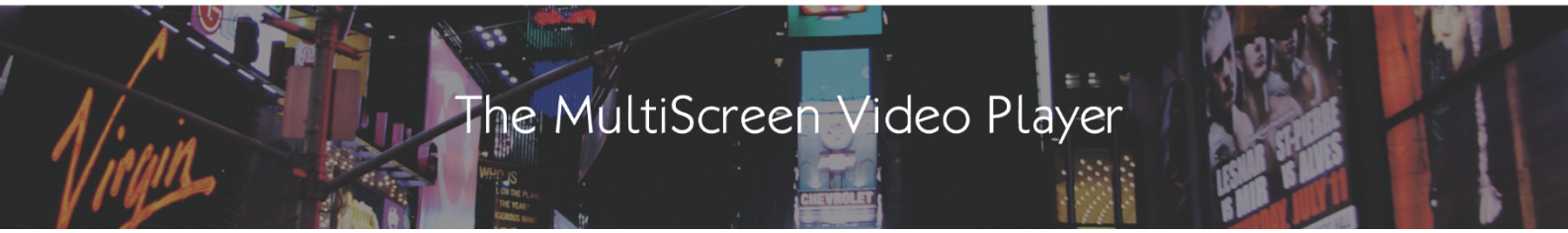


nexxPLAY Integration

11.03.17

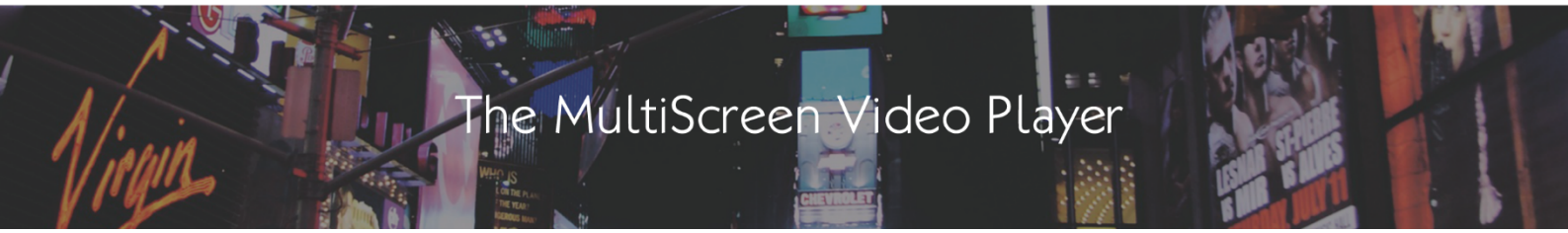
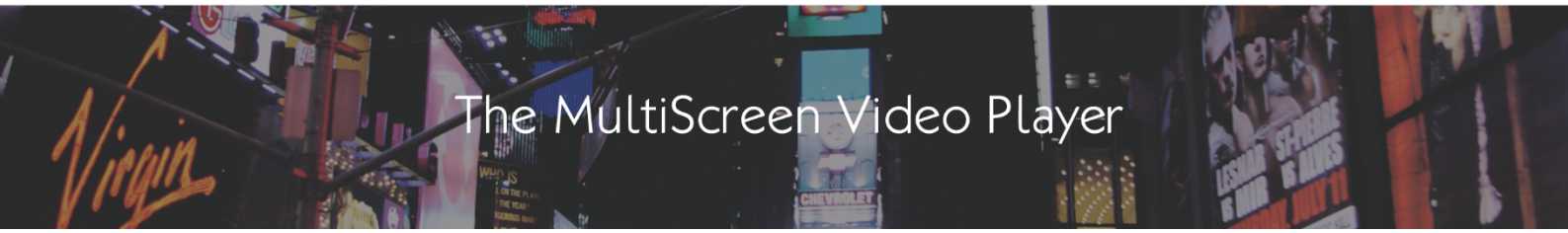


Table of Contents



Integration Guidelines

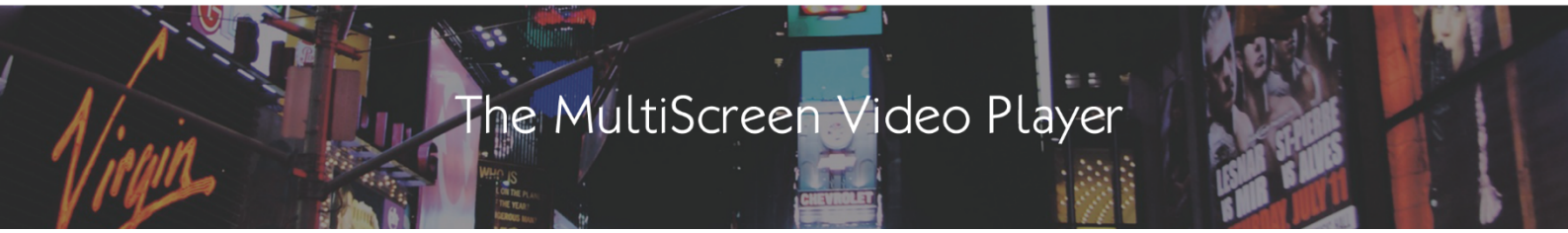
This section shows how to integrate the media player into an Android app. The App *NexxPlayerTestSdk* delivered also as source code with this SDK is an example of integration. The App contains a launcher activity *MainActivity* used to start *PlayerActivity*. The later uses NexxPlayer SDK to create a player, providing the rendering surface and a notification listener.

Copy "sdkrelease.aar" into "libs" directory of the application. Android Studio must be instructed to search "libs" folder for repositories (in *build.gradle*):

```
repositories {
    …
    flatD
    ir {
        dirs 'libs'
    }
    …
}
```

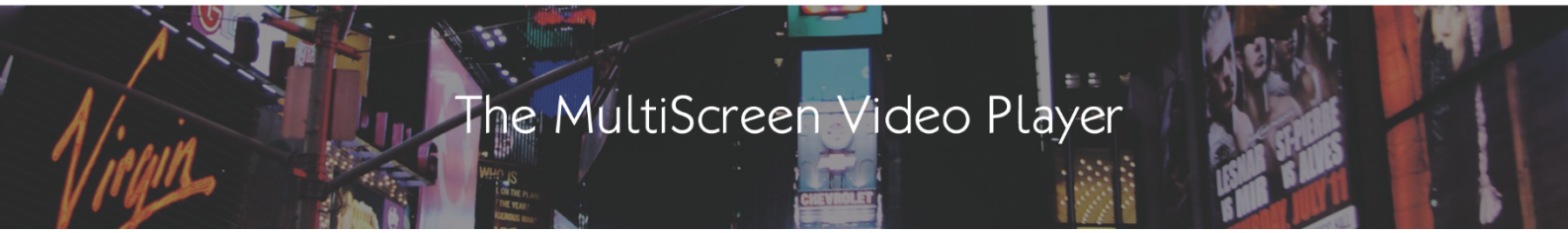
The library must be then compiled into the app (see app/build.gradle)

```
dependencies {
    ...
    compile(name:'sdkrelease', ext:'aar')
    ...
}
```



Since the NexxPlayer SDK library uses thirdparty components which could not be included with *sdkrelease.aar*, they must be also included into the build. Be sure that your *build.gradle* contains also these lines:

```
repositories
    { mavenCentral()

...
    maven { url "https://repo.spring.io/release" }
    maven { url "https://repo.spring.io/
    milestone" } maven { url "https://
... repo.spring.io/snapshot" }
}


dependencies {
    …
    compile 'com.google.code.gson:gson:2.3.1'
    compile 'org.springframework.android:springandroidresttemplate:2.0.0.M1'

    compile('org.simpleframework:simplexml:2.7.1') { exclude group: 'stax', module: 'staxapi'
            exclude group: 'xpp3', module: 'xpp3'
    }

    compile 'com.noveogroup.android:androidlogger:1.3.1' compile ,org.apache.mina:mina-
    statemachine:2.0.9'

    // if using ima:
    compile 'com.google.ads.interactivemedia.v3:interactivemedia:3.2.1'
    compile 'com.google.android.gms:play-services-ads:8.4.0'
}
```



Instantiating and preparing the player

Create the NexxPlayer object providing API by calling the factory method within your Activity:

```
this.nexxPlayerAndroid  =  NexxFactory.createNexxPlayer(this);
```

The NexxPlayer needs several views. All these views must be injected using their setters as follows. The root view is the view group where the media player control layout (the player skin) is shown.

```
ViewGroup root = (ViewGroup) findViewById(R.id.root);
this.nexxPlayerAndroid.setViewRoot(root);
```

The surface view shows the media.

```
this.nexxPlayerAndroid.setSurfaceView(surfaceView);
```

The debug view is optional and shows debugging information.

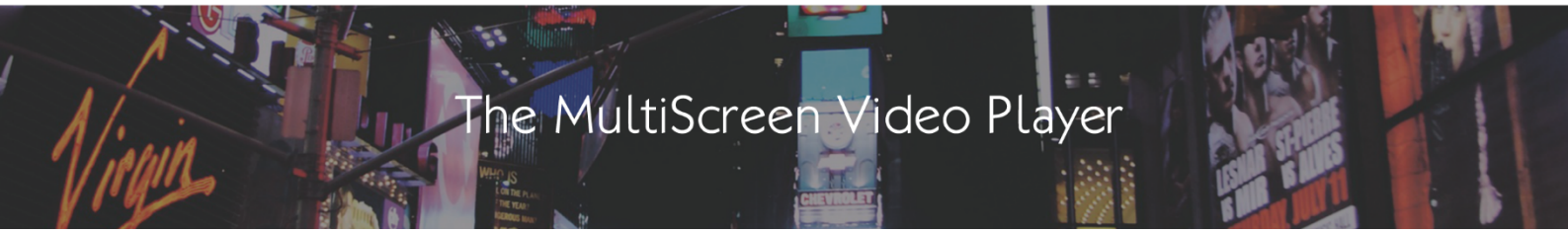
```
this.nexxPlayerAndroid.setDebugTextView(debugTextView);
```

Player state shows the current state of the player (Buffering, Playing, etc.)

```
this.nexxPlayerAndroid.setPlayerStateTextView(playerStateTextView);
```

Finally the subtitle view shows subtitles, for the case that media provides some.

```
this.nexxPlayerAndroid.setSubtitleView(subtitleView);
```

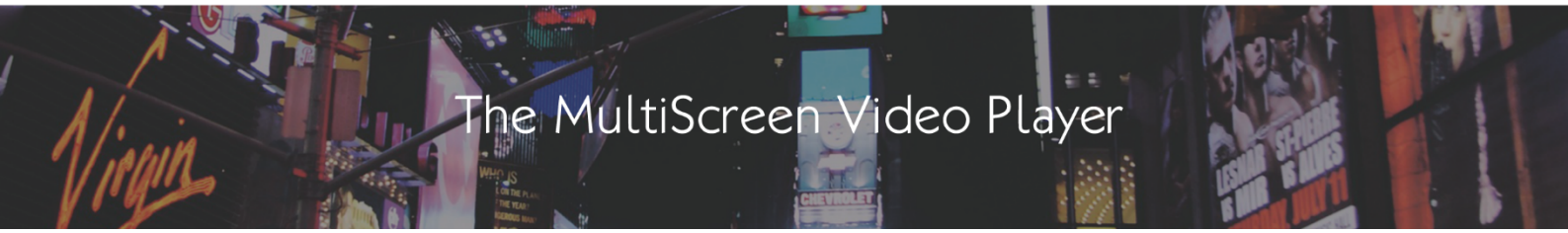


Notifications

Application may subscribe for receiving notifications from the player calling:

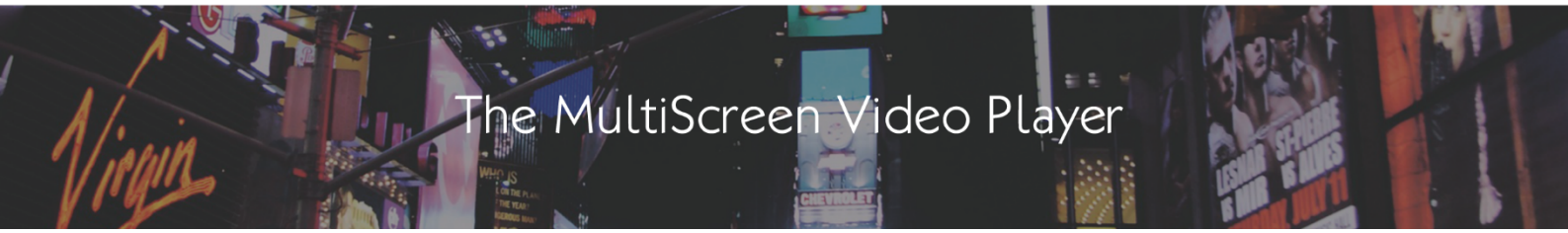
```
NexxPlayerNotification.Listener listener = ...
this.nexxPlayer.addEventListener(listener);
```

where listener is a custom class implementing *NexxPlayerNotification.Listener.*

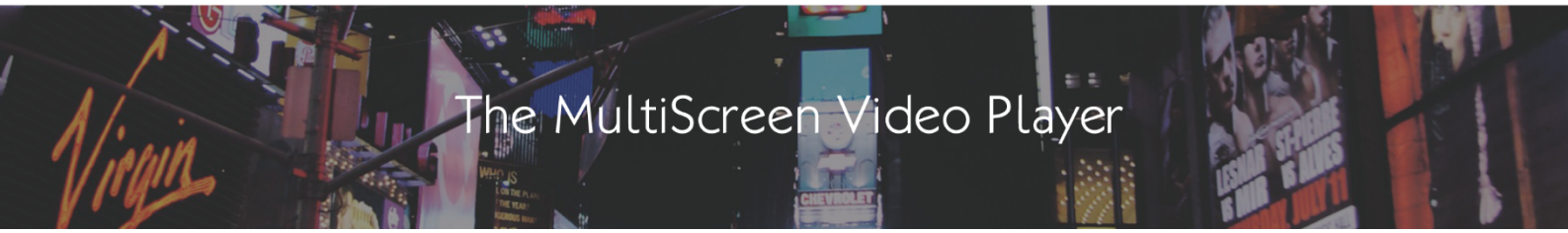


Player SDK Method summary

Modifier and type	Method and description
void	enableCordova() Enable Cordova mode.
MediaData	**getMediaData**() Obtain an object containing metadata of the current media item.
void	**overrideAdProvider**(java.lang.String providerCode) Manually set the VAST Provider Code (as defined within nexxOMNIA).
void	**overrideAutoPlay**(**AutoPlayMode** mode) Specifies how the autoplay should behave.
void	**overrideAutoPlayNext**(**AutoPlayNextMode** mode) Specifies how autonext should behave.
void	**overrideBanner**(java.lang.String url, int interval) Manually set the VAST Banner URL AND the interval in Minutes.
void	**overrideExitPlayMode**(**ExitPlayMode** mode) Specifies what happen at exit.
void	**overrideMenu**(**MenuVisibilityMode** mode) Specifies how the menu should be shown.
void	**overrideMidroll**(java.lang.String url, int interval) Manually set the VAST MidRoll URL AND the Interval in Minutes.
void	**overridePostroll**(java.lang.String url) Manually set the VAST PostRoll URL



Modifier and type	Method and description
void	**overridePreroll**(java.lang.String url) Manually set the VAST PreRoll URL
void	**overrideStartMuted**(**MutedMode** mode) whether or not to start muted.
void	**setPlayLicence**(**int playLicencse**) Specify domain that should be used
void	**setWebURLRepresentation**(**String** url) Set web url for given media to add in vast url
void	**setDataMode**(**tv.nexx.android.player.omnia.datamode.DataMode dataMode**) Set static or usual method to receive data. by default, API is set.
void	**clearCaches**() Clear all caches
void	**overrideVASTData**(java.lang.String providerCode, java.lang.String urlPreroll, java.lang.String urlMidroll, int intervalMidroll, java.lang.String urlPostroll, java.lang.String urlBanner, int intervalBanner) Manually define all URLs, Provider Data and intervals within one Call, see above.
void	**startPlay**(java.lang.String cid, java.lang.String client, **PlayMode** playMode, java.lang.String param, int startPosition, int startItem)
void	**swap**(java.lang.String param)



Modifier and type	Method and description
void	**swapComplex**(**PlayMode** playMode, java.lang.String param, int startPosition, int startItem) Complete Reset of Player with new Modes.
void	**swapToPos**(int newPosition)
void	**setLoaderSkin**(tv.nexx.android.player.control.LoaderSkin skin) Set loading indicator which should be used. Possible values: DEFAULT, METRO, MATERIAL, DOUBLE_BOUNCE DOUBLE_BOUNCE will be available if *Android-Spinkit* is included into your project. Read more in section below. MATERIAL is only available for devices running >= 21, there is a fallback to DEFAULT

Include Android Spinkit

In order to use the LoaderSkin *DOBLE_BOUNCE* you need to add the following lines to your build.gradle:

```
dependencies {
        ...
        compile 'com.github.ybq:Android-SpinKit:1.0.5'
}
```