

# Spil iOS Framework

Generated by Doxygen 1.8.1.2

Wed Aug 21 2013 10:30:12



# Contents

<b>1</b>	<b>Deprecated List</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	<ABTestDelegate> Protocol Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Function Documentation . . . . .	5
3.1.2.1	abtestSessionDidEnd . . . . .	5
3.1.2.2	abtestSessionDidStart . . . . .	5
3.1.2.3	abtestSessionDiffReceived: . . . . .	5
3.2	<AppSettingsDelegate> Protocol Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.2.2	Member Function Documentation . . . . .	6
3.2.2.1	appSettingsDidFailWithError: . . . . .	6
3.2.2.2	appSettingsDidLoad: . . . . .	6
3.2.2.3	appSettingsDidStartDownload . . . . .	7
3.3	Spil Class Reference . . . . .	7
3.3.1	Detailed Description . . . . .	8
3.3.2	Member Function Documentation . . . . .	8
3.3.2.1	abtestGetTestDiff . . . . .	8
3.3.2.2	abtestGetTestDiffForUser: . . . . .	8
3.3.2.3	abtestMarkSucceedTest:withParameters: . . . . .	8
3.3.2.4	abtestUpdateUserInfo . . . . .	8
3.3.2.5	abtestUpdateUserInfoWith: . . . . .	8
3.3.2.6	adsCacheNextInterstitial . . . . .	9
3.3.2.7	adsEnabled: . . . . .	9
3.3.2.8	adsNextInterstitial . . . . .	9

3.3.2.9	<a href="#">adsPlaceAdOn:at:</a>	9
3.3.2.10	<a href="#">adsRemovePlacedAds</a>	9
3.3.2.11	<a href="#">adsShowMoreGames</a>	9
3.3.2.12	<a href="#">getABTest:</a>	9
3.3.2.13	<a href="#">getAds:</a>	10
3.3.2.14	<a href="#">getExtendedTracking:</a>	10
3.3.2.15	<a href="#">getSettings:</a>	10
3.3.2.16	<a href="#">sharedInstance</a>	10
3.3.2.17	<a href="#">spilWithAppID:token:configs:</a>	10
3.3.2.18	<a href="#">trackAge:</a>	11
3.3.2.19	<a href="#">trackEndTimedEvent:</a>	11
3.3.2.20	<a href="#">trackEndTimedEvent:withParams:</a>	11
3.3.2.21	<a href="#">trackError:message:exception:</a>	11
3.3.2.22	<a href="#">trackEvent:</a>	12
3.3.2.23	<a href="#">trackEvent:action:label:value:</a>	12
3.3.2.24	<a href="#">trackEvent:withParams:</a>	12
3.3.2.25	<a href="#">trackGender:</a>	12
3.3.2.26	<a href="#">trackLatitude:longitude:horizontalAccuracy:verticalAccuracy:</a>	12
3.3.2.27	<a href="#">trackPage:</a>	13
3.3.2.28	<a href="#">trackStartGestureScreen:</a>	13
3.3.2.29	<a href="#">trackStopGestureScreen</a>	13
3.3.2.30	<a href="#">trackTimedEvent:</a>	13
3.3.2.31	<a href="#">trackUserID:</a>	13
3.4	<a href="#">SpilHelpers Class Reference</a>	14
3.4.1	<a href="#">Detailed Description</a>	14
3.4.2	<a href="#">Member Function Documentation</a>	14
3.4.2.1	<a href="#">connectionWithURL:onSuccess:onFailure:</a>	14
3.4.2.2	<a href="#">getAppName</a>	14
3.4.2.3	<a href="#">getAppVersion</a>	14
3.4.2.4	<a href="#">getCountryCode</a>	15
3.4.2.5	<a href="#">getDeviceModel</a>	15
3.4.2.6	<a href="#">getDevicePlatform</a>	15
3.4.2.7	<a href="#">getLanguage</a>	15
3.4.2.8	<a href="#">getMacAddress</a>	15
3.4.2.9	<a href="#">getOSVersion</a>	16
3.4.2.10	<a href="#">getPixelRatio</a>	16
3.4.2.11	<a href="#">getScreenHeight</a>	16

3.4.2.12	getScreenWidth	16
3.4.2.13	getUDID	16
3.5	<SpilTracker> Protocol Reference	16
3.5.1	Member Function Documentation	17
3.5.1.1	trackAge:	17
3.5.1.2	trackEndTimedEvent:params:	17
3.5.1.3	trackError:message:exception:	17
3.5.1.4	trackEvent:action:label:value:withParameters:	17
3.5.1.5	trackEvent:params:	18
3.5.1.6	trackGender:	18
3.5.1.7	trackLatitude:longitude:horizontalAccuracy:verticalAccuracy:	18
3.5.1.8	trackPage:	18
3.5.1.9	trackTimedEvent:	19
3.5.1.10	trackUserID:	19
3.6	<TrackingExtendedDelegate> Protocol Reference	19
3.6.1	Detailed Description	19
3.6.2	Member Function Documentation	19
3.6.2.1	trackExtendedDidStart	19
3.6.2.2	trackExtendedDidStop	19



## Chapter 1

# Deprecated List

### Member [\[SpilHelpers getMacAddress\]](#)

Gets the MAC address of this device. The address is returned as a upper case hex string without spaces or colons





## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">&lt;ABTestDelegate&gt;</a>	5
<a href="#">&lt;AppSettingsDelegate&gt;</a>	6
<a href="#">Spil</a>	7
<a href="#">SpilHelpers</a>	14
<a href="#">&lt;SpilTracker&gt;</a>	16
<a href="#">&lt;TrackingExtendedDelegate&gt;</a>	19



## Chapter 3

# Class Documentation

### 3.1 <ABTestDelegate> Protocol Reference

```
#import <ABTestDelegate.h>
```

#### Public Member Functions

- (void) - [abtestSessionDidStart](#)
- (void) - [abtestSessionDidEnd](#)
- (void) - [abtestSessionDiffReceived:](#)

#### 3.1.1 Detailed Description

Protocol to listen the events triggered by the A/B Testing subsystem

#### 3.1.2 Member Function Documentation

##### 3.1.2.1 - (void) abtestSessionDidEnd

Method to call back after the a/b test subsystem is successfully ended.

##### 3.1.2.2 - (void) abtestSessionDidStart

Method to call back after the a/b test subsystem is successfully started.

##### 3.1.2.3 - (void) abtestSessionDiffReceived: (NSArray \*) *diffs*

Method to call back after the a/b test subsystem receive the differences to apply over the original version. The differences come expressed as an array of objects. These objects are represented as dictionaries, where, always are defined the following keys:

- **uid**: an ID for this resource to test. A resource can contain many elements to test. Details in the next entry.

- **diff**: a dictionary with all the changes to apply to this resource. In this resource, many elements could be changed, for each element, an entry will appear in this dictionary. Each of this entry will contain a dictionary with exactly 2 keys: "new" and "old", referring to the original and value to replace with.
- **item\_class**: unused for the moment.

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

- /Users/ignacio/Documents/native-unity-plugin/libs/ios-api/src/Classes/abtest/ABTestDelegate.h

## 3.2 <AppSettingsDelegate> Protocol Reference

```
#import <AppSettingsDelegate.h>
```

### Public Member Functions

- (void) - [appSettingsDidLoad](#):
- (void) - [appSettingsDidFailWithError](#):
- (void) - [appSettingsDidStartDownload](#)

### 3.2.1 Detailed Description

Protocol to handle the responses from the App Settings subsystem

### 3.2.2 Member Function Documentation

#### 3.2.2.1 - (void) appSettingsDidFailWithError: (NSError \*) *error* [required]

Method to call back in case the settings couldn't be loaded. Usually the reasons to call this method will be:

- if there is any parsing error in the remote settings and in the local settings.
- if there is a connection error, and the file of the defaults can be found locally.

#### Parameters

<i>error</i>	Error describing what was wrong.
--------------	----------------------------------

#### 3.2.2.2 - (void) appSettingsDidLoad: (NSDictionary \*) *settings* [required]

Method to call back when the settings are finally loaded. This methods will receive the settings loaded in the form of a dictionary. The developers should know the structure of the dictionary since they created the default settings file.

#### Parameters

<i>settings</i>	The settings loaded. The format and the values are defined by the developer of the app.
-----------------	---

### 3.2.2.3 - (void) appSettingsDidStartDownload [required]

Method to call back when the download of the settings has been started.

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

- /Users/ignacio/Documents/native-unity-plugin/libs/ios-api/src/Classes/settings/AppSettingsDelegate.h

## 3.3 Spil Class Reference

```
#import <Spil.h>
```

### Public Member Functions

- (void) - [getSettings:](#)
- (void) - [getAds:](#)
- (void) - [adsEnabled:](#)
- (void) - [adsShowMoreGames](#)
- (void) - [adsNextInterstitial](#)
- (void) - [adsCacheNextInterstitial](#)
- (BOOL) - [adsPlaceAdOn:at:](#)
- (void) - [adsRemovePlacedAds](#)
- (void) - [getExtendedTracking:](#)
- (void) - [trackPage:](#)
- (void) - [trackEvent:](#)
- (void) - [trackEvent:action:label:value:](#)
- (void) - [trackEvent:withParams:](#)
- (void) - [trackTimedEvent:](#)
- (void) - [trackEndTimedEvent:withParams:](#)
- (void) - [trackEndTimedEvent:](#)
- (void) - [trackError:message:exception:](#)
- (void) - [trackUserID:](#)
- (void) - [trackAge:](#)
- (void) - [trackGender:](#)
- (void) - [trackLatitude:longitude:horizontalAccuracy:verticalAccuracy:](#)
- (void) - [trackStartGestureScreen:](#)
- (void) - [trackStopGestureScreen](#)
- (void) - [getABTest:](#)
- (void) - [abtestUpdateUserInfo](#)
- (void) - [abtestUpdateUserInfoWith:](#)
- (void) - [abtestGetTestDiff](#)
- (void) - [abtestGetTestDiffForUser:](#)
- (void) - [abtestMarkSucceedTest:withParameters:](#)

### Static Public Member Functions

- ([Spil \\*](#)) + [spilWithAppID:token:configs:](#)
- ([Spil \\*](#)) + [sharedInstance](#)

### 3.3.1 Detailed Description

Public interface to the functionalities of the [Spil](#) iOS framework.

### 3.3.2 Member Function Documentation

#### 3.3.2.1 - (void) abtestGetTestDiff

Send a request to retrieve the test differences for this user (MAC Address). The differences will be sent asynchronously to the [ABTestDelegate](#) implemented and set in the getABTest method.

#### 3.3.2.2 - (void) abtestGetTestDiffForUser: (NSString \*) *user*

For development purposes only. Send a request to retrieve the test differences for this user. The differences will be sent asynchronously to the [ABTestDelegate](#) implemented and set in the getABTest method.

##### Parameters

<i>user</i>	The user to force the different variants of the A/B test.
-------------	---

#### 3.3.2.3 - (void) abtestMarkSucceedTest: (NSString \*) *name* withParameters:(NSDictionary \*) *params*

Mark a particular resource as a success with the parameters that lead to that success. This method should be called with the exact name of the resource and also it must be called with the control version to be able to compare results.

##### Parameters

<i>name</i>	The name of the resource to mark as a successful one.
<i>params</i>	A dictionary with extra parameters relevant for the analysis of the action called.

#### 3.3.2.4 - (void) abtestUpdateUserInfo

Update the user basic information to create segments and improve A/B tests. This method will send:

- Country
- Language
- Device (ipad, ipod, iphone)
- OS Platform (version)

#### 3.3.2.5 - (void) abtestUpdateUserInfoWith: (NSDictionary \*) *extraInfo*

Update the user information to create segments and improve A/B tests. This method will send the basic information plus all the information included in the extra info

##### Parameters

<i>extraInfo</i>	A dictionary with all the extra parameter we want to submit.
------------------	--

## See Also

- [abtestUpdateUserInfo](#)

**3.3.2.6 - (void) adsCacheNextInterstitial**

Cache the next interstitial image to speed up the load time.

**3.3.2.7 - (void) adsEnabled: (BOOL) state**

Turn on/off if the ads should be displayed. The ads are displayed by default. For gameplay screens should be turned off. After return to the menus should be turned on again.

## Parameters

<i>state</i>	Indicates if the ads should be displayed or not.
--------------	--

**3.3.2.8 - (void) adsNextInterstitial**

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

**3.3.2.9 - (BOOL) adsPlaceAdOn: (UIView \*) container at:(CGRect) area**

Place one of the interstitial assets over the provided container filling the area specified. The area specifies the relative position to the container position and the dimensions of the space to be filled. If the area falls outside the container area, nothing will be shown and a message will be printed on the console. If the container is nil, nothing will be shown and a message will be printed on the console. The area can be equal to the area of the container view. This method returns immediately, although, the image can be applied later due to the asynchronous nature of chartboost.

## Parameters

<i>container</i>	A view that will contain the advert
<i>area</i>	The rectangle where the advert will be shown inside the view.

## Returns

NO if any of the conditions are not met, YES otherwise.

**3.3.2.10 - (void) adsRemovePlacedAds**

Remove the advertisements from their supervIEWS.

**3.3.2.11 - (void) adsShowMoreGames**

Force to show the More Games screen.

**3.3.2.12 - (void) getABTest: (id< ABTestDelegate >) delegate**

Method to set the [ABTestDelegate](#) and receive the proper notifications from it.

## Parameters

<i>delegate</i>	The delegate to handle the events generated by the A/B test subsystem.
-----------------	--

3.3.2.13 - (void) getAds: (id< AdsDelegate >) *delegate*

Method to set the AdsDelegate and receive the proper notifications from it. Also allows to receive the push notifications.

## Parameters

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

3.3.2.14 - (void) getExtendedTracking: (id< TrackingExtendedDelegate >) *delegate*

Method to set up the delegate for the extended tracking events. This delegate is optional, but its usage it's encouraged since this will guarantee the calls made are actually efective and not dropped because the extended tracking wasn't started yet.

3.3.2.15 - (void) getSettings: (id< AppSettingsDelegate >) *delegate*

Method to retrieve the AppSettings for this app. A delegate is required in order to deliver the settings downloaded from the server or loaded from the default files.

## Parameters

<i>delegate</i>	The delegate to handle the response of the AppSettings subsystem.
-----------------	---

## 3.3.2.16 + (Spil\*) sharedInstance

Method to retrieve the sharedInstance, since this class is a singleton. The instance returned could be nil if the constructor above haven't been called or if was error ocured.

## Returns

The shared instance of this [Spil](#) object.

3.3.2.17 + (Spil\*) spilWithAppID: (NSString \*) *applicationID* token:(NSString \*) *authenticationToken* configs:(NSDictionary \*) *configurations*

Create a [Spil](#) object singleton with an application ID and authentication token that will be used along the framework for multiple services. Also you must specify some configurations to control the behavior of the framework. Most important, if the framework should act like a development environment or a production environment. Some validations about the configurations are made, if one fails, a nil object is returned, and the error is written in the console log.

## Parameters

<i>applicationID</i>	The application ID provided by <a href="#">Spil</a> Games, it can't be nil.
<i>authenticationToken</i>	The authentication token provided by <a href="#">Spil</a> Games, it can't be nil.
<i>configurations</i>	A dictionary with the posible settings to be used by spil framework.



**Returns**

The [Spil](#) object that will be use for further calls.

**3.3.2.18 - (void) trackAge: (int) age**

Track the age of the user is logged in the application. Useful for demographic information.

**Parameters**

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

**3.3.2.19 - (void) trackEndTimedEvent: (NSString \*) 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:](#)

**3.3.2.20 - (void) trackEndTimedEvent: (NSString \*) event withParams:(NSDictionary \*) 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:](#)

**3.3.2.21 - (void) trackError: (NSString \*) event message:(NSString \*) msg exception:(NSException \*) 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.

**3.3.2.22 - (void) trackEvent: (NSString \*) 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.
--------------	---------------------

**3.3.2.23 - (void) trackEvent: (NSString \*) category action:(NSString \*) action label:(NSString \*) label value:(int) 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.

**3.3.2.24 - (void) trackEvent: (NSString \*) event withParams:(NSDictionary \*) 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.

**3.3.2.25 - (void) trackGender: (BOOL) male**

Track the gender of the user is logged in the application. Useful for demographic information.

**Parameters**

<i>male</i>	YES TRUE if the player is male, NO FALSE if the player is female.
-------------	---

**3.3.2.26 - (void) trackLatitude: (CLLocationDegrees) latitude longitude:(CLLocationDegrees) longitude horizontalAccuracy:(CLLocationAccuracy) hAccuracy verticalAccuracy:(CLLocationAccuracy) vAccuracy**

Track the location information of the player if it's available. If the tracking system support it natively the equivalent method will be used. Otherwise, an event with the information of the location (latitude, longitude and accuracy) is issued.

## Parameters

<i>latitude</i>	The latitude where the device is. (it's a double value)
<i>longitude</i>	The longitude where the device is. (it's a double value).
<i>hAccuracy</i>	The horizontal accuracy of the measurement. (it's a double value).
<i>vAccuracy</i>	The vertical accuracy of the measurement. (it's a double value).

**3.3.2.27 - (void) trackPage: (NSString \*) 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.
-------------	-----------------------------

**3.3.2.28 - (void) trackStartGestureScreen: (NSString \*) screenName**

Start the recording of the gestures for a new screen, the gestures for this screen will be stored together regarding how many times this screen has been started.

## Parameters

<i>screenName</i>	The name of the screen to record.
-------------------	-----------------------------------

**3.3.2.29 - (void) trackStopGestureScreen**

Stop the recording gestures for this screen, the gestures are drop until you start a new screen.

**3.3.2.30 - (void) trackTimedEvent: (NSString \*) 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.
--------------	---------------------

**3.3.2.31 - (void) trackUserID: (NSString \*) 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 class was generated from the following file:

- /Users/ignacio/Documents/native-unity-plugin/libs/ios-api/src/Classes/Spil.h

## 3.4 SpilHelpers Class Reference

```
#import <SpilHelpers.h>
```

### Static Public Member Functions

- (NSString \*) + [getMacAddress](#)
- (NSString \*) + [getUDID](#)
- (NSString \*) + [getAppVersion](#)
- (NSString \*) + [getAppName](#)
- (NSString \*) + [getLanguage](#)
- (NSString \*) + [getCountryCode](#)
- (NSString \*) + [getDeviceModel](#)
- (NSString \*) + [getDevicePlatform](#)
- (NSString \*) + [getOSVersion](#)
- (int) + [getScreenWidth](#)
- (int) + [getScreenHeight](#)
- (float) + [getPixelRatio](#)
- (void) + [connectionWithURL:onSuccess:onFailure:](#)

### 3.4.1 Detailed Description

Helper class to retrieve information about information relative to the device.

### 3.4.2 Member Function Documentation

#### 3.4.2.1 + (void) connectionWithURL: (NSURL \*) url onSuccess:(NSData \*) *successCallback* onFailure:(NSError \*) *failureCallback*

Simplest method to start a GET connection to the URL provided as parameter. The responses, successful or not, are notified to the respective blocks (if any).

#### Parameters

<i>url</i>	The URL to send the request. This URL can contain extra parameters in the correct query string form.
<i>onSuccess</i>	Block to notify the data retrieved if everything was ok.
<i>onFailure</i>	Block to notify the error.

#### 3.4.2.2 + (NSString\*) getAppName

Gets the current name of the app from the Info.plist

#### Returns

The CFBundleName entry in the Info.plist

#### 3.4.2.3 + (NSString\*) getAppVersion

Gets the current version of the app from the Info.plist

**Returns**

The CFBundleVersion entry in the Info.plist

**3.4.2.4 + (NSString\*) getCountryCode**

Gets the current country configured in the device.

**Returns**

The short country code configured in the device.

**3.4.2.5 + (NSString\*) getDeviceModel**

Gets the device model (ipod, iphone, ipad, simulator, etc)

**Returns**

The device model.

**3.4.2.6 + (NSString\*) getDevicePlatform**

Gets the device platform family (iphone, ipad)

**Returns**

The device family.

**3.4.2.7 + (NSString\*) getLanguage**

Gets the current preferred language for the OS.

**Returns**

The short version of the preferred language.

**3.4.2.8 + (NSString\*) getMacAddress**

**Deprecated** Gets the MAC address of this device. The address is returned as a upper case hex string without spaces or colons

**Returns**

The MAC address of this device.

#### 3.4.2.9 + (NSString\*) getVersion

Gets the OS version installed in the device.

##### Returns

The OS version running in the device.

#### 3.4.2.10 + (float) getPixelRatio

Gets the screen pixel ratio. A ratio 1-to-1 means a traditional screen. A ration 2-to-1 means a retina display.

##### Returns

The screen ratio.

#### 3.4.2.11 + (int) getScreenHeight

Gets the screen height on points.

##### Returns

The screen height on points.

#### 3.4.2.12 + (int) getScreenWidth

Gets the screen width on points.

##### Returns

The screen width on points.

#### 3.4.2.13 + (NSString\*) getUDID

Gets a new UDID based on OpenUDID implementation

##### Returns

A UDID generated by OpenUDID which will be shared across all apps which implements OpenUDID.

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

- /Users/ignacio/Documents/native-unity-plugin/libs/ios-api/src/Classes/SpilHelpers.h

## 3.5 <SpilTracker> Protocol Reference

### Public Member Functions

- (void) - [trackPage](#):

- (void) - [trackEvent:action:label:value:withParameters:](#)
- (void) - [trackEvent:withParams:](#)
- (void) - [trackTimedEvent:](#)
- (void) - [trackEndTimedEvent:withParams:](#)
- (void) - [trackError:message:exception:](#)
- (void) - [trackUserID:](#)
- (void) - [trackAge:](#)
- (void) - [trackGender:](#)
- (void) - [trackLatitude:longitude:horizontalAccuracy:verticalAccuracy:](#)

### 3.5.1 Member Function Documentation

#### 3.5.1.1 - (void) trackAge: (int) *age*

Track the age of the user is logged in the application. Useful for demographic information.

##### Parameters

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

#### 3.5.1.2 - (void) trackEndTimedEvent: (NSString \*) *event* withParams:(NSDictionary \*) *params*

Track request 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:](#)

#### 3.5.1.3 - (void) trackError: (NSString \*) *event* message:(NSString \*) *msg* exception:(NSException \*) *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.

#### 3.5.1.4 - (void) trackEvent: (NSString \*) *category* action:(NSString \*) *action* label:(NSString \*) *label* value:(int) *value* withParameters:(NSDictionary \*) *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.

## 3.5.1.5 - (void) trackEvent: (NSString \*) event withParams:(NSDictionary \*) 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.

## 3.5.1.6 - (void) trackGender: (BOOL) male

Track the gender of the user is logged in the application. Useful for demographic information.

## Parameters

<i>male</i>	YES TRUE if the player is male, NO FALSE if the player is female.
-------------	---

## 3.5.1.7 - (void) trackLatitude: (CLLocationDegrees) latitude longitude:(CLLocationDegrees) longitude horizontalAccuracy:(CLLocationAccuracy) hAccuracy verticalAccuracy:(CLLocationAccuracy) vAccuracy

Track the location information of the player if it's available. If the tracking system support it natively the equivalent method will be used. Otherwise, an event with the information of the location (latitude, longitude and accuracy) is issued.

## Parameters

<i>latitude</i>	The latitude where the device is. (it's a double value)
<i>longitude</i>	The longitude where the device is. (it's a double value).
<i>hAccuracy</i>	The horizontal accuracy of the measurement. (it's a double value).
<i>vAccuracy</i>	The vertical accuracy of the measurement. (it's a double value).

## 3.5.1.8 - (void) trackPage: (NSString \*) 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.
-------------	-----------------------------



**3.5.1.9 - (void) trackTimedEvent: (NSString \*) event**

Track requesto 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.
--------------	---------------------

**3.5.1.10 - (void) trackUserID: (NSString \*) 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 protocol was generated from the following file:

- /Users/ignacio/Documents/native-unity-plugin/libs/ios-api/src/Classes/tracking/SpilTracker.h

**3.6 <TrackingExtendedDelegate> Protocol Reference**

```
#import <TrackingExtendedDelegate.h>
```

**Public Member Functions**

- (void) - [trackExtendedDidStart](#)
- (void) - [trackExtendedDidStop](#)

**3.6.1 Detailed Description**

Protocol to handle the responses from the Extended Tracking subsystem

**3.6.2 Member Function Documentation****3.6.2.1 - (void) trackExtendedDidStart**

Method to call back when the any of the extended trackers are started. If the camera tracker is set up this method is called after the confirmation pop up is done, and if there is at least one tracker active. When the camera tracker is not set up, this method is called when any of the other are activated.

**3.6.2.2 - (void) trackExtendedDidStop**

Method to call back when ALL the extended trackers are stopped. This is an informative call.

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

- /Users/ignacio/Documents/native-unity-plugin/libs/ios-api/src/Classes/tracking/TrackingExtendedDelegate.h

# Index

<ABTestDelegate>, 5  
<AppSettingsDelegate>, 6  
<SpilTracker>, 16  
<TrackingExtendedDelegate>, 19

ABTestDelegate-p  
    abtestSessionDidEnd, 5  
    abtestSessionDidStart, 5  
    abtestSessionDiffReceived:, 5  
abtestGetTestDiff  
    Spil, 8  
abtestGetTestDiffForUser:  
    Spil, 8  
abtestMarkSucceedTest:withParameters:  
    Spil, 8  
abtestSessionDidEnd  
    ABTestDelegate-p, 5  
abtestSessionDidStart  
    ABTestDelegate-p, 5  
abtestSessionDiffReceived:  
    ABTestDelegate-p, 5  
abtestUpdateUserInfo  
    Spil, 8  
abtestUpdateUserInfoWith:  
    Spil, 8  
adsCacheNextInterstitial  
    Spil, 9  
adsEnabled:  
    Spil, 9  
adsNextInterstitial  
    Spil, 9  
adsPlaceAdOn:at:  
    Spil, 9  
adsRemovePlacedAds  
    Spil, 9  
adsShowMoreGames  
    Spil, 9  
AppSettingsDelegate-p  
    appSettingsDidFailWithError:, 6  
    appSettingsDidLoad:, 6  
    appSettingsDidStartDownload, 6  
appSettingsDidFailWithError:  
    AppSettingsDelegate-p, 6  
appSettingsDidLoad:  
    AppSettingsDelegate-p, 6  
appSettingsDidStartDownload

AppSettingsDelegate-p, 6

connectionWithURL:onSuccess:onFailure:  
    SpilHelpers, 14

getABTest:  
    Spil, 9  
getAds:  
    Spil, 10  
getAppName  
    SpilHelpers, 14  
getAppVersion  
    SpilHelpers, 14  
getCountryCode  
    SpilHelpers, 15  
getDeviceModel  
    SpilHelpers, 15  
getDevicePlatform  
    SpilHelpers, 15  
getExtendedTracking:  
    Spil, 10  
getLanguage  
    SpilHelpers, 15  
getMacAddress  
    SpilHelpers, 15  
getOSVersion  
    SpilHelpers, 15  
getPixelRatio  
    SpilHelpers, 16  
getScreenHeight  
    SpilHelpers, 16  
getScreenWidth  
    SpilHelpers, 16  
getSettings:  
    Spil, 10  
getUDID  
    SpilHelpers, 16  
  
sharedInstance  
    Spil, 10  
Spil, 7  
    abtestGetTestDiff, 8  
    abtestGetTestDiffForUser:, 8  
    abtestMarkSucceedTest:withParameters:, 8  
    abtestUpdateUserInfo, 8  
    abtestUpdateUserInfoWith:, 8

- adsCacheNextInterstitial, 9
- adsEnabled:, 9
- adsNextInterstitial, 9
- adsPlaceAdOn:at:, 9
- adsRemovePlacedAds, 9
- adsShowMoreGames, 9
- getABTest:, 9
- getAds:, 10
- getExtendedTracking:, 10
- getSettings:, 10
- sharedInstance, 10
- spilWithAppID:token:configs:, 10
- trackAge:, 11
- trackEndTimedEvent:, 11
- trackEndTimedEvent:withParams:, 11
- trackError:message:exception:, 11
- trackEvent:, 11
- trackEvent:action:label:value:, 12
- trackEvent:withParams:, 12
- trackGender:, 12
- trackLatitude:longitude:horizontalAccuracy:vertical-Accuracy:, 12
- trackPage:, 13
- trackStartGestureScreen:, 13
- trackStopGestureScreen, 13
- trackTimedEvent:, 13
- trackUserID:, 13
- SpilHelpers, 14
  - connectionWithURL:onSuccess:onFailure:, 14
  - getAppName, 14
  - getAppVersion, 14
  - getCountryCode, 15
  - getDeviceModel, 15
  - getDevicePlatform, 15
  - getLanguage, 15
  - getMacAddress, 15
  - getOSVersion, 15
  - getPixelRatio, 16
  - getScreenHeight, 16
  - getScreenWidth, 16
  - getUDID, 16
- SpilTracker-p
  - trackAge:, 17
  - trackEndTimedEvent:withParams:, 17
  - trackError:message:exception:, 17
  - trackEvent:action:label:value:withParameters:, 17
  - trackEvent:withParams:, 18
  - trackGender:, 18
  - trackLatitude:longitude:horizontalAccuracy:vertical-Accuracy:, 18
  - trackPage:, 18
  - trackTimedEvent:, 18
  - trackUserID:, 19
- spilWithAppID:token:configs:
  - Spil, 10
  - trackAge:
    - Spil, 11
    - SpilTracker-p, 17
  - trackEndTimedEvent:
    - Spil, 11
  - trackEndTimedEvent:withParams:
    - Spil, 11
    - SpilTracker-p, 17
  - trackError:message:exception:
    - Spil, 11
    - SpilTracker-p, 17
  - trackEvent:
    - Spil, 11
  - trackEvent:action:label:value:
    - Spil, 12
  - trackEvent:action:label:value:withParameters:
    - SpilTracker-p, 17
  - trackEvent:withParams:
    - Spil, 12
    - SpilTracker-p, 18
  - trackExtendedDidStart
    - TrackingExtendedDelegate-p, 19
  - trackExtendedDidStop
    - TrackingExtendedDelegate-p, 19
  - trackGender:
    - Spil, 12
    - SpilTracker-p, 18
  - trackLatitude:longitude:horizontalAccuracy:vertical-Accuracy:
    - Spil, 12
    - SpilTracker-p, 18
  - trackPage:
    - Spil, 13
    - SpilTracker-p, 18
  - trackStartGestureScreen:
    - Spil, 13
  - trackStopGestureScreen
    - Spil, 13
  - trackTimedEvent:
    - Spil, 13
    - SpilTracker-p, 18
  - trackUserID:
    - Spil, 13
    - SpilTracker-p, 19
  - TrackingExtendedDelegate-p
    - trackExtendedDidStart, 19
    - trackExtendedDidStop, 19