

Installation Guide for Developers

/******

CastOffline is an Android App to stream local phone media (photos, videos and audios) to a Chromecast.

Copyright 2014 Divya Anna Marcus

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Link for repository: <https://github.com/danna123/CastOffline>

License terms are specified in the License file. Link :
<https://github.com/danna123/CastOffline/blob/master/LICENSE>

*****/

Instructions to build the application:

CastOffline project build has two parts: Android Sender and a Custom Receiver

Android Sender Application:

1. Installation of JDK for eclipse ADT

- JDK 6 or JDK 7 (JRE is not sufficient)

<http://www.oracle.com/technetwork/java/javase/downloads/index.htm>

- Download and follow Eclipse ADT bundle instructions

<https://developer.android.com/sdk/index.htm>

<https://developer.android.com/sdk/installing/index.html?pkg=adt>

2. Android sender downloads for CastOffline:

- Install the Android Support Libraries through the Android SDK Manager. The support libraries need to be revision **19.0.1 or later**.

Link: <http://developer.android.com/tools/support-library/setup.html>

- Install the Google Play services SDK through the Android SDK Manager. The Google Play services SDK needs to be revision **4.2 or later**.

Link: <http://developer.android.com/google/play-services/setup.html>

More details: https://developers.google.com/cast/docs/android_sender

3. Library Dependencies:

- android-support-v7-appcompat which can be found at

<SDK install location> /extras/android/support/support/v7/appcompat

- android-support-v7-mediarouter which can be found at

<SDK install location>/extras/android/support/v7/mediarouter (this has a dependency on android-support-v7-appcompat)

- google-play-services_lib which can be found at

<SDK install location>/extras/google/google_play_services/libproject/google-play-services_lib

Import the above as library projects for your IDE.

More details: https://developers.google.com/cast/docs/android_sender

4. Google Chromecast setup:

- Installing Chromecast (First Time User)

<https://cast.google.com/chromecast/setup/>

- Mobile setup for Chromecast:

Install Chromecast app from google play and follow the instructions.

Important: To enable Chromecast in debugging mode, check the box in Chromecast settings page that says “Send this Chromecast’s serial number when checking for updates”.

Link:

<https://play.google.com/store/apps/details?id=com.google.android.apps.chromecast.app&hl=en>

Custom Receiver Application:

Receiver Application can be a default receiver or a custom receiver. Custom Receiver is explained in step 5 and Default Receiver is explained in step 6.

5. Custom Receiver Application Development:

a) Register Chromecast on Google Cast SDK Developer Console.

Details on Registration under the title “devices” in the following link.

Link: <https://developers.google.com/cast/docs/registration>

Once the device is ready for testing, the receiver application can be debugged using

`http://<chromecast_device_ip>:9222`

Link for Registration : <https://cast.google.com/publish/>

b) Register the application URL to get App ID:

In this project, the receiver application uses google drive for hosting the custom receiver. Any web server can be used to host the web page. Since this has been developed as part of a course project and requires url to be accessible publicly(to avoid the pain of generating app id each time),google drive served the purpose.

c) Google drive for web hosting: Google drive can be used to host html,javascript and css web pages.

Steps are as follows:

1. Create a folder in the google drive and share it as public.

2. Place html ,css and javascript pages in this folder and share them as public
3. To Traverse through the web page, make the below changes:

eg: https://drive.google.com/?authuser=0#folders/0B61ekPEN_94sOVpfbFAxN2VJelU is the folder containing files and shared as public

Changes: https://www.google.com/hosting/docs/0B61ekPEN_94sOVpfbFAxN2VJelU

d) **Register the url in developer's console:**

https://www.google.com/hosting/docs/0B61ekPEN_94sOVpfbFAxN2VJelU/XXX.html in developer console to generate the app id.

e) **App id** : Enter the app id obtained in the strings.xml file of the project.

Link: <https://github.com/danna123/CastOffline/blob/master/res/values/strings.xml>

6. Default Receiver:

If you want to use a default receiver then add the following to the strings.xml file.

Link: <https://github.com/danna123/CastOffline/blob/master/res/values/strings.xml>

Add: CastMediaControlIntent.DEFAULT_MEDIA_RECEIVER_APPLICATION_ID as appid to the following tag

```
<string name="app_id"> </string>
```

Project Download:

1. Download the project from <https://github.com/danna123/CastOffline> using clone in desktop in github page.
2. The project can be opened in eclipse using File-> Import ->Existing Android Code Into workspace
3. Check for the library dependencies

Right click on project CastOffline in eclipse->properties->Android->Library-> Add

Add all library dependencies in Step 3 and it shouldn't show any red mark. Click Ok

4. If receiver chosen as default receiver, follow step 6.

5. If receiver chosen as custom receiver follow step 5. The receiver files can be found in

Link: <https://github.com/danna123/CastOffline/tree/master/receiver>

Note: Chromecast and android phone should be in the same network.

7. The project cannot be run on an emulator and hence connect the phone to the laptop running Eclipse ADT.

1. Right click on the project CastOffline that has been imported → Run as android application.

2. Select the target as choose a running android device (the connected phone details should be seen here) and click ok.

8. Receiver Application can be debugged using `http://<chromecast_ip>:9222`