

## Spil Android Framework

Generated by Doxygen 1.8.3.1

Wed Aug 14 2013 16:31:11



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>Class Documentation</b>	<b>3</b>
2.1	com.spilgames.framework.listeners.AdsListener Interface Reference . . . . .	3
2.1.1	Member Function Documentation . . . . .	3
2.1.1.1	onAdsFailedToLoad . . . . .	3
2.1.1.2	onAdsLoaded . . . . .	3
2.2	com.spilgames.framework.environment.DevEnvironment Enum Reference . . . . .	3
2.3	com.spilgames.framework.environment.DevStores Enum Reference . . . . .	4
2.4	com.spilgames.framework.SpillInterface Interface Reference . . . . .	4
2.4.1	Member Function Documentation . . . . .	5
2.4.1.1	setAdsListener . . . . .	5
2.4.1.2	showInterstitial . . . . .	5
2.4.1.3	trackEndTimedEvent . . . . .	5
2.4.1.4	trackEndTimedEvent . . . . .	5
2.4.1.5	trackError . . . . .	6
2.4.1.6	trackEvent . . . . .	6
2.4.1.7	trackEvent . . . . .	6
2.4.1.8	trackEvent . . . . .	6
2.4.1.9	trackEvent . . . . .	7
2.4.1.10	trackPage . . . . .	7
2.4.1.11	trackTimedEvent . . . . .	7
2.4.1.12	trackUserId . . . . .	7
	<b>Index</b>	<b>7</b>



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">com.spilgames.framework.listeners.AdsListener</a>	3
<a href="#">com.spilgames.framework.environment.DevEnvironment</a>	3
<a href="#">com.spilgames.framework.environment.DevStores</a>	4
<a href="#">com.spilgames.framework.SpillInterface</a>	4



## Chapter 2

# Class Documentation

## 2.1 com.spilgames.framework.listeners.AdsListener Interface Reference

### Public Member Functions

- void [onAdsLoaded](#) ()
- void [onAdsFailedToLoad](#) (String *cause*)

### 2.1.1 Member Function Documentation

#### 2.1.1.1 void com.spilgames.framework.listeners.AdsListener.onAdsFailedToLoad ( String *cause* )

Method to call back after if the ad subsystem couldn't be started due to any reason.

#### Parameters

<i>cause</i>	The reason why the ad subsystem failed to start.
--------------	--

#### 2.1.1.2 void com.spilgames.framework.listeners.AdsListener.onAdsLoaded ( )

Method to call back after the ad subsystem is successfully started.

The documentation for this interface was generated from the following file:

- /home/jose/Projects/Android-Front-end/src/SpilFramework/src/com/spilgames/framework/listeners/Ads-Listener.java

## 2.2 com.spilgames.framework.environment.DevEnvironment Enum Reference

### Public Member Functions

- **DevEnvironment** (String value)
- String **getValue** ()

### Static Public Member Functions

- static [DevEnvironment](#) **toValue** (String toValue)

### Public Attributes

- **SG\_ENVIRONMENT\_DEV\_VALUE** = ("SG\_ENVIRONMENT\_DEV\_VALUE")
- **SG\_ENVIRONMENT\_LIVE\_VALUE** = ("SG\_ENVIRONMENT\_LIVE\_VALUE")
- **SG\_ENVIRONMENT\_DEBUG\_VALUE** = ("SG\_ENVIRONMENT\_DEBUG\_VALUE")
- **SG\_ENVIRONMENT\_STG\_VALUE** = ("SG\_ENVIRONMENT\_STG\_VALUE")

### Static Public Attributes

- staticfinal String **SG\_ENVIRONMENT\_KEY** = "SG\_ENVIRONMENT\_KEY"

The documentation for this enum was generated from the following file:

- /home/jose/Projects/Android-Front-end/src/SpilFramework/src/com/spilgames/framework/environment/Dev-Environment.java

## 2.3 com.spilgames.framework.environment.DevStores Enum Reference

### Public Member Functions

- **DevStores** (String value, String id)
- String **getValue** ()
- String **getId** ()

### Static Public Member Functions

- static [DevStores](#) **toValue** (String toValue)

### Public Attributes

- **SG\_STORE\_IOS** = ("SG\_STORE\_IOS","1")
- **SG\_STORE\_AMAZON** = ("SG\_STORE\_AMAZON","2")
- **SG\_STORE\_GOOGLE\_PLAY** = ("SG\_STORE\_GOOGLE\_PLAY","3")

### Static Public Attributes

- staticfinal String **SG\_STORE\_ID** = "SG\_STORE\_ID"

The documentation for this enum was generated from the following file:

- /home/jose/Projects/Android-Front-end/src/SpilFramework/src/com/spilgames/framework/environment/Dev-Stores.java

## 2.4 com.spilgames.framework.SpilInterface Interface Reference

### Public Member Functions

- void [trackPage](#) (String page)
- void [trackEvent](#) (String event)
- void [trackEvent](#) (String category, String action, String label, long value)



- void [trackEvent](#) (String category, String action, String label, long value, Map< String, String > params)
- void [trackEvent](#) (String event, Map< String, String > params)
- void [trackTimedEvent](#) (String event)
- void [trackEndTimedEvent](#) (String event)
- void [trackEndTimedEvent](#) (String event, Map< String, String > params)
- void [trackError](#) (String event, String msg, String exception)
- void [trackUserId](#) (String userId)
- void [showInterstitial](#) ()
- void [setAdsListener](#) ([AdsListener](#) adsListener)
- [SpillInterface](#) [start](#) (Context context, String applicationID, String authToken, Map< String, String > configs)
- void [onStartTracking](#) (final Activity context)
- void [onStopTracking](#) (Activity context)
- void [onAdsStart](#) (Activity activity)
- void [onAdsDestroy](#) (Activity activity)
- void [onAdsStop](#) (Activity activity)

### 2.4.1 Member Function Documentation

#### 2.4.1.1 void com.spilgames.framework.SpillInterface.setAdsListener ( [AdsListener](#) *adsListener* )

Method to set the AdsListener and receive the proper notifications from it.

##### Parameters

<i>adListener</i>	The listener to handle the events generated by the Ads subsystem.
-------------------	---

#### 2.4.1.2 void com.spilgames.framework.SpillInterface.showInterstitial ( )

The ads are displayed based on a timer, this method force the ad to be shown right away, and the timer is reset.

#### 2.4.1.3 void com.spilgames.framework.SpillInterface.trackEndTimedEvent ( [String](#) *event* )

Track requesto to register the end of an event that was started. If the tracking system support it natively the equivalent method will be used. Otherwise, an event with the end timestamp(epoch) is issued.

##### Parameters

<i>event</i>	The event to track. Should match with the starting event.
--------------	---

##### See Also

[trackTimedEvent](#):

#### 2.4.1.4 void com.spilgames.framework.SpillInterface.trackEndTimedEvent ( [String](#) *event*, Map< [String](#), [String](#) > *params* )

Track requesto to register the end of an event that was started. If the tracking system support it natively the equivalent method will be used. Otherwise, an event with the end timestamp(epoch) is issued.

##### Parameters

<i>event</i>	The event to track. Should match with the starting event.
<i>params</i>	The parameters when the event was finished.

## See Also

[trackTimedEvent](#):

#### 2.4.1.5 void com.spilgames.framework.SpilInterface.trackError ( String *event*, String *msg*, String *exception* )

Track request to register an error that has occurred in the application.

## Parameters

<i>event</i>	The event to track.
<i>msg</i>	The message with the detail of the error.
<i>exception</i>	The exception that causes the error.

#### 2.4.1.6 void com.spilgames.framework.SpilInterface.trackEvent ( String *event* )

Track request to register an event. The event could be actions taken on some object like unlocking an achievement, or a getting a hi score. In google analytics jargon they will be register as a "category"="event" and the event passed as parameter as the action taken. In flurry jargon, an event with parameters is created and the parameters match with google analytics (category,action,label,value)

## Parameters

<i>event</i>	The event to track.
--------------	---------------------

#### 2.4.1.7 void com.spilgames.framework.SpilInterface.trackEvent ( String *category*, String *action*, String *label*, long *value* )

Track request to register an event under a particular category. The parameters match with the google analytics' ones. For flurry, a event with parameters is issued.

## Parameters

<i>category</i>	The category for this event.
<i>action</i>	The action took on that category.
<i>label</i>	Optional. A string label to specify something about the action.
<i>value</i>	Optional. A integer value. useful to specify error codes.

#### 2.4.1.8 void com.spilgames.framework.SpilInterface.trackEvent ( String *category*, String *action*, String *label*, long *value*, Map< String, String > *params* )

Track request to register an event under a particular category. The parameters match with the google analytics' ones. For flurry, a event with parameters is issued.

## Parameters

<i>category</i>	The category for this event.
<i>action</i>	The action took on that category.
<i>label</i>	Optional. A string label to specify something about the action.
<i>value</i>	Optional. A integer value. useful to specify error codes.
<i>params</i>	Optional. Additional parameters to attach to the event.

**2.4.1.9 void com.spilgames.framework.SpillInterface.trackEvent ( String *event*, Map< String, String > *params* )**

Track request to register an event with particular parameters. This match the Flurry's logEvent:withParameters:. It's not supported for GAN.

**Parameters**

<i>event</i>	The event to track
<i>params</i>	Additional parameters to attach to the event.

**2.4.1.10 void com.spilgames.framework.SpillInterface.trackPage ( String *page* )**

Track request to register a particular page. It can be used to keep track of the current screen separately from the events. If the session is not started yet, this request is ignored.

**Parameters**

<i>page</i>	The page name/url to track.
-------------	-----------------------------

**2.4.1.11 void com.spilgames.framework.SpillInterface.trackTimedEvent ( String *event* )**

Track request to register the start of an event that should end in a finite time. If the tracking system support it natively the equivalent method will be used. Otherwise, an event with the start timestamp(epoch) is issued.

**Parameters**

<i>event</i>	The event to track.
--------------	---------------------

**2.4.1.12 void com.spilgames.framework.SpillInterface.trackUserId ( String *userId* )**

Track the user ID that logged in the application. Useful to keep track of how many users return to the application.

**Parameters**

<i>userId</i>	The user ID to track.
---------------	-----------------------

The documentation for this interface was generated from the following file:

- /home/jose/Projects/Android-Front-end/src/SpilFramework/src/com/spilgames/framework/SpillInterface.java

# Index

com.spilgames.framework.environment.DevEnvironment,  
3

com.spilgames.framework.environment.DevStores, 4

com.spilgames.framework.listeners.AdsListener, 3

com.spilgames.framework.SpillInterface, 4

com::spilgames::framework::SpillInterface

    setAdsListener, 5

    showInterstitial, 5

    trackEndTimedEvent, 5

    trackError, 6

    trackEvent, 6

    trackPage, 7

    trackTimedEvent, 7

    trackUserId, 7

com::spilgames::framework::listeners::AdsListener

    onAdsFailedToLoad, 3

    onAdsLoaded, 3

onAdsFailedToLoad

    com::spilgames::framework::listeners::AdsListener,  
3

onAdsLoaded

    com::spilgames::framework::listeners::AdsListener,  
3

setAdsListener

    com::spilgames::framework::SpillInterface, 5

showInterstitial

    com::spilgames::framework::SpillInterface, 5

trackEndTimedEvent

    com::spilgames::framework::SpillInterface, 5

trackError

    com::spilgames::framework::SpillInterface, 6

trackEvent

    com::spilgames::framework::SpillInterface, 6

trackPage

    com::spilgames::framework::SpillInterface, 7

trackTimedEvent

    com::spilgames::framework::SpillInterface, 7

trackUserId

    com::spilgames::framework::SpillInterface, 7