

Integration und Konfiguration der Android Tracking Library von Webtrekk

Version 3.0



Inhalt

1	Vorwort	3
1.1	Systemvoraussetzungen	3
1.2	Integration der Tracking Library	3
2	Tracking-Funktionalität	4
2.1	Initialisieren des Trackings	4
2.2	Starten und Stoppen des Trackings	5
2.3	Seiten-Tracking	6
2.3.1	Contentgruppen (Seitenkategorien)	7
2.3.2	Seitenparameter (Eigene Parameter)	7
2.3.3	Produkterfassung	8
2.3.4	Messung von Bestellungen	10
2.3.5	E-Commerce-Parameter (Eigene Parameter)	11
2.3.6	Kampagnen	12
2.3.7	Sessionparameter	13
2.3.8	Interne Suche	14
2.3.9	Eigene Besucher-IDs	14
2.4	Aktions-Tracking	15
2.5	Media Tracking	16
2.5.1	Play	16
2.5.2	Position	17
2.5.3	Pause	17
2.5.4	Seek	17
2.5.5	SeekEnd	17
2.5.6	Stop	18
2.6	App Version	18
2.7	App Updates	19
2.8	Vorinstallierte Apps	20
3	Webtrekk Ever-ID	20
4	Opt-Out Funktionalität	21
5	Parameterübersicht	22
6	Kontakt	23

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

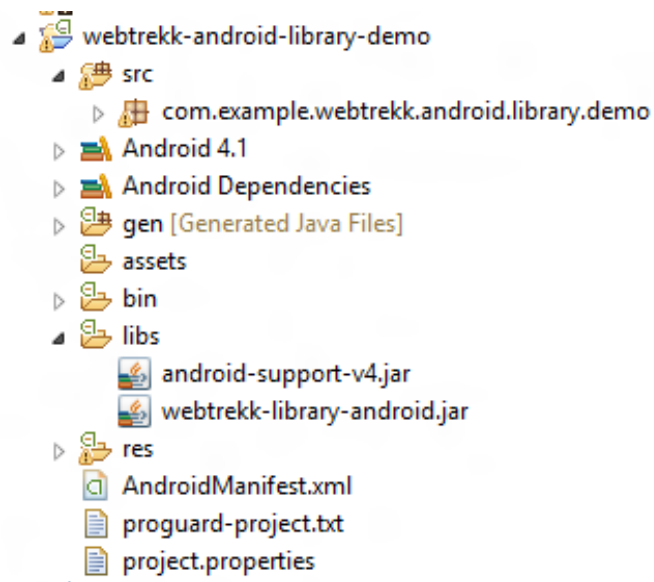
Trackbare Systeme: Alle Geräte mit dem Betriebssystem Android Version 2.3 oder höher.

Entwicklungsumgebung: Die Einbindung in die Android App erfolgt über Eclipse oder wahlweise Android Studio.

1.2 Integration der Tracking Library

Bevor Sie mit der Verpixelung der App starten, muss die Library in Ihr Android-Projekt eingebunden werden.

Fügen Sie die .jar -Datei und dessen Build Path zum Projekt hinzu. Im Falle des Android SDKs unter Eclipse ist dies der „libs“ Ordner des Projekts.



Damit das Android Betriebssystem die Netzwerkanfragen nicht blockt, muss im Manifest des Android App Projekts die Permission „Internet“ gesetzt sein.

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

2 Tracking-Funktionalität

Die Tracking-Library erlaubt es, Seiten- und Aktionsrequests an Webtrekk zu übermitteln. Genau wie in einer Standard-Webseitenverpixelung können diese Requests durch weitere Parameter angereichert werden.

Das Tracking eines App-Besuchs ist dabei immer analog aufgebaut.

- Starten des trackings beim Starten der App
- Senden beliebig vieler Tracking-Requests während der Nutzung
- Beenden des trackings beim Stoppen der App

2.1 Initialisieren des Trackings

Zur Nutzung der Tracking-Klassen innerhalb einer Quellcodedatei, müssen Sie diese importieren.

```
import com.webtrekk.android.tracking.Webtrekk;
```

Um Requests an Webtrekk zu übermitteln müssen zunächst die Basisinformationen an die Tracking Library übergeben werden. Zu diesem Zweck nutzen sie am besten die „Application“ – Klasse, die jede Android App definieren kann:

```
public class Application extends android.app.Application {  
  
    @Override  
    public void onCreate() {  
        super.onCreate();  
  
        Webtrekk.setContext(this);  
        Webtrekk.setServerUrl("http://q3.wt-eu02.net");  
        Webtrekk.setTrackId("11111111111111");  
  
        Webtrekk.setLoggingEnabled(true); // optional  
        Webtrekk.setSamplingRate(0);      // optional  
        Webtrekk.setSendDelay(5*60*1000); // optional  
        Webtrekk.setAppVersionParameter("cs5"); // optional  
    }  
}
```

Bitte übergeben Sie der Klasse „Webtrekk“, Ihre Webtrekk Track-Domain, Webtrekk Track-ID und den App Context.

In Absprache mit Webtrekk kann optional eine Sampling Rate angegeben werden. Das Sampling unterdrückt die Requests an Webtrekk um den angegebenen Wert, d.h. es wird nur jeder n-te User erfasst. Die Daten werden dann bei der Analyse entsprechend hochgerechnet (Konfiguration muss bei Webtrekk vorgenommen werden).

Mit der Methode „setSendDelay“ können Sie das Zeitintervall in Sekunden definieren, in dem die Requests verschickt werden. Standardmäßig ist das Intervall auf 5 Minuten gestellt. Um Akkuleistung zu sparen, können Sie das Intervall z.B. auf 10 Minuten stellen.

Des Weiteren können Sie mit „setLoggingEnabled“ das Debugging aktivieren bzw. deaktivieren.

Beim Verschieben der App in den Hintergrund oder beim direkten Schließen der App werden sämtliche noch nicht versendeten Request an Webtrekk übermittelt. Insofern eine Internetverbindung vorliegt.

Ebenfalls können Sie einen eigenen Parameter definieren, welcher die aktuelle Version Ihrer Applikation in jedem Request mitschickt, siehe Kapitel 2.6.

2.2 Starten und Stoppen des Trackings

Das Tracking sollte in den Lifecycle Methoden der Activities „onStart“ und „onStop“ erfolgen.

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

    Webtrekk.activityStart(this);    // Aktiviert das App-Tracking
}

@Override
protected void onStop() {
    Webtrekk.activityStop(this);    // Deaktiviert das App-Tracking

    super.onStop();
}
```

Hinweis: Für die Methoden „Webtrekk.activityStart“ und „Webtrekk.activityStop“ wird eine aktive Activity als Parameter erwartet, da diese jedoch nicht immer verfügbar ist, steht für diesen Usecase ein Workaround bereit.

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

    // kann z.B. vom Service bezogen werden
    final Context applicationContext = ...;

    // fake-activity als Workaround
    Webtrekk.activityStart(new Activity() {
        {
            this.attachBaseContext(applicationContext);
        }
    });
}
```

2.3 Seiten-Tracking

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

Beispiel für eine vollständige Konfiguration:

```
Webtrekk.trackPage("de.myPage");
```

Diese Requests unter können mit weiteren Parametern angereichert werden. Dies sind z.B. Bestell- und Produktinformationen in Onlineshops oder auch eigene Seitenparameter und -Kategorien.

2.3.1 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 für eine vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();  
parameter.put("cg1", "damen");           // Hauptkategorie  
parameter.put("cg2", "oberbekleidung");  // Unterkategorie  
  
Webtrekk.trackPage("de.myPage", parameter);
```

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.

2.3.2 Seitenparameter (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 (siehe Kap. 2.3.1) 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 für eine vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();  
parameter.put("cp1", "green");  
parameter.put("cp2", "200.51");  
  
Webtrekk.trackPage("de.myPage", parameter);
```

2.3.3 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 keine Tausendertrenner in den Preisangaben. Dezimalstellen werden per Punkt oder Komma getrennt.
Order / Basket Daten werden beim „Aktionstracking“ **nicht** verarbeitet.

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.

Produktanzahl (optional)

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

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“.

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.

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.

Gutscheinwert (optional)

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

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 (siehe Kap. 2.3.5) 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.

Beispiel für eine vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();
parameter.put("ba", "pullover;jeans");           // Produktname
parameter.put("ca1", "oberbekleidung;hosen");    // Produktkategorie 1
parameter.put("ca2", "noname;levis");             // Produktkategorie 2
parameter.put("qn", "2;1");                       // Produktanzahl
parameter.put("co", "99.90;69.95");              // Produktkosten
parameter.put("st", "view");                      // Warenkorbstatus
parameter.put("cr", "EUR");                      // Währungscode

Webtrekk.trackPage("de.myProductView", parameter);
```

2.3.4 Messung von Bestellungen

Webtrekk bietet die Möglichkeit, Bestellungen zu messen. Neben der Bestellnummer wird dazu der Bestellwert übertragen. „0“-Werte sind zulässig. Die Auflistung der gemessenen Bestellungen erfolgt im Webtrekk Tool unter „E-Commerce > Bestellungen“.

Der Unterschied zum Produkttracking ist, dass sich die Informationen nicht auf einzelne Produkte beziehen, sondern wie der Gesamtbestellwert Informationen zur Bestellung übermittelt übertragen werden. Der Gesamtbestellwert kann z.B. neben der Summe der gekauften Produkte auch Rabatte, Versandkosten und Verpackungskosten beinhalten.

Nutzen Sie keine Tausendertrenner in den Preisangaben. Dezimalstellen werden per Punkt oder Komma getrennt.

Der Parameter „Gesamtbestellwert“ muss zwingend ausgefüllt werden, wenn Gesamtbestellwerte gemessen werden sollen.

Der Parameter „Bestellnummer“ (optional) enthält eine eindeutige ID, die der Bestellung zugeordnet werden kann. Die Nutzung dieser Einstellung gewährleistet, dass keine Bestellungen doppelt gezählt werden.

Die Währung einer Bestellung kann hier auch übertragen werden.

Beispiel für eine vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();  
parameter.put("oi", "M-12345"); // Bestellnummer  
parameter.put("ov", "52.99"); // Gesamtbestellwert  
parameter.put("cb563", "10.00"); // Gutscheinwert  
  
Webtrekk.trackPage("de.myOrder", parameter);
```

2.3.5 E-Commerce-Parameter (Eigene Parameter)

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

E-Commerce-Parameter müssen in der Konfiguration (Konfiguration > Eigene Parameter > E-Commerce-Parameter) eingerichtet werden, bevor sie erfasst werden. Bei der Einrichtung wird u.a. eine für das Tracking relevante ID und ein Datentyp (Text/Zahl) für den Parameter festgelegt.

Bezug des Parameters

Mit E-Commerce-Parametern können weitere Produktinformationen (z.B. Größe, Farbe) übergeben werden. Bei mehreren Produkten muss die Anzahl der einzelnen Parameter-Werte mit der Anzahl der Produkte übereinstimmen. Die einzelnen Werte werden mit Semikolon getrennt.

E-Commerce-Parameter können aber auch verwendet werden, um Informationen zu einer Bestellung zu übergeben, z.B. Bezahlart, Versandart. In diesen Fällen ist das Messen von Bestellungen zwingend notwendig. Es reicht, diese Parameter je Bestellung einmal zu übergeben. Sie gelten gleichermaßen für alle Produkte im Warenkorb.

Der Bezug (Produkt oder Bestellung) wird bei der Konfiguration der Tracking Library gewählt. Ist „einzelner Wert“ gewählt bezieht sich der Parameter auf die Bestellung. Ist „mehrere Werte“ kann sich der Partner sowohl auf Produkt als auch Bestellung beziehen.

Hinweis: Da Webseitenziele in Webtrekk immer als E-Commerce Parameter erfasst werden müssen ist es auch möglich E-Commerce Parameter losgelöst von Bestellungen und Produkten zu übergeben.

E-Commerce-Parameter vom Typ „Text“ werden im Webtrekk Tool unter „E-Commerce > E-Commerce-Parameter“ ausgewiesen. E-Commerce-Parameter vom Typ „Zahl“ stehen in den Analysen als Metrik zur Verfügung.

Beispiel für vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();
parameter.put("ba", "pullover;jeans"); // Produktname
parameter.put("ca1", "oberbekleidung;hosen"); // Produktkategorie 1
parameter.put("ca2", "noname;levis"); // Produktkategorie 2
parameter.put("qn", "2;1"); // Produktanzahl
parameter.put("co", "99.90;69.95"); // Produktkosten
parameter.put("st", "view"); // Warenkorbstatus
parameter.put("cr", "EUR"); // Währungscode

parameter.put("oi", "M-12345"); // Bestellnummer
parameter.put("ov", "52.99"); // Gesamtbestellwert
parameter.put("cb563", "10.00"); // Gutscheinwert

parameter.put("cd1", "L;32"); // Produktparameter 1
parameter.put("cb2", "grün;blau"); // Produktparameter 2
parameter.put("cb3", "rechnung"); // Produktparameter 3

Webtrekk.trackPage("de.myOrder", parameter);
```

2.3.6 Kampagnen

Das Kampagnen-Tracking wird im Webtrekk Tool konfiguriert (Konfiguration > Kampagnenkonfiguration). Ohne diese Konfiguration werden keine Kampagneninformationen, wie z.B. Kampagnenklicks, erfasst. Als Kampagnenklick kann unter anderem der Aufruf bestimmter Seiten in einer App oder die Erfassung definierter Verweise gemessen werden.

Es besteht die Möglichkeit, selbst eine Kampagnen-ID im Konfigurationsteil zu setzen. Eine Kampagnen-ID besteht aus einem Mediacode-Namen und dem dazugehörigen Wert, getrennt durch ein „%3D“.

Ebenso können Kampagnen mittels Kampagnenparameter um Zusatzinformationen ergänzt werden.

Beispiel für vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();
parameter.put("mc", "mc%3Dnewsletter_2014_12"); // Kampagnen-ID

Webtrekk.trackPage("de.myCampaign", parameter);
```

Kampagnenparameter

Mit "Eigenen Parametern" können Sie die Analysedaten mit Ihren App seitenspezifischen Informationen bzw. Metriken anreichern.

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

Kampagnenparameter beziehen sich immer auf ein Werbemittel (kleinste in Webtrekk vorhandene Untereinheit einer Kampagne).

Beispiel für vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();
parameter.put("mc", "mc%3Dnewsletter_2014_12"); // Kampagnen-ID
parameter.put("ccl", "personalized");           // Kampagnenparameter 1

Webtrekk.trackPage("de.myCampaign", parameter);
```

Webtrekk bietet die Möglichkeit, Google Kampagnen automatisch zu erfassen, wenn über diese die App aus dem App-Store geladen und installiert wurde.

Anhand des Referrer-Parameters kann nachvollzogen werden, über welche Wege der Nutzer im Play Store zur App gekommen ist.

Zuerst muss der BroadcastReceiver der Library in das Manifest der Applikation eingetragen werden. Dazu öffnet man die XML Ansicht des Manifests und fügt folgenden Code innerhalb des <application> Tags ein:

```
<receiver
    android:name="com.webtrekk.android.tracking.ReferrerReceiver"
    android:exported="true" >
    <intent-filter>
        <action android:name="com.android.vending.INSTALL_REFERRER" />
    </intent-filter>
</receiver>
```

Der Aufbau von Links zum Play Store samt Informationen zum Kampagnen-Tracking ist auf der folgenden Website dokumentiert: <https://developers.google.com/analytics/devguides/collection/android/v2/campaigns>. Bis auf gclid werden alle dort angegebenen Parameter unterstützt.

Die von uns erfasst Kampagne setzt sich dann wie folgt zusammen:

- MediaCode (wt_mc): utm_source.utm_medium.utm_content.utm_campaign
- Kampagnen Keyword (wt_kw): utm_term

Hinweis: Bitte richten Sie Ihre Kampagnenkonfiguration diesbezüglich ein.

2.3.7 Sessionparameter

Sessionparameter beziehen sich immer auf eine Session, also einen Visit. Wird der Wert für den Parameter innerhalb eines Visits mehrmals übertragen, wird jeweils nur der zuletzt übermittelte Wert ausgewertet.

Zum Beispiel könnte der Status, ob ein User während des Visits eingeloggt war, übergeben werden. Standardmäßig würde jeder Besuch zu Beginn in einem Sessionparameter als „nicht eingeloggt“ gekennzeichnet. Ein-Login wird an den gleichen Parameter übergeben und überschreibt damit den ersten Wert.

Im Unterschied zu einem Seitenparameter kann nicht ausgewertet werden, auf welcher Seite ein Sessionparameter gesetzt wurde. Zudem belässt ein Seitenparameter jeden Wert auswertbar, der innerhalb des Visits gesetzt wurde.

Beispiel für vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();  
parameter.put("cs1", "logged_in");           // Sessionparameter 1  
  
Webtrekk.trackPage("de.myLoginPage", parameter);
```

2.3.8 Interne Suche

Analysieren Sie die Suchbegriffe, die Besucher in Ihrer App eingeben, indem Sie diese im Tracking mitgeben. Tragen Sie dynamisch den genutzten Suchbegriff in den Konfigurationsparameter ein.

Beispiel für vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();  
parameter.put("is", "adidas schuh");        // Suchbegriffe  
  
Webtrekk.trackPage("de.mySearch", parameter);
```

Die Auflistung der gemessenen Sucheingaben erfolgt im Tool unter „Marketing > Suchphrasen > interne Suchphrasen“ bzw. „Marketing > Suchbegriffe > interne Suchbegriffe“.

2.3.9 Eigene Besucher-IDs

Um die Besuchererkennung zu verbessern, können Sie anstatt des Webtrekk Ever-ID, eigene Besucher-IDs verwenden.

Für die Nutzung eigener Besucher-IDs geben Sie der Tracking Library einen eindeutigen Identifikator aus Ihrem System mit. Sollten Sie keine eindeutigen Besucher-IDs in Ihrer App einsetzen, ist alternativ die Emailadresse des Besuchers als eindeutiger Identifikator denkbar. In diesem Fall sollten Sie aus datenschutzrechtlichen Gründen die Emailadresse unlesbar machen (z.B. mit dem MD5 Hash).

Besucher-IDs können z.B. nach einem erfolgreichen Login oder nach einer erfolgreichen Bestellung an die Tracking Library übergeben werden. Diese eigenen Besucher-IDs können im Webtrekk Tool unter „Besucher > Besucher > letzte eigene Besucher-Ids“ ausgewertet werden.

Beispiel für vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();  
parameter.put("cd", "372d1a04d003eebc09e17330d5d3117c"); // Besucher-ID  
  
Webtrekk.trackPage("de.myLoginPage", parameter);
```

Mit den optionalen Besucher Kategorien können Sie zusätzlich den Besucher kategorisieren. Diese URM Kategorien müssen zuvor im Tool angelegt werden. Im unten gezeigten Beispiel wird dem Besucher, der Familienstand zugewiesen.

Beispiel für vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();  
parameter.put("cd", "372d1a04d003eebc09e17330d5d3117c"); // Besucher-ID  
parameter.put("uc1", "ledig"); // Besucher Kategorie 1  
  
Webtrekk.trackPage("de.myLoginPage", parameter);
```

2.4 Aktions-Tracking

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 müssen beim mobilen Aktionstracking immer Aktions- und Seitenname angegeben werden und Webtrekk manuell über diese Ereignisse informieren.

Beispiel für vollständige Konfiguration:

```
Webtrekk.trackAction("de.myPage", "de.myAction");
```

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

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 eine Buttons anhängen.

Beispiel für vollständige Konfiguration:

```
Map<String, String> parameter = new HashMap<String, String>();
parameter.put("ck1", "green"); // Aktions-Parameter 1
parameter.put("ck2", "top"); // Aktions-Parameter 2

Webtrekk.trackAction("de.myPage", "de.myButton", parameter);
```

2.5 Media Tracking

Webtrekk bietet Ihnen die Möglichkeit, die Mediennutzung in Ihrer App zu erfassen.

Das Einrichten des Media Trackings erfolgt in zwei Schritten:

- Das Tracking muss aktiviert sein (siehe Kap. 2.2)
- Starten des Media Trackings für ein Video wenn dieses Video aufgerufen wird.

Hierfür wird eine Instanz der Klasse „MediaSession“ benutzt:

Beispiel für vollständige Konfiguration:

```
MediaCategories mediaCategories = new MediaCategories();
mediaCategories.category1 = "media-category-1";
mediaCategories.category2 = "media-category-2";
mediaCategories.category3 = "media-category-3";

MediaSession mediaTracking = Webtrekk.trackMedia(
    "mediaId", // Name des Videos
    120000, // Dauer des Videos in Millisekunden
    0, // Startposition des Videos in Millisekunden
    mediaCategories // optional - Mediakategorien
);
```

Eine Media-Session wird geöffnet, sobald das Video für den Nutzer sichtbar ist. So wird auch die Anzeige des Videos (auch wenn es noch nicht abgespielt wird) gemessen.

Jede Media-Session wird beendet, sobald die Aktion „MediaEvent.Stop“ erfasst wird. Sollen danach noch weitere Ereignisse folgen, muss vorher eine neue Media-Session geöffnet werden.

2.5.1 Play

Das Play-Ereignis muss gesendet werden, wenn der Nutzer die Wiedergabe des Videos startet oder aber die Wiedergabe nach einem Pause-Ereignis fortsetzt.

Beispiel für vollständige Konfiguration:

```
// Play-Ereignis bei Sekunde 10  
mediaTracking.trackEvent(MediaEvent.Play, 10000);
```

2.5.2 Position

Das Position-Ereignis sollte während der Wiedergabe des Videos im Abstand von 30 Sekunden gesendet werden. So kann die Ansicht des Videos akkurat gemessen werden.

Beispiel für vollständige Konfiguration:

```
// Position-Ereignis bei Sekunde 30  
mediaTracking.trackEvent(MediaEvent.Position, 30000);
```

2.5.3 Pause

Das Pause-Ereignis muss gesendet werden, wenn der Nutzer die Wiedergabe des Videos pausiert.

Beispiel für vollständige Konfiguration:

```
// Pause-Ereignis bei Sekunde 120  
mediaTracking.trackEvent(MediaEvent.Pause, 120000);
```

2.5.4 Seek

Das Seek-Ereignis muss gesendet werden, wenn der Nutzer beginnt, die Position des Videos zu verändern.

Beispiel für vollständige Konfiguration:

```
// Seek-Ereignis bei Sekunde 150  
mediaTracking.trackEvent(MediaEvent.Seek, 150000);
```

2.5.5 SeekEnd

Das Seek End-Ereignis muss gesendet werden, wenn der Nutzer die Positionsänderung innerhalb des Videos abschließt. Es wird die neue Position des Videos übergeben.

Beispiel für vollständige Konfiguration:

```
// SeekEnd-Ereignis bei Sekunde 300  
mediaTracking.trackEvent(MediaEvent.SeekEnd, 300000);
```

2.5.6 Stop

Das Stop-Ereignis muss gesendet werden, wenn der Nutzer die Wiedergabe des Videos stoppt, das Ende des Videos erreicht wurde oder das Video entfernt wird (z.B. wenn die Activity geschlossen oder eine andere geöffnet wird).

Beispiel für vollständige Konfiguration:

```
// Stop-Ereignis bei Sekunde 350  
mediaTracking.trackEvent(MediaEvent.Stop, 350000);
```

Das Stop-Ereignis beendet die Media-Session. Zum messen weiterer Ereignisse (z.B. einer erneuten Wiedergabe), muss die Media-Session neu erstellt wird.

2.6 App Version

Webtrekk bietet Ihnen Zwei Möglichkeiten, um die Versionen Ihrer Applikationen zu messen. Zum einen haben Sie die Möglichkeit beim Initialen Aufruf der Webtrekk SDK, die Methode „setAppVersionParameter“ mit einem von Ihnen gewünschten Parameter aufzurufen. Innerhalb dieses Parameters wird mit jedem Request die aktuelle Version Ihrer Applikation mitversendet.

Hinweis: Durch die Verwendung von „setAppVersionParameter“ wird beim Initialisieren Ihrer Applikation ein separater Seitenrequest erzeugt, wenn ein Update der Applikation stattgefunden hat. Die Content ID dieses zusätzlichen Request lautet „update“ und kann nicht editiert oder überschrieben werden.

Alternativ können Sie den Parameter auch zu einem, von Ihnen gewünschten, späteren Zeitpunkt über die Methode „setAppVersionParameter“ festlegen.

```
public class Application extends android.app.Application {

    @Override
    public void onCreate() {
        super.onCreate();

        Webtrekk.setContext(this);
        Webtrekk.setServerUrl("http://q3.wt-eu02.net");
        Webtrekk.setTrackId("11111111111111");

        Webtrekk.setLoggingEnabled(true); // optional
        Webtrekk.setSamplingRate(0);      // optional
        Webtrekk.setSendDelay(5*60*1000); // optional
        Webtrekk.setAppVersionParameter("cs5"); // optional
    }
}
```

```
// Alternativ können Sie den Parameter auch zu einem späteren Zeitpunkt
// über dieselbe Methode setzen
Webtrekk.setAppVersionParameter("cs5");
```

Sobald Sie einen Parameter für das Versenden der Versionsnummer Ihrer Applikation angeben, wird diese mit jedem Request versendet.

2.7 App Updates

Über die Methode „isThisVersionAnUpdate“ stellt die Webtrekk SDK eine Funktionalität zur Verfügung, um auf ein Update Ihrer Applikation zu prüfen. Somit können Sie nach erfolgreichem Update der Applikation einen manuellen Request an Webtrekk versenden.

```
// Um das Update einer Applikation zu übermitteln
Map<String, String> parameter = new HashMap<String, String>();

if(Webtrekk.isThisVersionAnUpdate()) {
    parameter.put("cs2", "1");
}
else {
    parameter.put("cs2", "0");
}

// Um die aktuelle Version der App ebenfalls zu übermitteln
Webtrekk.setAppVersionParameter("cs5");

Webtrekk.trackPage("app-update", parameter);
```

2.8 Vorinstallierte Apps

Über die Methode „isThisAppPreInstalled“ stellt die Webtrekk SDK eine Funktionalität zur Verfügung, um auf eine Vorinstallation Ihrer Applikation zu prüfen.

Beispiel für vollständige Konfiguration:

```
// Um das Update einer Applikation zu übermitteln
Map<String, String> parameter = new HashMap<String, String>();

if (Webtrekk.isThisAppPreInstalled ()) {
    parameter.put("cp4", "1");
}
else {
    parameter.put("cp4", "0");
}

Webtrekk.trackPage("app-pre-installed", parameter);
```

3 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 eid = Webtrekk.getEverId();
```

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

- „wt_eid“: die Ever-ID
- „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 Ever-ID erhalten und so als 1 Besucher zählen!).

Beispiel für eine URL mit Ever-ID und Timestamp:

```
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 für eine URL mit Ever-ID, Timestamp und Referrer:

```
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!

Dieses Feature setzt voraus, dass Sie auf der Webseite die **Pixelversion 3.2.3** oder höher nutzen.

4 Opt-Out Funktionalität

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.

Beispiel für vollständige Konfiguration:

```
// [Nutzer wird nicht gemessen - true | Nutzer wird gemessen - false]  
Webtrekk.setOptedOut(true);
```

Mit dem Aufruf „optedOut“ können Sie prüfen, ob der Nutzer gemessen werden möchte oder nicht.

Beispiel für vollständige Konfiguration:

```
boolean optOut = Webtrekk.isOptedOut();
```

5 Parameterübersicht

Folgende Parameter können Sie als „optionale“ Parameter an Webtrekk übergeben werden.

Hinweis: Sämtliche Kategorien (z.B. Contentgruppen, Produktkategorien, URM-Kategorien) und eigene Parameter (z.B. Seiten-, Session-, Aktions-, E-Commerce- oder unabhängige Parameter) müssen vorab in der Konfiguration der Webtrekk-Oberfläche anlegen sein.

Parameter	Javascript-Parameter	Beschreibung
cp	wt.customParameter	Seitenparameter [cp2, cp3...]
cs	wt.customSessionParameter	Session Parameter [cs2, cs3...]
cb	wt.customEcommerceParameter	Parameter E-Commerce [cb2, cb3...]
cb563	wt.couponValue	Gutscheinwert
ov	wt.orderValue	Bestellwert
oi	wt.orderId	Order ID
ba	wt.product	Produkte im Warenkorb
co	wt.productCost	Produktkosten
cr	wt.currency	Produkt Währung
qn	wt.productQuantity	Anzahl der Produkte
ca	wt.productCategory	Produktkategorie [ca2, ca3...]
st	wt.productStatus	Status des Warenkorbes [add conf view]
cd	wt.customerId	Kunden-ID
uc	wt.urmCategory	Kunden Parameter [uc2, uc3...]
uc701	wt.emailRID	E-Mail Receiver-ID
uc702	wt.emailOptin	E-Mail Opt-In [1 = yes 2 = no]
uc706	wt.gender	Geschlecht [1 = male 2 = female]
uc707	wt.birthday	Geburtstag [JJJJMDD]
cq	wt.contentGroup	Seitenkategorie [cg2, cg3...]
is	wt.internalSearch	Suchbegriff der internen Suchfunktion
mc	wt.campaignId	Kampagnen ID bestehend aus Mediacode Parameter und Wert („wt_mc%3Dnewsletter“)
ck	wt.customClickParameter	Aktions-Parameter [ck2, ck3...]
cc	wt.customCampaignParameter	Kampagnen Parameter [cc2, cc3...]

6 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>