

Integration und Konfiguration der Android Tracking Library von Webtrekk

Version 4.0

Inhalt

1	Vorwort.....	4
1.1	Systemvoraussetzungen.....	4
1.2	Dateien.....	4
1.3	Begriffe.....	4
2	Integration der Tracking Library.....	5
3	Integration der Tracking Funktionalität.....	6
4	Tracking Konfiguration.....	6
4.1	Konfiguration.....	7
4.2	Konfigurationsparameter.....	7
4.2.1	webtrekk_track_domain.....	7
4.2.2	webtrekk_track_id.....	7
5	Tracking Parameter.....	7
5.1	Übergabe von Tracking Parametern.....	8
5.2	Übergabe von individuellen Tracking Parametern.....	8
5.3	Automatisch erfasste Tracking Parameter.....	9
5.3.1	SCREEN_RESOLUTION.....	9
5.3.2	SCREEN_DEPTH.....	9
5.3.3	DEV_LANG.....	9
5.3.4	TIMEZONE.....	9
5.3.5	OS_NAME("osname").....	9
5.3.6	OS_VERSION.....	9
5.3.7	DEVICE.....	9
5.3.8	TRACKING_LIB_VERSION.....	9
5.3.9	APP_LANGUAGE.....	9
5.3.10	EVERID.....	9
5.3.11	ACTIVITY_NAME.....	9
5.4	Optionale automatisch erfasste Tracking Parameter.....	9
5.4.1	APP_VERSION_NAME.....	10
5.4.2	IP.....	10
5.4.3	API_LEVEL.....	10
5.4.4	APP_FIRST_START.....	10
5.4.5	APP_VERSION_CODE.....	10
5.4.6	APP_UPDATE.....	10
5.4.7	APP_PREINSTALLED.....	10
5.4.8	PLAYSTORE_GNAME.....	10
5.4.9	PLAYSTORE_SNAME.....	11
5.4.10	PLAYSTORE_MAIL.....	11
5.4.11	ADVERTISER_ID.....	11
5.4.12	ADVERTISER_OPTOUT.....	11
5.5	Manuell erfassbare Tracking Parameter.....	11
6	Tracking Implementierung.....	13
6.1	Tracking des Aufrufs einer Activitys.....	13
6.2	Tracking von Aktionen.....	14
6.3	Contentgruppen (Seitenkategorien).....	14
6.4	Tracking von Seitenparametern (Eigene Parameter).....	15
6.5	Produkterfassung.....	16
6.5.1	Produktname.....	16
6.5.2	Produktanzahl (optional).....	16
6.5.3	Produktpreis (optional).....	16
6.5.4	Währungscode (optional).....	16

6.5.5Warenkorbstatus (optional).....	16
6.5.6Gutscheinwert (optional).....	16
6.5.7Produktkategorien (optional).....	17
6.5.8Beispiel.....	17
6.6Media Tracking und Arbeit mit Scopes.....	18
6.7Anwendungsdetails.....	18
6.8Webtrekk Ever-ID.....	18
6.9Opt Out.....	18
7Kontakt.....	18

1 Vorwort

Mit Hilfe der Android Tracking Library ist es möglich Nutzungshäufigkeit und Aktivitäten Ihrer Android Apps in Webtrekk zu erfassen. Dieses Dokument beschreibt die technische Integration des Trackings in Ihre Android App.

Um die Analysemöglichkeiten von Webtrekk optimal einzusetzen, sollten Sie sich in der Konzeptionsphase an unseren Schulungsunterlagen „Grundlagen zur Datenerfassung“ orientieren. Sollte Ihnen das Dokument nicht vorliegen, können Sie sich jeder Zeit an support@webtrekk.com oder Ihre(n) persönlichen Consultant wenden.

Durch den Einsatz der Tracking API werden App-Aufrufe und Aktivitäten des Nutzers als Requests an das Webtrekk Tracking System geschickt.

In Abhängigkeit zu dem Onlinestatus des Nutzers werden die Requests an Webtrekk entsprechend gesendet:

1. Ist das mobile Gerät online, werden die Requests in einem definierbaren Zeitintervall gesendet.
2. Ist das Gerät offline, werden die Aktivitäten zwischengespeichert und versendet, sobald das Gerät wieder online ist.

1.1 Systemvoraussetzungen

Trackbare Systeme: Alle Geräte mit dem Betriebssystem Android Version 4.0 oder höher

Entwicklungsumgebung: Die Einbindung in die App ist grundsätzlich mit jeder Entwicklungsumgebung möglich, die vorliegende Dokumentation bezieht sich auf Android Studio, da dies die von Google mitentwickelte Standardentwicklungsumgebung für Android ist.

1.2 Dateien

Die folgenden Dateien sind für die Einbindung des Tracking SDK notwendig:

webtrekksdk-release.aar: Diese Datei enthält das SDK im .aar Format. Das .aar Format ist die Binärdistribution eines Android Bibliothek Projektes.

wt_tracking_config.xml: Dies ist die Hauptkonfigurationsdatei für das Tracking, Anpassungen am Verhalten des SDK können hier entsprechend der Dokumentation vorgenommen werden.

webtrekk_sdk_4_de.pdf: Die Dokumentation zur Integration und Konfiguration des SDK.

MyApplication.zip: Dies ist eine mit Android Studio erstellte Beispielanwendung, welche Sie direkt als Projekt importieren können. Sie zeigt die Initialisierung und Konfiguration des Tracking.

1.3 Begriffe

2 Integration der Tracking Library

Bevor Sie mit dem Tracking Ihrer App starten können, muss die Library in ihr Android Projekt eingebunden werden.

Nachfolgend werden die Schritte aufgezeigt wie dies unter Android Studio mit Gradle als Buildsystem erfolgen kann. Die Pfade beziehen sich dabei auf das Android Default Projekt Layout.

1.

Erstellen Sie, falls noch nicht vorhanden, im Hauptordner Ihres zu trackenden Android-Modules einen Ordner „libs“

2.

Kopieren Sie die Datei webtrekk-release.aar in diesen libs Ordner

3.

Fügen Sie diesen Ordner in der Gradle Build Datei als lokales repository des Android-Modules hinzu, indem sie in der build.gradle Datei des Moduls folgende Zeilen ergänzen:

```
repositories{
    flatDir{
        dirs 'libs'
    }
}
```

4.

Ergänzen sie die .aar Datei als Abhängigkeit, ebenfalls in der build.gradle Datei des Moduls.

```
dependencies {
    ...
    compile 'com.android.support:appcompat-v7:22.0.0'
    compile(name:'wtracksdk-release', ext:'aar')
    ...
}
```

5.

Permissions anpassen:

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
```

Wenn Sie den Benutzernamen oder anderen Profilinformationen aus dem Playstore wie die Email tracken wollen, müssen zusätzlich die folgenden beiden Permissions ergänzt werden. Wenn Sie diese Parameter nicht verwenden, sollten sie auch die Permissions nicht setzen.

```
<uses-permission android:name="android.permission.READ_PROFILE" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
```

3 Integration der Tracking Funktionalität

1.

Die Webtrekk Tracking Funktionalität wird über die Webtrekk Klasse zur Verfügung gestellt. Diese ist als Singleton implementiert und muss einmalig vor Nutzung initialisiert werden. Dies erfolgt am besten in der onCreate Methode Ihrer MainActivity..

```
private Webtrekk webtrekk;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    webtrekk = Webtrekk.getInstance();
    webtrekk.initWebtrekk(this);
    // optional für das automatische Aktivitätstracking
    webtrekk.initAutoTracking(getApplication());
}
```

2.

Kopieren Sie die Konfigurationsdatei für das Tracking in den Ressourcen Ordner Ihres Android Moduls unter: module/res/values/webtrekk_config.xml. Falls das Tracking individuell angepasst werden soll, kann dies entsprechend dem Abschnitt Konfiguration in dieser Datei erfolgen.

3.

Ab jetzt können Sie die grundlegende Nutzung der Aktivitäten innerhalb der App tracken. Rufen Sie dazu in der onStart() Methode der zu trackenden Aktivität die webtrekk.startActivity Funktion auf mit dem Namen der Aktivität als Parameter. In der onStop Methode rufen Sie dazu analog die webtrekk.stopActivity Funktion auf. Zwischen diesen Aufrufen können sie weitere individuelle Tracking Requests versenden. Dies erfolgt jeweils über die webtrekk.track Funktion. Ausführlichere weitere Beispiele finden sie im weitere Verlauf der Dokumentation.

```
@Override
public void onStart()
{
    super.onStart();
    webtrekk.startActivity("MainActivity");
    webtrekk.track();
}
@Override
public void onStop()
{
    webtrekk.stopActivity();
    super.onStop();
}
```

4 Tracking Konfiguration

Die Tracking Konfiguration erfolgt über die bereits erwähnte **wt_tracking_config.xml**:

4.1 Konfiguration

Der folgende Ausschnitt zeigt die minimal Konfiguration für das Tracking, alle anderen Parameter sind optional:

```
<?xml version="1.0" encoding="utf-8"?>
<webtrekkConfiguration>
<version>2</version>
<trackDomain type="text">http://q3.webtrekk.net</trackDomain>
<trackId type="text">11111111111</trackId>
<sampling type="text">0</sampling>
<initial_send_delay type="text">0</initial_send_delay>
<send_delay type="text">60</send_delay>
<max_requests type="number">5000</max_requests>
<auto_tracked type="text">false</auto_tracked>
<enable_plugin_hello_world>true</enable_plugin_hello_world>
<trackingConfigurationUrl>http://localhost/tracking_config.xml</trackingConfigurationU
rl>
    <activity>
        <classname type="text">MainActivity</classname>
        <mappingname type="text">Startseite</mappingname>
        <autotrack type="text">true</autotrack>
    </activity>
    <customvalues>
        <exampleval1>custom example val 1</exampleval1>
    </customvalues>
</webtrekkConfiguration>
```

Setzen Sie hier den Konfigurationsparameter trackDomain und trackId auf die Werte, welche Sie von Webtrekk erhalten haben.

Wenn Sie einen Wert setzen oder überschreiben wollen, geben sie als Namen den jeweiligen **Konfigurationsparameter** an und setzen Sie den gewünschten **Wert**.

4.2 Konfigurationsparameter

4.2.1 trackDomain

Optional: Nein

Wert: muss eine gültige Webtrekk Tracking Domain enthalten

Beschreibung:

Die gesammelten Requests werden an diese Webtrekk Tracking Domain gesendet.

4.2.2 trackId

Optional: Nein

Wert: muss eine gültige Webtrekk Tracking ID enthalten

Beschreibung:

Die Webtrekk Tracking ID, welche Ihnen zugewiesen wurde.

4.2.3 Version

Optional: Nein

Wert: muss einen gültigen Integer Wert enthalten

Beschreibung:

Die Versionsnummer ist lediglich wichtig wenn Online Updates der Konfiguration genutzt werden. Hier wird dann jeweils die aktuellste Version genutzt.

4.2.4 sampling

Optional: ja

Wert: muss einen gültigen Integer Wert enthalten

Beschreibung:

Wenn nur ein Teil der Nutzerdaten ausgewertet werden soll, erfolgt dies Anhand des samplings. Der Parameter legt den Anteil der Nutzer fest welche erfasst werden. Ein Sampling von 10 bedeutet das jeder 10. Nutzer erfasst wird.

4.2.5 send_delay

Optional: Nein

Wert: muss einen gültigen Integer Wert enthalten

Beschreibung:

Gibt das Intervall an, in dem die Tracking Requests an den Webtrekk Server übertragen werden.

4.2.6 initial_send_delay

Optional: Nein

Wert: muss einen gültigen Integer Wert enthalten

Beschreibung:

Gibt die initiale Wartezeit an, nach der die ersten Tracking Requests an den Webtrekk Server übertragen werden.

4.2.7 max_requests

Optional: Nein

Wert: muss einen gültigen Integer Wert enthalten

Beschreibung:

Gibt die Maximale Anzahl der Requests an die auf dem Gerät zwischengespeichert werden.

4.2.8 auto_tracked

Optional: Ja

Wert: true/false

Beschreibung:

Wenn das automatische Tracking aktiviert ist wird das Starten und Stoppen von Aktivitäten automatisch erfasst.

4.2.9 trackingConfigurationUrl

Optional: Ja

Wert: muss eine gültige URL enthalten

Beschreibung:

Falls auto_tracked aktiviert ist, wird beim Start der getrackten App versucht von dieser URL eine neue XML Konfiguration zu laden, welche die aktuellen Parameter überschreibt.

5 Tracking Parameter

Nachdem Sie das Tracking integriert und konfiguriert haben, können Sie mit dem Tracking beginnen. Der nachfolgende Abschnitt zeigt die verschiedenen Möglichkeiten des Tracking auf und wie diese implementiert werden können.

Das Tracken eines Ereignisses erfolgt immer über eine gültige Webtrekk Instanz und den Aufruf der track() Funktion wie nachfolgend gezeigt:

```
import com.webtrekk.webbtrekksdk.TrackingParameter.Parameter;
import com.webtrekk.webbtrekksdk.TrackingParameter;
...
public void onButtonActionClicked(View view) {
    TrackingParameter tp = new TrackingParameter();
    tp.add(Parameter.ACTION_NAME, "Action Button clicked");
    webtrekk.track(tp);
}
```

In dem Beispiel wird ein neues TrackingParameter Objekt erzeugt, dann wird der Trackingparameter Parameter.ACTION_NAME hinzugefügt und abschließend der Trackingrequest versendet.

Analog diesem Beispiel sind alle weiteren Trackingaufrufe aufgebaut. Sie bestehen immer aus den folgenden drei Schritten:

1. Trackingparameter Objekt erstellen
2. Trackingparameter hinzufügen
3. Trackingrequest versenden

5.1 Übergabe von Tracking Parametern

Über die verschiedenen Tracking Parameter wird dem Webtrekk Tracking Server mitgeteilt wie die enthaltenen Werte verarbeitet werden sollen. Die folgende Liste zeigt eine Übersicht über die verfügbaren Tracking Parameter.

Der Einsatz ist immer identisch zu folgendem Beispiel:

```
TrackingParameter tp = new TrackingParameter();  
tp.add(Parameter.XXX, "WERT")
```

Sie übergeben den Tracking Parameter und entsprechend den Wert.

5.2 Übergabe von individuellen Tracking Parametern

Die folgenden Tracking Parameter können Sie im Webinterface individuell konfigurieren.

PAGE, SESSION, ECOM, AD, ACTION, USER_CAT, PAGE_CAT, PRODUCT_CAT, MEDIA_CAT

Deren genaue Bedeutung können Sie im Abschnitt 5.5 nachlesen. All diese Parameter erlauben es Ihnen in der Webtrekk Konfigurationsoberfläche eigene Parameter zu hinterlegen.

Beispielsweise die Ecommerce Parameter.

Im Webinterface konfigurieren sie die Parameter 1 = Colour, 2 = Size, ...

Dann können sie auf einer Activity wie folgt die individuellen E-Commerce Parameter übergeben:

```
tp.add(Parameter.ECOM, 1, "red");  
tp.add(Parameter.ECOM, 2, "XXL");
```

Zusätzlich zum Wert des Tracking-Parameters kommt also noch der von Ihnen im Webinterface definierte Index hinzu. Der Aufruf der track() Funktion ist identisch.

5.3 Automatisch erfasste Tracking Parameter

Die folgenden Parameter werden von der Tracking Library automatisch erfasst und immer mit übertragen.

5.3.1 SCREEN_RESOLUTION

Erfasst die Auflösung des Gerätes.

5.3.2 SCREEN_DEPTH

Farbtiefe des Gerätes.

5.3.3 DEV_LANG

Die Sprache des Gerätes.

5.3.4 TIMEZONE

Die Zeitzone des Gerätes.

5.3.5 OS_NAME("osname")

Das Betriebssystem des Benutzers.

5.3.6 OS_VERSION

Die Version des Betriebssystems, z.B. 4.1

5.3.7 DEVICE

Die Bezeichnung des Endgerätes z.B. Samsung Galaxy S5.

5.3.8 TRACKING_LIB_VERSION

Die Version der Tracking Library.

5.3.9 APP_LANGUAGE

Die Sprache der getrackten Anwendung.

5.3.10 EVERID

Eine eindeutige ID des Benutzers welche zur Identifikation verwendet wird.

5.3.11 ACTIVITY_NAME

Der Name der Android Activity von der aus der aktuelle Tracking Request übermittelt wird.

5.4 Optionale automatisch erfasste Tracking Parameter

Die folgenden Parameter werden auf Wunsch von der Tracking Library automatisch erfasst und versendet. Falls Sie einen dieser Werte nicht zu übertragen wünschen, können Sie den entsprechenden Parameter auf false setzen in der XML Konfigurationsdatei.

```
<bool name="auto_track_app_update">true</bool>
<bool name="auto_track_advertiserid">false</bool>
```

Das Beispiel zeigt die entsprechende Konfiguration, um einen Parameter automatisiert zu erfassen, setzen sie den Wert auto_track_PARAMETER_NAME auf true. Analog dazu können Werte die Sie nicht benötigen auf false gesetzt werden.

5.4.1 APP_VERSION_NAME

Überträgt den Namen der Android App Version z.B. 2.7.5.

5.4.2 IP

Erfasst die IP-Adresse des Smartphones. Hier wird die lokale IP Adresse des WLAN Interfaces erfasst. Die Remote Adresse sehen sie im Request. Über das setzen der IP ist es möglich den Nutzer zu anonymisieren, da dieser Wert serverseitig nicht ausgewertet wird, wenn ein eigener Wert gesetzt wurde.

5.4.3 API_LEVEL

Erfasst das Android API-Level.

5.4.4 APP_FIRST_START

Wenn APP_FIRST_START in der XML Konfiguration auf true gesetzt ist, wird beim ersten Starten der App auf einem Gerät ein Tracking Request versendet.

5.4.5 APP_VERSION_CODE

Immer Wenn Sie eine neue Version ihrer App im Playstore veröffentlichen erhält diese eine eindeutige Integer Zahl zur Identifizierung. Diese ist aufsteigend und ermöglicht das erkennen von Updates.

5.4.6 APP_UPDATE

Wenn APP_UPDATE Tracking auf true gesetzt ist, wird immer wenn der APP_VERSION_CODE Parameter erhöht ein entsprechender Tracking Request verschickt, anhand dessen Sie erkennen können das die App aktualisiert wurde.

5.4.7 APP_PREINSTALLED

Erfasst ob die App bereits auf dem Gerät vorinstalliert war.

5.4.8 PLAYSTORE_GNAME

Erfasst den Vornamen des Benutzers des ersten Google Kontos eines Gerätes. Benötigt zusätzlich die Android Permissions „READ_PROFILE“.

5.4.9 PLAYSTORE_SNAME

Erfasst den Nachnamen des Benutzers des ersten Google Kontos eines Gerätes. Benötigt zusätzlich die Android Permissions „READ_PROFILE“.

5.4.10 PLAYSTORE_MAIL

Erfasst die mit dem ersten Google Konto des Gerätes assoziierte Email-Adresse. Benötigt zusätzlich die Android Permissions „READ_PROFILE“.

5.4.11 ADVERTISER_ID

Die Advertiser ID ist eine eindeutige Benutzerkennung die Google über den Playstore an jeden Benutzer vergibt. Diese ermöglicht es Benutzer wiederzuerkennen auch über verschiedene Geräte hinweg, solange sie sich mit dem selben Google Account anmelden. Der Benutzer kann diese Advertiser ID zurücksetzen. Gemäß den Playstore AGB ist dies die einzige für Werbung zulässige Methode um Benutzer zu identifizieren.

5.4.12 ADVERTISER_OPTOUT

Gibt dem Anwender der App die Möglichkeit keine Werbung zu sehen.

5.5 Manuell erfassbare Tracking Parameter

Die folgenden Parameter können zusätzlich manuell erfasst werden. Dazu müssen sie wie im obigen Beispiel gezeigt dem TrackingParameter Objekt hinzugefügt werden.

Parameter	Beschreibung
ACTIVITY_CATEGORY	Kategorie einer Aktivität
BIRTHDAY	Geburtstag
CAMPAIGN	Kampagne
CITY	Stadt
COUNTRY	Land
VOUCHER	Gutscheinwert
CURRENCY	Währung
CAMPAIGN_PARAMS	Kapagnen Parameter
CUSTOMER_ID	Kundennummer
EMAIL	Email
EMAIL_RID	
NEWSLETTER	
GNAME	
SNAME	
SSL	
GENDER	
INTERN_SEARCH	
ACTION_NAME	

Parameter	Beschreibung
ORDER_NUMBER	
ORDER_TOTAL	
ZIP	
STREET	
STREETNUMBER	
PHONE	
PRODUCT	Produkt
PRODUCT_COST	Produktpreis
PRODUCT_COUNT	Produktmenge
PRODUCT_STATUS	Produktstatus
USER_CATEGORY	
IP_ADDRESS	
TIMESTAMP	
USERAGENT	
ADVERTISEMENT	
MEDIA_FILE	Mediendatei
MEDIA_ACTION	Aktion (play, pause, stop, pos, seek, eof)
MEDIA_POS	Position
MEDIA_LENGTH	Länge der Mediendatei
MEDIA_BANDWIDTH	Bandbreite der Mediendatei
MEDIA_VOLUME	Lautstärke
MEDIA_MUTED	Stummschaltung
MEDIA_TIMESTAMP	Zeitstempel für die Mediendatei, um Caching zu umgehen

Ergänzend zu den bisher gelisteten Parametern gibt es auch noch Parameter, die Sie im Webinterface individuell einstellen können. Details zur Implementierung dieser stehen im Abschnitt 6.

Parameter	Beschreibung
PAGE	Seitenparameter
SESSION	Session Parameter
ECOM	Ecommerce Parameter
AD	Werbe Parameter
ACTION	Actions Parameter
USER_CAT	Benutzer Kategorien
PAGE_CAT	Seiten Kategorien
PRODUCT_CAT	Product Kategorien

5.6 Globale Tracking Parameter

Neben den individuellen TrackingParametern die Sie je nach Aktivität setzen, gibt es auch die Möglichkeit globale Tracking Parameter zu setzen, welche für alle Requests innerhalb der getrackten App gelten.

Dazu gibt es in der globalen Instanz der Webtrekk Klasse ein TrackingParameter Object, welches sie Abfragen und bearbeiten können.

```
TrackingParameter globalTrackingParameter = webtrekk.getGlobalTrackingParameter();
globalTrackingParameter.add(Parameter.ECOM, "1", "GLOBALTEST");
```

In dem gezeigten Beispiel wird dann für die gesamte Anwendung der ECOM Parameter „1“ mit GLOBALTEST belegt und entsprechend gesendet. Hier können Werte hinterlegt werden die Benutzerspezifisch sind oder aber sich nicht mit dem Wechseln der Aktivitäten ändern.

6 Tracking Implementierung

Wie auf einer Onlineplattform geht auch das App-Tracking davon aus, dass die Anwendung aus einzelnen Seiten(Activitys) besteht deren Nutzung erfasst werden soll. Die Tracking Implementierung ermöglicht Ihnen die Übergabe von Anwendungsinhalten wie z.B. Seiten oder E-Commerce-Werten. Die Inhalte werden als Page-Impression gewertet und erscheinen in der Webtrekkoberfläche in der Seitenanalyse.

Die nachfolgenden Beispiele gehen jeweils davon aus das sie bereits ein gültige Instanz der **Webtrekk** Klasse erzeugt haben mit dem Namen „webtrekk“. Dies kann beispielsweise in der onCreate Methode einer Activity erfolgen, oder aber in den individuellen Methoden einer Activity/Java Klasse.

6.1 Tracking des Aufrufs einer Activitys

Das einfachste Beispiel ist es, zu Tracken wenn eine Activity ausgeführt wird.

```
webtrekk.track();
```

Mit diesem Aufruf werden alle oben erläuterten automatisch erfassbaren Parameter übertragen inklusive des Names der aktuellen Aktivität.

Den vollständigen Beispielcode finden sie in der Demo in der PageExampleActivity.

6.2 Tracking von Aktionen

Mit dem Aktions-Tracking können Sie Ereignisse in der App wie zum Beispiel Klicks auf Buttons oder Auswählen von Checkboxes verfolgen.

Die Auflistung der gemessenen Aktionen erfolgt im Tool unter „Navigation > Aktionen“.

Anders als beim Erfassen von Standard-Webseiten, kann Webtrekk nicht automatisch erkennen auf welcher Seite ein Klick stattgefunden hat. Deshalb muss beim mobilen Aktionstracking immer der Aktionsname angegeben werden.

Beispiel:

```
public void onButtonActionClicked(View view) {  
    TrackingParameter tp = new TrackingParameter();  
    tp.add(Parameter.ACTION_NAME, "Action Button clicked");  
    webtrekk.track(tp);  
}
```

Aktionsparameter müssen in der Konfiguration (Konfiguration > Eigene Parameter > Aktionsparameter) eingerichtet werden, bevor sie erfasst werden. Bei der Einrichtung wird u.a. die ID und der Datentyp (Text/Zahl) je Parameter festgelegt.

Analog zum Seiten-Tracking können Sie bei jedem Request weitere Parameter wie z.B. Platzierung oder Farbe eines Buttons anhängen.

Beispiel:

```
public void onButtonActionParamsClicked(View view) {  
    TrackingParameter tp = new TrackingParameter();  
    tp.add(Parameter.ACTION, "Action Button clicked")  
        .add(Parameter.ACTION, "1", "grey")  
        .add(Parameter.ACTION, "2", "pos3");  
    webtrekk.track(tp);  
}
```

...

6.3 Contentgruppen (Seitenkategorien)

Mit Contentgruppen können Seiten zusammengefasst und somit Bereiche Ihrer App gebildet werden.

Contentgruppen können als Text oder Zahl erfasst werden. Der Datentyp „Text“ kann dabei auch zur Abbildung der Hierarchie der App dienen. Contentgruppen vom Datentyp „Zahl“ können verwendet werden, um jeden Aufruf einer Seite mit dem für sie hinterlegten Zahlwert zu bewerten.

Contentgruppen müssen im Webtrekk Tool konfiguriert werden, damit sie gemessen werden. Die Konfiguration erfolgt unter „Konfiguration > Kategorisierung > Contentgruppen“. Bei der Einrichtung wird u.a. die ID und der Datentyp (Text/Zahl) je Contentgruppe festgelegt.

Contentgruppen können mit der Tracking Library übergeben werden, oder durch einen Import per Excel oder SOAP-Schnittstelle.

Die Auflistung der gemessenen Contentgruppen vom Datentyp „Text“ erfolgt im Webtrekk Tool unter „Navigation > Contentgruppen“. Contentgruppen vom Datentyp „Zahl“ stehen als Metrik in den Analysen zur Verfügung.

Beispiel:

```
public void onCheckboxActionClicked(View view) {
    TrackingParameter tp = new TrackingParameter();
    tp.add(Parameter.PAGE_CAT, "1", "Herren")
        .add(Parameter.ACTION, "2", "Schuhe")
        .add(Parameter.ACTION, "3", "Sportschuhe");
    webtrekk.track(tp);
}
```

Contentgruppen-Parameter dürfen die Länge von 255 Zeichen nicht überschreiten. Alle weiteren Zeichen werden gelöscht.

Contentgruppen werden einmalig einer Seite zugeordnet. Wird eine Seite zusammen mit einer Contentgruppe gemessen, werden alle folgenden Aufrufe dieser Seite auch dieser Contentgruppe zugeordnet. Bei der Übergabe der Contentgruppe mit der Tracking Library wird entsprechend nur der initiale Aufruf einer Seite (ContentID) beachtet. Folglich empfiehlt Webtrekk bei der Pixeleinbindung ContentIDs und Contentgruppen zeitgleich zu implementieren.

6.4 Tracking von Seitenparametern (Eigene Parameter)

Mit "Eigenen Parametern" (paketabhängig) können Sie die Analysedaten mit Ihren App spezifischen Informationen bzw. Metriken anreichern.

Seitenparameter müssen in der Konfiguration (Konfiguration > Eigene Parameter > Seitenparameter) eingerichtet werden, bevor sie erfasst werden. Bei der Einrichtung wird u.a. die ID und der Datentyp (Text/Zahl) je Parameter festgelegt.

Seitenparameter beziehen sich auf eine einzelne Seite und werden dieser direkt zugeordnet. Im Gegensatz zur Contentgruppe muss dabei der Bezug zwischen der Seite und dem Seitenparameter nicht eindeutig sein. Damit können Sie den Aufruf einer Seite spezifizieren, z.B. durch Angabe der Variante oder einen Zahlwert.

Die Auflistung der gemessenen Seitenparameter vom Datentyp „Text“ erfolgt im Webtrekk Tool unter „Navigation > Seitenparameter“. Seitenparameter vom Typ „Zahl“ stehen als Metrik in den Analysen zur Verfügung.

Beispiel:

```
TrackingParameter tp_pageparams = new TrackingParameter();
tp_pageparams.add(Parameter.PAGE, "1", "green")
    .add(Parameter.PAGE, "2", "4")
    .add(Parameter.PAGE, "3", "234");
webtrekk.track(tp_pageparams);
```

6.5 Produkterfassung

Die folgenden Parameter dienen der detaillierten Messung von Produkten. Die Produkte können an Webtrekk übermittelt werden, wenn ein Produkt angesehen, in den Warenkorb gelegt oder wenn der Warenkorb gekauft wurde. Die Auflistung der gemessenen Produkte erfolgt im Webtrekk Tool unter „E-Commerce > Produkte“. Nutzen Sie bitte keine Tausendertrennzeichen in den Preisangaben.

6.5.1 Produktname

Speichert die Produkte, die in den Warenkorb gelegt wurden. Wenn mehrere Produkte im Warenkorb liegen, werden diese jeweils mit einem Semikolon getrennt. Dieser Parameter muss zwingend ausgefüllt werden, wenn Produkte gemessen werden sollen. Alle weiteren Parameter sind für die Produktmessung optional. Jedes Einzelprodukt darf nicht mehr als 110 Zeichen enthalten.

6.5.2 Produktanzahl (optional)

Enthält die Produktanzahl. Wenn mehrere Produkte übertragen werden, werden diese jeweils mit einem Semikolon getrennt. Der Standardwert ist „1“.

6.5.3 Produktpreis (optional)

Enthält den Produktpreis („0“-Preise sind zulässig). Wenn Sie ein Produkt mehrfach übergeben (Produktanzahl größer 1), nutzen Sie den Gesamtpreis, nicht den Einzelpreis. Wenn mehrere Preise übertragen werden, werden diese jeweils mit einem Semikolon getrennt. Der Standardwert ist „0“.

6.5.4 Währungscode (optional)

Enthält den Währungscode eines Produktes oder einer Bestellung, dabei muss der Wert nach ISO-Standard an das Webtrekk Pixel übergeben werden. Werden auf einer Seite mehrere Produkte übermittelt (z.B. auf der Bestellbestätigungsseite, wenn mehr als 1 Produkt gekauft wurde) gilt für alle Produkte nur 1 Währung. Dementsprechend muss der Wert auch nur einmal gesetzt werden.

Hinweis: Die Übergabe der Währung dient lediglich der Währungsumrechnung. D.h. ggf. wird in die im Webtrekk-Frontend hinterlegte Währung (Konfiguration -> Systemkonfiguration: Account) umgerechnet. Dort wird immer nur eine Währung ausgewiesen.

6.5.5 Warenkorbstatus (optional)

Enthält den Status des Warenkorbs. Wird ein Produkt angesehen (z.B. auf einer Produktdetailansicht), lautet der Status „view“. Dieser Status sollte immer dann gesetzt werden, wenn das Produkt in den Warenkorb gelegt werden kann.

Wenn ein Produkt in den Warenkorb gelegt wurde, lautet der Status „add“. Wurde der Warenkorb

gekauft, wird der Status „conf“ übergeben. Wird bei Tracking eines Produktes kein Status übergeben, wird der Standardwert „view“, also eine Produktansicht, angenommen.

6.5.6 Gutscheinwert (optional)

Enthält den Wert eines Gutscheins. Nutzen Sie diesen Parameter, wenn der Kunde eine Bestellung mit einem Gutschein tätigt.

6.5.7 Produktkategorien (optional)

Mit Produktkategorien können Produkte zusammengefasst werden. Der Bezug zwischen Produkt und Produktkategorie muss eindeutig sein. Es ist also z.B. nicht möglich, dass das Produkt „Schuhe“ einmal der Produktkategorie „Damen“ und einmal der Produktkategorie „Sale“ zugeordnet wird. Solche nicht eindeutigen Beziehungen lassen sich über E-Commerce-Parameter abbilden. Produktkategorien vom Datentyp „Zahl“ können verwendet werden, um jeden Aufruf eines Produktes mit dem hinterlegten Zahlwert zu bewerten. Produktkategorien müssen im Webtrekk Tool konfiguriert sein, damit Webtrekk diese Information misst. Die Konfiguration erfolgt unter „Konfiguration > Kategorisierung > Produktkategorien“. Jede Produktkategorie darf nicht mehr als 110 Zeichen enthalten. Bis auf den Parameter „Warenkorbstatus“ müssen alle anderen Parameter die gleiche Anzahl von Objekten enthalten, z.B. 2 Produkte, 2 Produktpreise, 2 Kategorien, usw.

Die Auflistung der gemessenen Produktkategorien vom Datentyp „Text“ erfolgt im Webtrekk Tool unter „E-Commerce > Produktkategorien“. Produktkategorien vom Datentyp „Zahl“ stehen als Metrik in den Analysen zur Verfügung.

Produktkategorien werden einmalig einem Produkt zugeordnet. Wird ein Produkt zusammen mit einer Kategorie gemessen, werden alle Produkte auch dieser Kategorie zugeordnet. Wenn vor dem Kauf eines Produktes zwingend der Produktstatus „view“ aufgerufen werden muss, reicht es daher, wenn Produktkategorien nur dort übergeben werden.

6.5.8 Beispiel

Den Beispielcode finden Sie in unserer Tracking Demo App in der ShopExampleActivity.

```
tp = new TrackingParameter();
tp.add(Parameter.PRODUCT, "Brauner Herrenschuh Leder: Mike")
    .add(Parameter.PRODUCT_CAT, "1", "Herren")
    .add(Parameter.PRODUCT_CAT, "2", "Schuhe")
    .add(Parameter.PRODUCT_CAT, "3", "Leder")
    .add(Parameter.PRODUCT_COUNT, "1")
    .add(Parameter.PRODUCT_COST, "99,95")
    .add(Parameter.PRODUCT_STATUS, "view")
    .add(Parameter.CURRENCY, "EUR");
webtrekk.track(tp);
```

Wenn Sie das TrackingParameter Objekt tp in den Scope der Klasse legen, brauchen Sie wenn der Kunden dann das Objekt mit dem ButtonOrder Klick kauft nur die geänderten Parameter

übergeben.

Beispiel:

```
public void onButtonOrderClicked(View view) {
    tp.add(Parameter.PRODUCT_STATUS, "add")
    .add(Parameter.ACTION_NAME, "orderButton");
    webtrekk.track(tp);
}
```

6.6 Media Tracking und Arbeit mit Scopes

Webtrekk bietet Ihnen die Möglichkeit, die Mediennutzung in Ihrer App zu erfassen. Das Tracking erfolgt analog zu den bisher gezeigten Beispielen.

In der Beispiel App sehen Sie eine konkrete Implementierung der hier besprochenen Punkte.

Um eine Media Datei zu tracken, richten Sie zunächst die entsprechenden TrackingParams ein, dort können Sie den Dateinamen, die Länge, die Bandbreite und weitere Details konfigurieren.

Beispielcode:

```
        tp = new TrackingParams();
        tp.add(TrackingParams.Params.MEDIA_FILE,
getResources().getResourceEntryName(R.raw.marv3));
        tp.add(TrackingParams.Params.MEDIA_LENGTH,
String.valueOf(getDuration()));
...
```

Im Anschluss daran, können Sie in den entsprechenden Aktionsmethoden wie Play, Pause, Stop, Suche jeweils einen Tracking Request versenden. Das folgende Beispiel zeigt eine Erweiterung der Android MediaPlayer Klasse welche die Methoden der Basis Klasse um das Tracking ergänzt.

Beispielcode:

```
public class TrackedVideoView extends VideoView {
    private Webtrekk webtrekk;
    TrackingParameter tp;
    public TrackedVideoView(Context context) {
        super(context);
        initMediaFile();
    }
    public TrackedVideoView(Context context, AttributeSet attrs) {
        super(context, attrs);
        initMediaFile();
    }
    public TrackedVideoView(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);
        initMediaFile();
    }
    public void initMediaFile() {
        tp = new TrackingParameter();
        tp.add(TrackingParameter.Parameter.MEDIA_FILE,
getResources().getResourceEntryName(R.raw.marv3));
        tp.add(TrackingParameter.Parameter.MEDIA_LENGTH,
String.valueOf(getDuration()));
        tp.add(TrackingParameter.Parameter.MEDIA_POS,
String.valueOf(getCurrentPosition()));
    }
}
```

```

        tp.add(TrackingParameter.Parameter.MEDIA_CAT, "1", "mp3");
        tp.add(TrackingParameter.Parameter.MEDIA_CAT, "1", "example");
    }
    @Override
    public void pause() {
        super.pause();
        tp.add(TrackingParameter.Parameter.MEDIA_ACTION, "pause");
        tp.add(TrackingParameter.Parameter.MEDIA_POS,
String.valueOf(getCurrentPosition()));
        webtrekk.track(tp);
    }
    @Override
    public void start() {
        super.start();
        tp.add(TrackingParameter.Parameter.MEDIA_ACTION, "start");
        tp.add(TrackingParameter.Parameter.MEDIA_POS,
String.valueOf(getCurrentPosition()));
        webtrekk.track(tp);
    }
    @Override
    public void seekTo(int msec) {
        super.seekTo(msec);
        tp.add(TrackingParameter.Parameter.MEDIA_ACTION, "seek");
        tp.add(TrackingParameter.Parameter.MEDIA_POS, String.valueOf(msec));
        webtrekk.track(tp);
    }
    @Override
    public void stopPlayback() {
        super.stopPlayback();
        tp.add(TrackingParameter.Parameter.MEDIA_POS,
String.valueOf(getCurrentPosition()));
        tp.add(TrackingParameter.Parameter.MEDIA_ACTION, "stop");
        webtrekk.track(tp);
    }
    public Webtrekk getWebtrekk() {
        return webtrekk;
    }
    public void setWebtrekk(Webtrekk webtrekk) {
        this.webtrekk = webtrekk;
    }
}

```

Sie können neben den Aktionsparametern wie play und pause auch mit jedem TrackingRequest weitere Parameter versenden, z.B. die Media Kategorien, oder aber die aktuelle Position in der Mediendatei.

Da das TrackingParameter Objekt im Scope der TrackedVideoView Klasse liegt, müssen Sie die Werte auch nur einmalig setzen und trotzdem werden sie bei jedem nachfolgenden Request übertragen. Solange bis der Scope ungültig ist, oder aber die TrackingParameter explizit gelöscht werden.

Dadurch können sie auch bei den einzelnen Aktionen sehen, auf welche Mediendatei sich diese beziehen, ohne das die Parameter wie Dateiname erneut gesetzt werden muss.

6.7 Anwendungsdetails

Hier die Möglichkeit rein ob die App vorinstalliert ist, geupdated wurde usw

6.8 Webtrekk Ever-ID

Die Webtrekk Ever-ID wird verwendet um Nutzer zu erkennen. Auf einer Standardwebseite wird diese automatisch gesetzt und übermittelt.

Da die technischen Möglichkeiten zu Nutzererkennung auf mobilen Geräten eingeschränkt sind, haben Sie die Möglichkeit diese ID manuell auszulesen und zu übermitteln, wenn Ihre Nutzer z.B. von der App in den mobilen Browser wechseln. Somit behält ein solcher Nutzer während dem Besuch der Webseite die Ever-ID aus der App. Der Besuch kann somit übergreifend gemessen werden.

Zum Auslesen der Ever-ID können Sie unten beschriebene Methode verwenden.

```
String everId = GetWtrack().getEverId();
```

Das Übermitteln an den Browser erfolgt durch das Anhängen zweier Parameter an den Ziel-Link (URL).

1. „wt_eid“: die Ever-ID
2. „wt_t“: aktueller Unix-Timestamp in Millisekunden. Dieser ist notwendig, dass die Ever-ID in der URL eine maximale Gültigkeit von 15 Minuten hat. Somit wird die Wahrscheinlichkeit, dass ein solcher Link z.B. gepostet wird und dieser so häufig mit denselben IDs gewertet wird, minimiert (sonst würden alle Besucher dieselbe EverId erhalten und so als 1 Besucher zählen!).

Beispieldomain:

http://new.domain.com/page?wt_eid=2135817235100536325&wt_t=1358414378580

Beim Klick auf den Link öffnet sich der Webbrowser und der Nutzer gelangt auf die neue Seite. Dort erhält er die gleiche Ever-ID, die er auch in der App besessen hat.

Zusätzlich dazu können Sie noch einen eigenen Referrer mitgeben, damit der Einstieg in Ihre Webseite nicht als Direkt ausgewiesen wird. Zum Simulieren eines eigenen Referrers hängen Sie zusätzlich den URL-Parameter „wt_ref“ an die Link-URL. Auch hier wird wieder der Parameter „wt_t“ verwendet, der eine Gültigkeit von 15 Minuten hat.

Beispiel:

http://new.domain.com/start.html?wt_eid=2135817235100536325&wt_ref=http%3A%2F%2Fwww.webtrekk.com%2Fen%2Fhome.html&wt_t=1358414378580

Bitte achten Sie darauf, dass die Referrer-URL kodiert werden muss.

Hinweis: Durch falsche Handhabung dieses Features kann es passieren, dass viele Nutzer die gleiche Ever-ID erhalten. Dies würde extreme Auswirkungen auf Ihre Datenqualität mit sich ziehen.

Bitte achten Sie bei Einsatz auf die korrekte Verwendung!

6.9 Opt Out

Nach §15 des Telemediengesetzes können Webseitenbesucher der Datenspeicherung Ihrer anonymisiert erfassten Besucherdaten widersprechen, so dass sie in Zukunft nicht mehr erfasst werden. Um diesen Widerspruch zu gewährleisten, muss die Methode „setOptedOut“ genutzt werden.

Beispielcode:

```
webtrekk.setOptout(true);
```

Ebenso können sie über das Tracker Objekt auch prüfen ob der Nutzer dem Tracking zugestimmt hat.

Beispielcode:

```
...  
webtrekk.isOptout();
```

7 Kontakt

Wenn Sie Fragen zur Einrichtung haben sollten, stehen wir Ihnen selbstverständlich zur Verfügung. Für priorisierten Support oder umfassendere Beratung bietet Webtrekk Support- und Consultingpakete an. Sprechen Sie uns an, wir unterbreiten Ihnen gerne ein individuelles Angebot.

Webtrekk GmbH

Robert-Koch-Platz 4

10115 Berlin

fon 030 - 755 415 - 0

fax 030 - 755 415 - 100

support@webtrekk.com

<http://www.webtrekk.com>