

## Integration Guide QdsInappWrapperLib für Android

Die Einbindung der QdsInappWrapperLib erfolgt analog zur IOLib von INFOnline (siehe INFOnline SZM Library Integration Guide Android).

### Hinweis

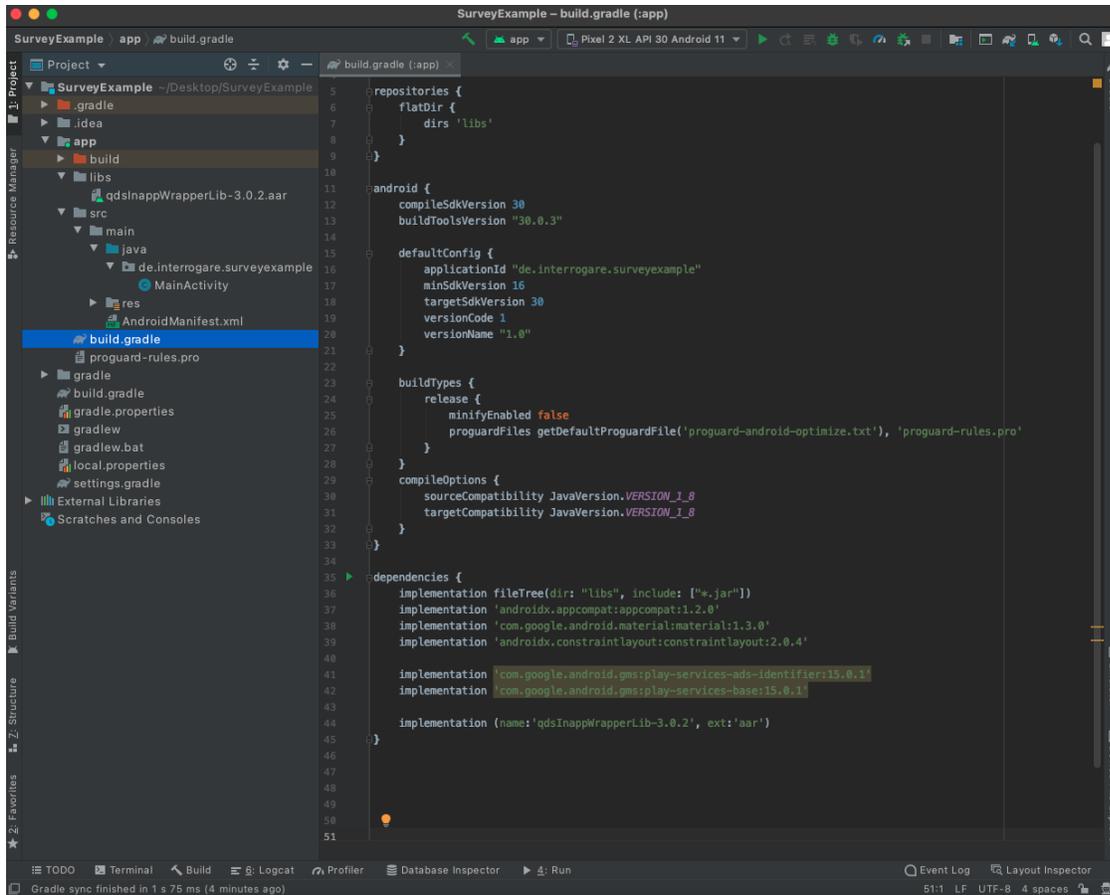
Bitte beachten Sie, dass die Library für In-App-Befragungen der agof nur dann initialisiert werden darf, wenn eine entsprechende Zustimmung des Nutzers zur Teilnahme an den agof-Studien (TCF 2.0 - Purpose 1 und 9 oder inhaltlich identisch) vorliegt (vgl. Integration Guide zur InApp-Messslibrary).

### Einbindungsmöglichkeit in Android Studio (oder IntelliJ IDEA)

Die QdsInappWrapperLib wird als Android Library Project (.aar) importiert. Dazu wird die Datei qdsInappWrapperLib-xxx.aar in den Ordner libs des Anwendungsmoduls kopiert.

In der Datei build.gradle des Anwendungsmoduls müssen die Library selbst und die google-play-services (mind. Version 15) als Abhängigkeit hinzugefügt werden:

```
dependencies {  
    implementation 'com.google.android.gms:play-services-ads-  
identifizier:15.0.1'  
    implementation 'com.google.android.gms:play-services-base:15.0.1'  
  
    implementation (name:'qdsInappWrapperLib-3.0.2', ext:'aar')  
    ...  
}
```



Damit die Abhängigkeit korrekt eingebunden werden kann, muss zur Eigenschaft `repositories` folgender Eintrag hinzugefügt werden:

```

repositories {
    ...
    flatDir {
        dirs 'libs'
    }
}

```

Dabei muss die library im `libs` Ordner des Projektes liegen, da sonst der Pfad nicht passt.

## Einbindungsmöglichkeit in Eclipse

Im Zuge der Umstellung auf das Gradle-Buildsystem und der Veröffentlichung von Android Studio, gibt es seitens Google keine offizielle Unterstützung für den Import von `.aar`-Dateien in Eclipse. Einen inoffiziellen Weg bietet z.B. das `gradle-eclipse-aar-plugin` (siehe: <https://github.com/ksoichiro/gradle-eclipse-aar-plugin>).

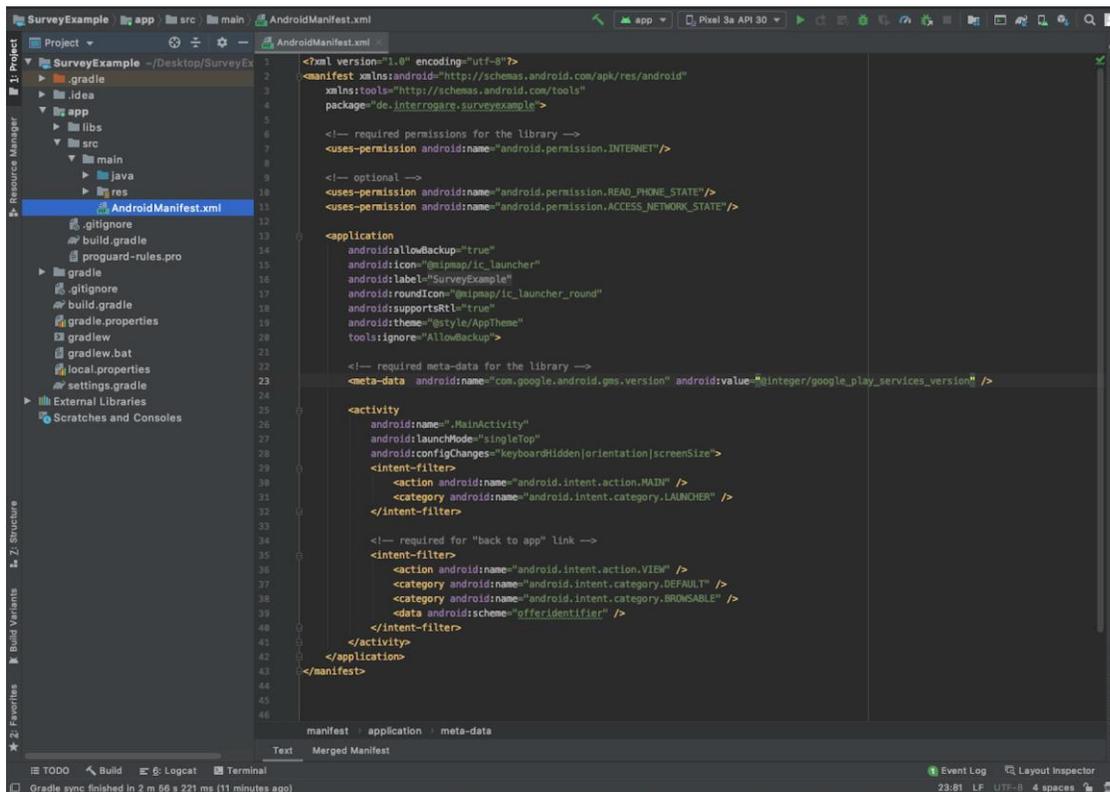
### Wichtig:

*Folgende Informationen müssen in `AndroidManifest.xml` der App eingetragen werden:*

Sollte die IO Lib bereits eingebunden sein, müssen die Permissions schon gesetzt sein. Die folgenden Permissions sind in der Datei AndroidManifest.xml einzutragen, falls keine von diesen vorhanden sind.

```
<!-- required permissions for the library -->
<uses-permission android:name="android.permission.INTERNET" />

<!-- optional -->
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
...
<application>
...
<!-- required meta-data for the library -->
<meta-data android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />
...
</application>
```



Um die korrekte Anzeige des Befragungsdialogs nach Bildschirmrotation zu gewährleisten, sollten in der entsprechenden Activity (welche die QdsInappWrapperLib aufruft) noch zwei weitere Eigenschaften ergänzt werden:

```
<activity
...
    android:launchMode="singleTop"
    android:configChanges="orientation|screenSize"
```

```
>  
...  
</activity>
```

INFO: Ab dem (minimalen) API-Level 11 (Android 3.0) kann auf das Setzen der Eigenschaft `android:configChanges` verzichtet werden.

## Integration der Rücksprungadresse (optional)

Der Rücksprung zur App geschieht durch das Registrieren eines Benutzerdefinierten-URL-Schemas. Dabei muss zur entsprechenden Activity ein Intent-Filter hinzugefügt werden:

```
<activity  
...  
    android:launchMode="singleTop"  
    android:configChanges="orientation|screenSize">  
...  
    <intent-filter>  
        <action android:name="android.intent.action.VIEW"/>  
        <category android:name="android.intent.category.DEFAULT"/>  
        <category android:name="android.intent.category.BROWSABLE"/>  
        <data android:scheme="offeridentifler" />  
    </intent-filter>  
</activity>
```

Der Ausdruck `offerIdentifler` muss mit dem von INFOnline vergebenen Kennzeichner ersetzt werden (Siehe: Funktionsweise).

## Funktionsweise

Die `QdsInappWrapperLib` wird durch die Klasse `QdsInappWrapper` repräsentiert. Eine konkrete Instanz dieser Klasse wird erzeugt mit:

```
new QdsInappWrapper(Activity activity);
```

Die `QdsInappWrapperLib` muss beim Aufruf der App initialisiert werden. Dieses geschieht über den oben erwähnten Aufruf. Für `activity` sollte die `MainActivity` übergeben werden. Um das Anzeigen einer Befragungseinladung zu ermöglichen muss die Methode

```
showSurveyInvitation(String offerIdentifler, String countryCode, Boolean  
userHasAcceptedPermissionForTracking);
```

an der Instanz aufgerufen werden; z.B. in der `MainActivity` in der Methode `onStart()`. Hier müssen der `offerIdentifler`, der Ländercode (mit "de") und die Information, ob die App die Erlaubnis zum Tracking vom Nutzer eingeholt hat oder nicht, übergeben werden. Der Parameter `offerIdentifler` ist die eindeutige Kennung des Angebots der jeweiligen App. Der `offerIdentifler` wird von INFOnline pro App und pro Betriebssystem eindeutig vergeben. Der Parameter `userHasAcceptedPermissionForTracking` ist das flag, ob der Nutzer einem Tracking zugestimmt hat oder nicht.

### **Wichtig:**

Im besten Fall sollen alle anfänglichen Ladeprozesse abgeschlossen sein, da unmittelbar nach dem Aufruf von `showSurveyInvitation()` eine Befragungseinladung angezeigt werden kann.

```
QdsInappWrapper qdsInappWrapper = new QdsInappWrapper(this);  
qdsInappWrapper.showSurveyInvitation("offerIdentifizier", "de", true);
```

Falls beim Start der App Ladedialoge angezeigt werden, sollte `showSurveyInvitation()` erst im Anschluss aufgerufen werden, da sonst der Dialog mit der Einladung zu früh angezeigt werden kann. Generell bedeutet der Aufruf von `showSurveyInvitation()`, dass nach dem Aufruf der Methode ein Dialog mit einer Befragungseinladung angezeigt werden kann. `showSurveyInvitation()` sollte dann aufgerufen werden, wenn die App bereit ist, bei Bedarf einen Fragebogen anzuzeigen. Grundsätzlich kann `showSurveyInvitation()` immer wieder aufgerufen werden.

### **Hinweise:**

Die Initialisierung der `QdsInappWrapperLib` sollte beim Start der „MainActivity“ geschehen. Falls dies aus technischen oder inhaltlichen Gründen nicht möglich ist, muss als Alternative ein geeigneter Einstiegspunkt vom Entwickler gewählt werden. Wichtig ist, dass der Benutzer die Möglichkeit bekommt mit der entsprechenden Activity und dem potentiellen Umfragedialog zu interagieren.

Die **Opt-Out-Funktion** wird von der `QdsInappWrapperLib` über die Befragungseinladung gesteuert, somit ist dafür keine weitere Aktion notwendig. Studien- und Datenschutzinformationen zur agof Studie werden dem Befragungsteilnehmer innerhalb der Befragung bei Bedarf zur Verfügung gestellt.

### **Test - Einblendung Dialogfenster:**

Um das Gerät als Testgerät zu markieren, verwenden Sie bitte den INFOnline SZM-Checker. Die Dokumentation zum INFOnline SZM-Checker erhalten Sie unter [https://www.infonline.de/downloads/9-1356-942/INFOnline\\_Manual\\_SZM-Checker\\_InAPP-Befragung.pdf](https://www.infonline.de/downloads/9-1356-942/INFOnline_Manual_SZM-Checker_InAPP-Befragung.pdf). Bitte überprüfen Sie die Einblendung des Befragungseinladung, die Klickbarkeit aller Schaltflächen sowie das Öffnen des Fragebogens durch einen Klick auf die `Teilnehmen` Schaltfläche.

Der Kunde ist für den Einbau der `QdsInappWrapperLib` im Einklang mit den jeweils gültigen Richtlinien der Prüforganisation verantwortlich. Der Kunde ist dafür verantwortlich, dass auch bei einer Änderung der App die Implementierung der Library korrekt ist.

### **Supportkontakt:**

agof Service Center

Tel.: 0800 / 410 29 77 (Anruf kostenlos)

Aus dem Ausland: 0049 – (0)228 / 410 29 47

E-Mail: [servicecenter@agof.de](mailto:servicecenter@agof.de)