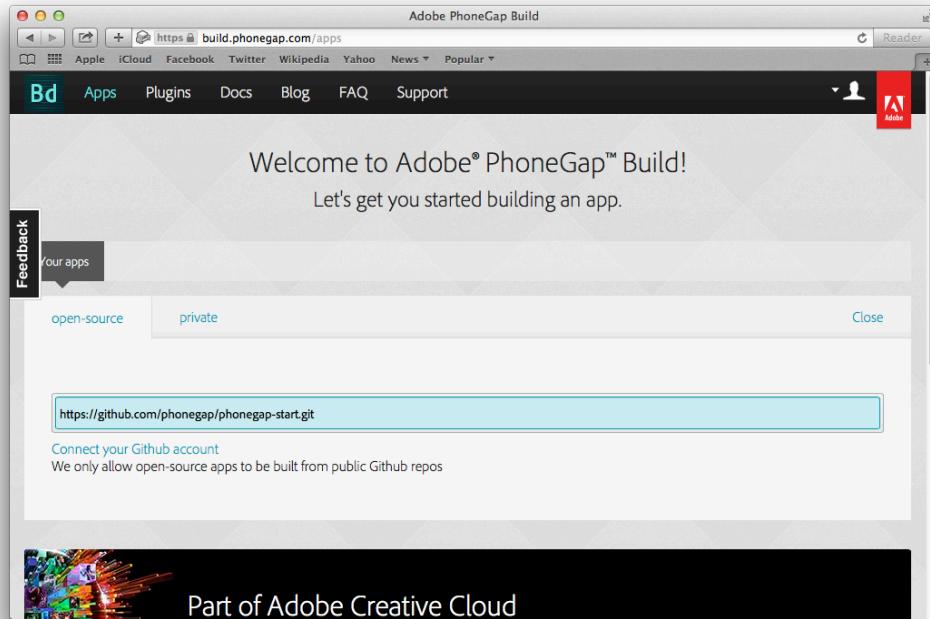
1. Visit the PhoneGap Build website at <http://build.phonegap.com/>
2. Log in with your PhoneGap Build credentials (Adobe ID).

Building a mobile application

1. Navigate to the apps list in PhoneGap Build at <https://build.phonegap.com/apps>
2. Click on the “+ new app” button



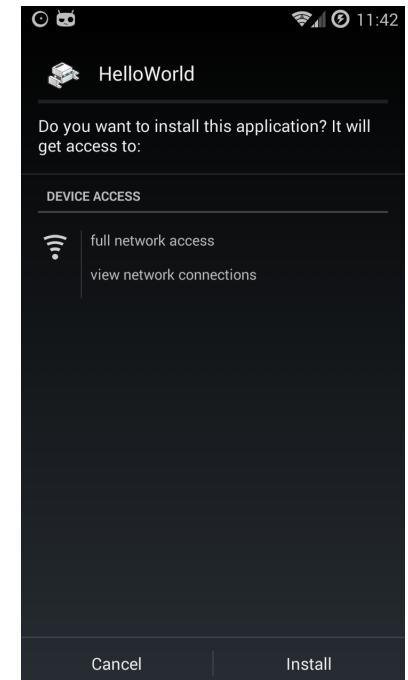
3. Make sure that the “open-source” tab is selected
4. In the “paste .git repo” text box, enter the following: <https://github.com/phonegap/phonegap-start.git>



5. Once PhoneGap Build has finished “Fetching repo”, click on “Apps” in the top-left menu
6. Click on “Ready to build”
7. On the next screen, wait until the Android and Windows Phone logos turn blue.
8. Click on the Hello World title and explore the different options and settings.

Building a mobile application remotely for Android

1. Build and run an existing PhoneGap app by typing the following into the Terminal:
`phonegap remote run android`
2. Enter your PhoneGap Build account credentials when prompted.
Note: this will bind your account to your application. Subsequent invocations will reuse your stored account credentials.
3. Log into PhoneGap Build at <https://build.phonegap.com/> to view your application.



Installing a mobile application on Android

1. If you have an Android mobile phone, install a QR Code reader from the Play Store at <https://play.google.com/store/search?q=qr>
2. Allow your phone to load unauthorized apps:
 - a. Open Settings
 - b. Under Security, enable “Unknown sources”
 - c. Remember to disable this again when you are done
3. Scan the QR code on the PhoneGap Build page using the QR scanning app



- a. Open the QR code link in the phone browser
 - b. When prompted "This type of file can harm your mobile device", click OK to install.
 - c. Open the notification area by swiping down from the top of the screen, and click on the notification about "Hello World" being downloaded.
 - d. When prompted to install, click Next, then Install.
 - e. If prompted about Google verifying apps, click Accept.
4. When the app is installed, click "Open" or open it from the apps list.

Remember to disable unauthorized apps when you are done.