Campaign Manager Sample Setup Instructions

# Required Software

## Tools

* Android Development Tools (Sample was built with Eclipse)
* Java SDK (<http://www.oracle.com/technetwork/java/javase/downloads>)
* Android SDK – All Packages
* Git (e.g. GitHub for Windows <http://github-windows.s3.amazonaws.com/GitHubSetup.exe> )
* eGit (Eclipse Git Plugin) - Git Repository Plugin (<http://www.eclipse.org/egit/download/> )
* Maven SDK Deployer <http://maven.apache.org/download.cgi>
* Maven
* For debugging on a device, configure USB Debugging
  + instructions are here <http://developer.android.com/tools/device.html>)
* Create an emulator based on a tablet e.g. the Nexus 10 (Note: on Windows, limit emulator device memory to 768MB or the VM will not run reliably).

## Android SDK

### Android SDK for Azure Active Directory

* Follow the set up instructions for prerequisites at <https://github.com/MSOpenTech/azure-activedirectory-library-for-android>
  + Get the Maven SDK Deployer from <https://github.com/mosabua/maven-android-sdk-deployer>
* From the Git Repository Exploring view clone <https://github.com/MSOpenTech/azure-activedirectory-library-for-android.git>
* Import com.microsoft.adal to workspace

### Android SDK for Office 365

* From the Git Repository Exploring view clone <https://github.com/OfficeDev/Office-365-SDK-for-Android.git>
* Import office365-lists-sdk
* Import office365-base-sdk (may already have been pulled in by lists sdk)

## This Sample

### Running the app from .apk

1. Find the latest .apk file in the bin folder of the sample
2. Transfer the file to your device e.g. by sending it as an attachment to the mail account configured for your device
3. Open the attachment and confirm that you’d like to install it on your device

### Running the app from source

* Import Existing Android Code into workspace from your download/extract directory
* Add references to the ADAL, office365-base-sdk and office365-lists-sdk if they are not there already

1. Right-click on the campaign manager
2. Select Properties
3. Click on Android
4. Add Library reference
5. Select all three libraries
6. Click ok

# Prerequisites

## Setting up Eclipse Android Tools

### Installing the Java SDK

Download the latest JDK from here: <http://www.oracle.com/technetwork/java/javase/downloads>

We recommend the 64 Bit version because some O365 SDK prerequisites require 64 bit Java

e.g. jdk-8u5-windows-x64.exe

### Install the SDK and Eclipse IDE

1. Download the ADK from here <https://developer.android.com/sdk/index.html>
2. Unpack the ZIP file (named adt-bundle-<os\_platform>.zip) and save it to an appropriate location, such as a "Development" directory in your home directory.
3. Open the adt-bundle-<os\_platform>/eclipse/ directory and launch eclipse.

Caution: Do not move any of the files or directories from the adt-bundle-<os\_platform> directory. If you move the eclipse or sdk directory, ADT will not be able to locate the SDK and you'll need to manually update the ADT preferences.

The most current instructions are here <https://developer.android.com/sdk/installing/bundle.html>

### Install Git for Eclipse Plugin

1. Add eGit Plugin
2. Help -> Install New Software
3. Add Site <http://download.eclipse.org/egit/updates/>
4. Select Eclipse Git Team Provider

### Install all Android SDK Packages

To download the complete Android SDK:

1. launch Android SDK Manager from Eclipse
2. Select all packages
3. Click Install xx packages
4. Accept all the license agreements
5. Click Install (this will take a while)
6. After all SDK components have been downloaded you may need to update the Android plugins (Eclipse will show a dialog if you need to update. However the check for updates button may yield nothing)
7. To get the latest plugin go to
8. Help -> Install New Software
9. Select the Android Developer Tools Update Site - https://dl-ssl.google.com/android/eclipse/
10. Filter by Android
11. Select All
12. Next, Next, Accept Licenses, Finish

## Set up Azure Active Directory SDK for Android

### Prerequisites

#### Git

Install GitHub for Windows from here <http://github-windows.s3.amazonaws.com/GitHubSetup.exe> or a Git shell of your choice

#### Install the Maven SDK Deployer

1. Download the Maven binaries here <http://maven.apache.org/download.cgi> (e.g. apache-maven-3.2.1-bin.zip)
2. Unzip the distribution archive, i.e. apache-maven-3.2.1-bin.zip to the directory you wish to install Maven 3.2.1. These instructions assume you chose C:\Program Files\Apache Software Foundation. The subdirectory apache-maven-3.2.1 will be created from the archive.
3. Add the M2\_HOME environment variable by opening up the system properties (WinKey + Pause), selecting the "Advanced" tab, and the "Environment Variables" button, then adding the M2\_HOME variable in the user variables with the value C:\Program Files\Apache Software Foundation\apache-maven-3.2.1. Be sure to omit any quotation marks around the path even if it contains spaces. Note: For Maven 2.0.9, also be sure that the M2\_HOME doesn't have a '\' as last character.
4. In the same dialog, add the M2 environment variable in the user variables with the value %M2\_HOME%\bin.
5. Optional: In the same dialog, add the MAVEN\_OPTS environment variable in the user variables to specify JVM properties, e.g. the value -Xms256m -Xmx512m. This environment variable can be used to supply extra options to Maven.
6. In the same dialog, update/create the Path environment variable in the user variables and prepend the value %M2% to add Maven available in the command line.
7. In the same dialog, make sure that JAVA\_HOME exists in your user variables or in the system variables and it is set to the location of your JDK, e.g. C:\Program Files\Java\jdk1.7.0\_51 and that %JAVA\_HOME%\bin is in your Path environment variable.
8. In the same dialog, add the ANDROID\_HOME environment variable in the user variables with the path to the Android SDK.
9. Open a new command prompt (Winkey + R then type cmd) and run mvn --version to verify that it is correctly installed.

Run Maven to set up project dependencies for Azure AD and Office 365 SDKs for Android

1. Open Git shell
2. navigate to the folder where you want to install maven e.g. a peer folder to the ADT directory
3. git clone <https://github.com/mosabua/maven-android-sdk-deployer.git>
4. cd maven-android-sdk-deployer\platforms\android-19
5. mvn clean install
6. cd ..\..\extras\compatibility-v4
7. mvn clean install

### Setup Azure AD SDK (ADAL)

1. Download ADAL Android Preview v0.6 from <https://github.com/MSOpenTech/azure-activedirectory-library-for-android/archive/v0.6-alpha.zip>
2. Extract the archive
3. Go back to the git shell
4. Navigate to the folder of the ADAL repo (e.g. C:\android\azure-activedirectory-library-for-android
5. run mvn clean install

### Import ADAL

1. Go to Eclipse
2. In the Package Explorer view right-click and select Import…
3. Select Android -> Existing Android Code Into Workspace
4. Click Next

As Root Directory select the folder where you extracted the ADAL SDK. This should bring up a list of projects that can be imported.

1. Deselect All and select only adal
2. Select Copy projects into workspace
3. Click Finish
4. Switch to Java view

### Import Office 365 SDK for Android

1. In Eclipse go to Window -> Open Perspective -> Other
2. Select Git (or Exploring Git Repositories)
3. This will open a tab labeled Git Repositories
4. Click on the button to Clone and import a repository
5. Select Clone URI
6. Enter <https://github.com/OfficeDev/Office-365-SDK-for-Android.git>
7. Click Next
8. Select master
9. Right-click anywhere within the Office 365 Repo
10. Select Import Projects
11. Select office365-base-sdk
12. Select Import Projects
13. Select office365-lists-sdk
14. Switch to the Java view

There will be an error because of a missing library called guava-16.0.1. To get that

1. Open Windows Explorer
2. Navigate to the libs folder of the office365-base-sdk (e.g. c:\users\username\git\Office-365-SDK-for-Android\sdk\office365-base-sdk\libs)
3. Right-click on getLibs.ps1
4. Select Run with Powershell
5. This will download the correct version of guava to the folder
6. Switch to Eclipse
7. In package explorer right-click and select Refresh
8. Now all errors should be resolved.

## Adding the Campaign Manager App

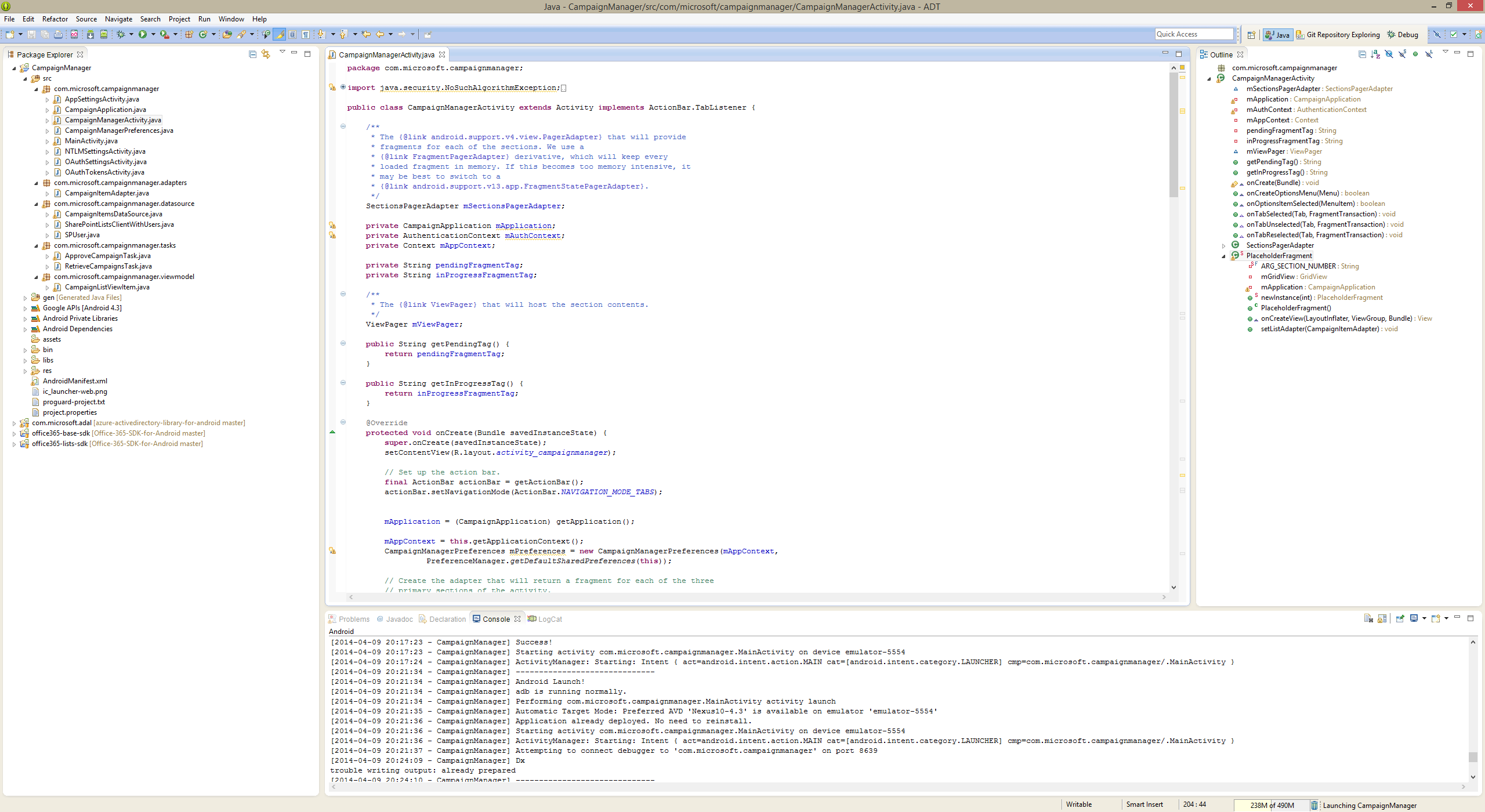
### Once the sample is live on GitHub

1. In Eclipse go to Window -> Open Perspective -> Other
2. Select Git (or Exploring Git Repositories)
3. This will open a tab labeled Git Repositories
4. Click on the button to Clone and import a repository
5. Select Clone URI
6. Enter <https://github.com/OfficeDev/Campaign-Manager-Code-Sample.git>
7. Next
8. Select master (it's ok to have any other branches, too)
9. Right-click anywhere within the Office 365 Repo
10. Select Import Projects
11. Select office365-base-sdk
12. Select Import Projects
13. Select office365-lists-sdk
14. Switch to the Java view

### Adding references

1. Right-click on Campaign Manager project
2. Select Properties
3. Select Android
4. Click Add
5. Select com.microsoft.adal
6. Click Ok
7. Repeat for o365-base-sdk and o365-lists-sdk libraries

Once Eclipse is set up correctly it should look something like this:



## Setting up Office 365 and Microsoft Azure AD

### Required Accounts

This sample requires a SharePoint site on an Office 365 subscription with an associated Microsoft Azure subscription.

### Creating an Azure subscription for an Office 365 Tenant

1. Sign up for Office 365 (if you don’t have an account already)
   * 1. Sign up for a [Developer Account](https://portal.microsoftonline.com/Signup/MainSignup15.aspx?OfferId=6881A1CB-F4EB-4db3-9F18-388898DAF510&DL=DEVELOPERPACK&ali=1)
     2. Sign up for an [Enterprise Trial Account](http://office.microsoft.com/en-us/business/office-365-enterprise-e3-business-software-FX103030346.aspx)
2. Log in to Office 365
3. In the same browser session browse to <https://manage.windowsazure.com>
4. If you don’t have an Azure account associated with your Office 365 Login, you will see what looks like an error page. In that case select “Sign up for Microsoft Azure”
5. If you already have an Azure account associated with your login, you will see the Azure portal

### Setting up the Site in SharePoint

1. On a site collection dedicated for testing, select or create a site for working with this sample app. The site can be anywhere on your SharePoint tenant including public or my site. We recommend using a site on a private site collection.
2. Note the site url. It will be needed to configure the app on first launch.

During first run, the app will create a list in this site with some sample data.

### Setting up the Application in Azure Active Directory

This is now optional. The app has a button to create the Application in your Office 365 Azure AD environment automatically using the Graph API. It requires tenant admin access. It is not recommended to use this option in a production environment.

1. Go to the Directory node within the Azure subscription
2. Add a new application
   1. Select “Add an application my organization is developing”
3. Give it a name e.g. Campaign Manager Mobile
4. Select native client application
5. Select Configure
6. Note the client id (you’ll need it later to configure the Android app)
7. Enter a redirect uri (it doesn’t have to be real, just a valid uri, e.g. http://home)
8. Add permissions for Office 365
   1. Select Application Office 365 SharePoint Online
   2. Select Create or delete items and lists in all site collections
   3. Select Edit or delete items in all site collections

# Configuring the App

1. Launch the app
2. On first launch the app will open a simplified OAuth configuration page
3. Enter the Office 365 Login you are going to use
4. Enter the site url of the SharePoint site you want to test the app with
5. Click the button to create the application in Azure AD to create OAuth client Id and redirect url (requires tenant admin permissions) or Enter the client ID and redirect URL you configured in Azure AD in 2.2.4
6. Click Authenticate &Create List to test the configuration

You will be prompted for your Office 365 credentials. With the resulting auth code, the app will request an access code for the SharePoint site and create a list called Campaigns and populate it with some sample data. Once complete the Launch button will be activated.

Click Launch to access the Campaign manager UI. On subsequent runs, the app will go there directly.

You can access the complete set of app settings from the gear menu in the top right corner.

# Running the App

Once configured correctly the app will open up in the Pending Approvals view.

Note: For now you have to close and restart the app after changing settings to get it to load with the updated settings. If you just see a white screen, close and restart the app.

Swipe horizontally to switch between Pending Approvals and Campaigns in Progress

If the user using the app is the approver, the approve/reject buttons will show in the Pending Approvals view

If not, the approval status of the campaign will show

Once approved, the app will refresh the Campaigns in Progress view and when swiping to it the newly approved campaign will show up.

Until the next refresh, the campaign will also show up in Pending Approvals but it’ll show the approved status. That’s by design to ensure the user sees the result of their action rather than having the campaign disappear on them.

The refresh button will refresh both views.

# Resources

All SharePoint Conference 2014 Talks

<http://channel9.msdn.com/Events/SharePoint-Conference/2014>

SharePoint Conference Talk by Josh Gavant

<http://channel9.msdn.com/Events/SharePoint-Conference/2014/SPC373>

Office 365 SDK for Android GitHub Page

<https://github.com/OfficeDev/Office-365-SDK-for-Android>

Windows Azure Active Directory Authentication Library (ADAL) for Android Github Page

<https://github.com/MSOpenTech/azure-activedirectory-library-for-android>