



# **Specifiche Integrazione Librerie per scarico dati da Sistemi JS e Stitch**

Rev 1.2 - Italian

---

## Indice

1. Premessa.....	3
2. Installazione.....	3
3. Utilizzo e metodi disponibili.....	4
3.1 Scarico dati EVA da sistemi JS.....	4
3.2 Scarico dati da dispositivo Stitch.....	5
3.3 Recupero dei risultati.....	5
4. Android 11 – SDK 30.....	7

---

## 1. Premessa

Tali librerie, per Android, vengono rilasciate su richiesta di sviluppatori esterni per l'integrazione delle funzionalità di scarico dati in formato EVA-DTS proprie dei sistemi di pagamento **JS**, e per l'utilizzo della periferica di download IrDa-Bluetooth denominata **Stitch** di Sistemi Vending, che ne detiene tutti i diritti e la proprietà dei marchi.

## 2. Installazione

Le classi java fornite rappresentano dei wrapper che consentono di utilizzare la funzionalità presenti nella nostra App Ziwi per l'utilizzo dei sistemi JS e Stitch.

All'utente verrà richiesto di effettuare il download dell'App Ziwi, ma **non è necessario** effettuare la registrazione al portale e **nemmeno concedere i permessi**, tranne quello per l'utilizzo del Bluetooth del Device.

Le classi java fornite sono :

- ZiwiIntegrator.java
- ZiwiResponse.java

Le due classi vanno incluse nell'App di destinazione, non necessitano di alcun riferimento esterno tranne che per la libreria Json di Google.

```
implementation 'com.google.code.gson:gson:2.8.6'
```

---

### 3. Utilizzo e metodi disponibili

La classe ZiwiIntegrator vuole sul costruttore il context di riferimento e presenta due chiamate asincrone a funzioni :

#### 3.1 Scarico dati EVA da sistemi JS

Il metodo disponibile nella classe ZiwiIntegrator è :

**void GetEva\_JS(String User\_key, boolean resetEVACounter)**

Richiede lo scarico del file EVA dai sistemi JS fornendo come parametri :

“**User\_key**” di tipo Stringa che è disponibile al proprietario dei sistemi JS sul proprio portale di supporto.

“**resetEVACounter**” di tipo booleano che informa il sistema se resettare i contatori parziali dopo lo scarico.

*Esempio di chiamata :*

```
public void GetEvaJS(View view){
    ziwiIntegrator zIntegrator = new ziwiIntegrator(this);
    String ziwiKey=<user_key>;/
    zIntegrator.GetEVA_JS(ziwiKey,true);
}
```

E' inoltre possibile richiedere lo scarico dati EVA-DTS, da uno specifico JS utilizzando il metodo della classe ZiwiIntegrator :

**void GetEva\_JS(String User\_key, boolean resetEVACounter, String Filter)**

Il parametro di tipo **String “Filter”**, case insensitive, richiede di filtrare tra i dispositivi che Ziwi rileverà per lo scarico solo il dispositivo corrispondente, evitando di visualizzare tutti i JS presenti nel raggio di azione Bluetooth.

*Esempio di chiamata :*

```
public void GetEvaJS(View view){
    ziwiIntegrator zIntegrator = new ziwiIntegrator(this);
    String ziwiKey=<user_key>;/
    zIntegrator.GetEVA_JS(ziwiKey,true,"008254");
}
```

Se il parametro **Filter** è una stringa vuota “”, verranno visualizzati tutti i JS rilevati.

---

## 3.2 Scarico dati da dispositivo Stitch

```
void GetEVA_Stitch(int IrDaSecurity, int IrDaPassword, int boudRate, boolean
resetEVACounter)
```

Richiede lo scarico dati dalla periferica Stitch fornendo come parametri :

“**IrDaSecurity**” di tipo int.

“**IrDaPassword**” di tipo int.

“**boudRate**” di tipo int, rappresenta una velocità prefissata, e può assumere i seguenti valori:

```
IRDABOUD_2400=16;
IRDABOUD_4800=32;
IRDABOUD_9600=48;
IRDABOUD_19200=64;
IRDABOUD_38400=80;
IRDABOUD_57600=96;
IRDABOUD_AUTO=112;
```

“**resetEVACounter**” di tipo booleano che informa il sistema se resettare i contattori parziali dopo lo scarico.

Esempio di chiamata :

```
public void GetEvaStitch(View view){
    ziwiIntegrator zIntegrator = new ziwiIntegrator(this);
    zIntegrator.GetEVA_Stitch(0,0,ziwiIntegrator.IRDABOUD_AUTO,true);
}
```

## 3.3 Recupero dei risultati

La risposta delle chiamate a funzione viene fornita nel metodo **onActivityResult** dell'activity chiamante.

Un esempio dell'utilizzo è il seguente :

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    String res="";
    ziwiResponse zr=new ziwiResponse();
    if (requestCode==0x0000978c){
        if (data!=null && data.getStringExtra("ZIWIRESPONSE")!=null){
            res=data.getStringExtra("ZIWIRESPONSE");
            zr=zr.GetResponse(res);
            Log.d("jj","" + res);
            Log.d("jj","" + zr.getEVA());
        }
    }
}
```

---

La risposta viene fornita sottoforma di stringa JSON nell'extra "ZIWIRESPONSE", e può essere convertito utilizzando il metodo della classe ZiwiResponse.

L'EVA-DTS in formato "leggibile" è disponibile nel metodo getEVA di ZiwiResponse.

ZiwiResponse fornisce :

- status
- errorCode
- msgStatus
- EVA

**status**

*0= Eseguito senza errori*

*1= Eseguito con errori*

*2= In progress*

**errorCode**

Rappresenta il codice di errore (vedi commenti nel codice della classe ZiwiResponse).

**msgStatus**

Messaggio di errore ricevuto.

**EVA**

Rappresentazione in BASE64 della stringa EVA ricevuta.

---

## 4. Android 11 – SDK 30

Nella versione di Android 11, Google ha limitato implicitamente la ricerca dei package installati a meno che non si autorizzi esplicitamente nel file manifest.xml

<https://developer.android.com/about/versions/11/privacy/package-visibility>

Tale funzionalità è necessaria per indicare all'utente di scaricare l'App ziwi qualora sul telefono non ci sia o non sia aggiornata alla versione 7.2 che permette l'integrazione con le App di terze parti.

Pertanto se la propria App ha come target l'SDK 30 :

```
compileSdkVersion 30
```

```
targetSdkVersion 30
```

E' necessario aggiungere nel manifest la richiesta di permesso per la ricerca di ziwi al S.O.

```
<manifest package="com.tua.app">

    <queries>
        <package android:name="sv.ziwi" />
        <package android:name="sv.ziwi.external" />
    </queries>
    ...
</manifest>
```