

Source Documents - Android SDK - Analytics

Error rendering macro 'navitabs' :

com.atlassian.rendererv2.macro.MacroException: The root page with the name 'Analytics SDK (Android)' does not exist in space with key 'DD'!

Version 1.3.4

ON THIS PAGE

This is Phunware's Android SDK for the Analytics module. Visit <http://maas.phunware.com/> for more details and to sign up.

Documentation

MaaS Analytics documentation is included in the Documents folder in the repository as both HTML and as a .jar. You can also find the latest documentation here: [Analytics API Reference \(iOS\)](#)

Overview

The MaaS Analytics SDK provides the ability to generate custom analytic events. Events can be created at a single point in time or with duration data for timed events.

Build Requirements

- Android SDK 2.2+ (API level 8) or above
- latest MaaS Core

Installation

The following libraries are required:

```
MaaSCore.jar
```

MaaS Analytics depends on MaaSCore.jar, which is available here: [Core SDK](#)

It's recommended that you add the MaaS libraries to the 'libs' directory. This directory should contain MaaSCore.jar and MaaSANalytics.jar, as well as any other MaaS libraries that you are using.

Update your `AndroidManifest.xml` to include the following permissions:

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

See [AndroidManifest.xml](#) for an example manifest file.

Prerequisites

The MaaS Analytics SDK requires the latest MaaS Core SDK.

Be sure to install the module in the `Application onCreate` method before registering keys. For example:

```
@Override
public void onCreate() {
    super.onCreate();
    /* other code */
    PwCoreSession.getInstance().installModules(PwAnalyticsModule.getInstance(), ...);
    /* other code */
}
```

Integration

Adding Events

To add events with MaaS Analytics:

```
public class AnalyticsSample
extends Activity
{
    @Override
    public void onCreate()
    {
        super.onCreate();

        /* Other code */

        //Requires a context
        and event name.

        PwAnalyticsModule.addEvent(this, "Featured Page View");

        /* Other code */
    }
}
```

All event names must be alphanumeric strings.

Timed Events

MaaS Analytics supports timed analytics:

```
public void startLevel()
{
    // Start a timed event like
    so:
    PwAnalyticsModule.startTimedE
vent(this, "My Awesome Game -
Level 1");
    // 'This' refers to a
    context.
}

public void endLevel()
{
    // To end a timed event, pass
    in the same event name like
    so:
    PwAnalyticsModule.endTimedEve
nt(this, "My Awesome Game -
Level 1");
    // 'This' refers to a
    context.
}
```

Pausing and Resuming Timed Events

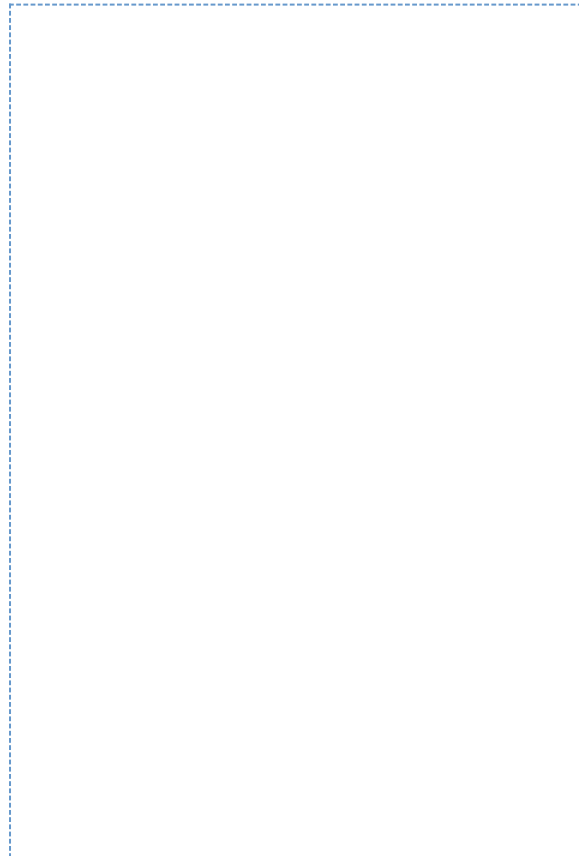
MaaS Analytics allows you to pause and resume timed events. If an event is in a paused state when `endTimedEvent` is called on it, then the paused timestamp will be used to calculate an event's duration.

```
public void pauseGame()
{
    // Pause a timed event like
    so:
    PwAnalyticsModule.pauseTimedE
vent("My Awesome Game - Level
1");
}

public void resumeGame()
{
    // To end a timed event, pass
    in the same event name like
    so:
    PwAnalyticsModule.resumeTimed
Event("My Awesome Game - Level
1");
}
```

Event Parameters

MaaS Analytics allows you to parameterize all of your events with up to 10 string key / value pairs. *All parameter keys and values must be alphanumeric strings.*



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

    /* other code */

    HashMap<String, String>
params = new HashMap<String,
String>();
    params.put("gender",
"male");

PwAnalyticsModule.addEventWith
Parameters(this, "Featured
Page View", params);

    params = new
HashMap<String, String>();
    params.put("difficulty",
"easy");

PwAnalyticsModule.startTimedEv
entWithParameters(this, "My
Awesome Game - Level 1",
params);
    /* other code */
}

@Override
public void onStop()
{
    super.onStop();
    params = new
HashMap<String, String>();
    params.put("difficulty",
"easy");
    params.put("attempts",
"5");
    // Keep in mind that
calling
endTimedEvent:withParameters
will replace any parameters
that you specified in
startTimedEvent:withParameters
.

PwAnalyticsModule.endTimedEven
tWithParameters(this, "My
Awesome Game - Level 1",
params);
}
```

