

Welcome to use JoywayLib to develop your application!

Android JoywayLib provide a 'aar' lib file, please copy it to ' \app\libs' folder, add follow line in you app' s build.gradle, dependencies section:

```
implementation(name: ' JoywayLib_V1.4.6(25)-release', ext: ' aar')
```

See example:

```
dependencies {  
    implementation(name: ' JoywayLib_V1.4.6(25)-release', ext: ' aar')  
    api fileTree(include: ['*.jar'], dir: 'libs')  
    androidTestImplementation(' androidx.test.espresso:espresso-core:3.2.0', {  
        exclude group: ' com.android.support', module: ' support-annotations'  
    })  
    implementation ' androidx.constraintlayout:constraintlayout:1.1.3'  
    implementation ' com.readystatesoftware.systembartint:systembartint:1.0.4'  
    testImplementation ' junit:junit:4.13'  
    implementation ' com.google.android.gms:play-services-maps:17.0.0'  
    implementation ' com.makeramen:roundedimageview:2.3.0'  
    implementation ' androidx.appcompat:appcompat:1.2.0'  
}
```

Add permissions in app's AndroidManifest.xml file:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />  
<uses-permission android:name="android.permission.READ_PHONE_STATE" />  
<uses-permission android:name="android.permission.RECORD_AUDIO" />  
<uses-permission android:name="android.permission.CAMERA" />  
  
<uses-feature android:name="android.hardware.camera" />  
<uses-feature android:name="android.hardware.camera.autofocus" />  
  
<uses-permission android:name="com.android.launcher.permission.READ_SETTINGS" />  
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />  
<uses-permission android:name="android.permission.BLUETOOTH" />  
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />  
  
<uses-feature  
    android:name="android.hardware.bluetooth_le"  
    android:required="true" />  
  
<uses-permission android:name="com.amap.api.v2.permission.INTERNET" />  
<uses-permission android:name="android.permission.VIBRATE" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.FLASHLIGHT" />
```

```

<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

```

To get permission from user, you need to add code when app startup. Asking user whether allow these permission.

For example, in your first Activity,

```

@Override
protected void onStart() {
    super.onStart();

    String[] permissions = {
        Manifest.permission.BLUETOOTH,
        Manifest.permission.ACCESS_FINE_LOCATION,
        Manifest.permission.ACCESS_COARSE_LOCATION,
        //Manifest.permission.READ_EXTERNAL_STORAGE,
        //Manifest.permission.WRITE_EXTERNAL_STORAGE,
        //Manifest.permission.CAMERA,
        //Manifest.permission.RECORD_AUDIO,
    };
    PermissionUtils.checkAndRequestMorePermissions(this, permissions, 0);
}

```

IMPORTANT!

In `Application.onCreate` event , you need to init our lib modules by follow codes:

```

PathHelper.init(JoywayAlarmApplication.sharedInstance());
AudioMgr.init(JoywayAlarmApplication.sharedInstance());

// you MUST set this KEY to ensure JoywayLib to work properly.
ConvertEx.KEY = "ee607a84-49d6-4ede-aa92-98c11f8a88d2";

// init bluetooth module
BT.init(JoywayAlarmApplication.sharedInstance(), 200, 5000);

//only scan ble tag which is named as 'JW-ALARM'
BT.addScanFilter_tagNameEqual("JW-ALARM");

//when tag is scanned, its default name will be 'My Tag '
BT.setTagDefaultDisplayName("My Tag");

//scan 4500ms, wait 5000ms, then scan 4500ms, and so on...
BT.setScanPeriodTimeLength(4500, 5000);

```

```
//will raise not-scanned-event if not scanned in 60 seconds
```

```
BT.setTagBeaconNotScannedTimeoutTimeLength(60*1000);
```

```
//filter rssi-value-shaking to get stable distance results.
```

```
BT.enableRssiSmoother(true, new float[] {0.1f, 0.15f, 0.5f, 0.15f, 0.1f});
```

```
//to scan 10 seconds, will scan forever if pass in param '-1'
```

```
BT.appendTimeLengthToScan(1000*10);
```

Then, you can listen event from lib anywhere:

Activity.onCreate:

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    mContext = Activity_base.this;
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

    BT.listenTagEvent(this, true);
}
```

So your Activity must implement event handler:

```
class MyActivity extends Activity implements OnTagEventHandler;
```

```
@Override
```

```
public void onTagScanStatusChanged(TagScanEvent event) {}
```

```
@Override
```

```
public void onTagConnectStatusChanged(String tagMac, TagConnectStatus oldStatus,
TagConnectStatus newStatus) {}
```

```
@Override
```

```
public void onTagData(String tagMac, byte[] data_bytes, String data_string) {}
```

```
@Override
```

```
public void onTagRssiChanged(String tagAddress, int oldRssi, int newRssi) {}
```

```
@Override
```

```
public void onDataSentToTag(String tagMac, byte[] data_bytes, String data_string) {}
```

To use bluetooth module in lib, please follow these steps:

- (1) App scan nearby ble devices, store device information in scan result event.
- (2) App can connect device once scanned, and waiting for event of connection.
- (3) Once connection is built, app can send and receive data.