

Package com.selligent.sdk

Interface Summary

Interface	Description
OnSMNotificationMessageRetrieved	This allows to write codes that will be executed after an event is sent to the Selligent platform
SMCallback	This allows to write codes that will be executed after an event is sent to the Selligent platform
SMinAppContentReturn	Interface used to implement the code that will be executed after retrieving the In-App contents
SMinAppMessageDisplay	Implement this interface if you do not want the SDK to display the In-App Message linked to a push notification and do it yourself
SMinAppMessageReturn	Interface used to implement the code that will be executed after retrieving the In-App contents
SMNotificationCallback	Deprecated. <i>Since 1.3, a broadcast is performed at the click on a button (cf.</i>
SMWebViewNavigationOverride	Implement this interface if you want to override the navigation in the WebViews displayed by the In-App messages.

Class Summary

Class	Description
SMAplication	The Application class of the SDK from which the app custom Application should be extended.
SMDeviceInfos	This class is a set of device properties that can be sent to the platform.
SMEvent	Object used to send a custom event to the Selligent platform with SMManager.sendEvent.
SMEventUserLogin	Object used to send a "login" event to the Selligent platform with SMManager.sendEvent.
SMEventUserLogout	Object used to send a "logout" event to the Selligent platform with SMManager.sendEvent.
SMEventUserRegister	Object used to send a "register" event to the Selligent platform with SMManager.sendEvent.
SMEventUserUnregister	Object used to send an "unregister" event to the Selligent platform with

	SMManager.sendEvent.
SMForegroundGcmBroadcastReceiver	Class implementing the receiver that will listen to connectivity changes.
SIMInAppContent	An In App content
SIMInAppContentHtmlFragment	This class implements a fragment that will display one or several HTML contents.
SIMInAppContentImageFragment	This class implements a fragment that will display In App Content containing an image.
SIMInAppContentUrlFragment	This class implements a fragment that will display In App Content containing an url.
SIMInAppMessage	An In App message
SIMLink	A link of an SIMInAppContent .
SIMManager	Singleton object used to interact with the Selligent Mobile SDK.
SIMMapMarker	This class represents a point on a map.
SIMNotificationButton	Object containing all the data used to implement a button in a notification/message.
SIMNotificationMessage	A notification message.
SIMObserverManager	This class is used to observe some events of the SDK.
SIMSettings	Configuration object passed to the 'start' method of SIMManager.

Enum Summary

Enum	Description
SIMClearCache	
SIMContentType	Enum listing the different types of In App Content
SIMInAppContent.DisplayMode	Enum listing the different display mode.
SIMInAppRefreshType	Enum with the different values for the refresh of the In App messages and In App contents.
SIMMessageType	Possible values for an In-app message type
SIMRemoteMessageDisplayType	List the different possibilities to display a remote message received while the app is in foreground - Automatic: the message is automatically displayed - Notification: a notification is created, the user needs to click on it to display the message - None: nothing is done, the message is not displayed, it is up to the app to display it (cf.

SMThemeCategories	Deprecated.
SMWebViewNavigationOption	Enum used by <code>shouldHandleURL(Context, String)</code> to tell the SDK how to handle the navigation: - ResumeNavigation: the navigation will continue in the WebView like usual.

Interface OnSMNotificationMessageRetrieved

```
public interface OnSMNotificationMessageRetrieved
```

This allows to write codes that will be executed after an event is sent to the Selligent platform

Since:

3.8

Version:

3.9

See Also:

`SMNotificationMessage, SMManager.retrieveNotificationMessage(Intent, OnSMNotificationMessageRetrieved)`

Method Summary

All Methods	Instance Methods	Abstract Methods	
Modifier and Type	Method	Description	
void	onError (java.lang.Exception exception)	Event triggered when an error occurred while retrieving the SMNotificationMessage	
void	onSuccess (SMNotificationMessage notificationMessage)	Event triggered when the SMNotificationMessage is retrieved.	

Method Detail

onSuccess

```
void onSuccess(@NotNull
               SMNotificationMessage notificationMessage)
```

Event triggered when the SMNotificationMessage is retrieved.

Parameters:

`notificationMessage` - the String returned by the web service.

onError

```
void onError(java.lang.Exception exception)
```

Event triggered when an error occurred while retrieving the SMNotificationMessage

Parameters:

exception - the Exception thrown by the web service call

Interface SMCallback

```
public interface SMCallback
```

This allows to write codes that will be executed after an event is sent to the Selligent platform

Since:

1.0

Version:

3.9

See Also:

[SMEvent](#), [SMManager.sendEvent\(SMEvent\)](#)

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method	Description
void	onError (int httpResponseCode, java.lang.Exception exception)	Event triggered when an error occured while sending the SMEvent
void	onSuccess (java.lang.String result)	Event triggered when the SMEvent was sent successfully.

Method Detail

onSuccess

```
void onSuccess(java.lang.String result)
```

Event triggered when the SMEvent was sent successfully.

Parameters:

result - the String returned by the web service.

onError

```
void onError(int httpResponseCode,
            java.lang.Exception exception)
```

Event triggered when an error occured while sending the SMEvent

Parameters:

httpResponseCode - the code of the error (404, 500, etc.)

exception - the Exception thrown by the web service call

Interface SMInAppContentReturn

```
public interface SMInAppContentReturn
```

Interface used to implement the code that will be executed after retrieving the In-App contents

Since:

2.1

Version:

3.9

See Also:

SManager, SManager.getInAppContents(String, SMContentType, int)

Method Summary

All Methods Instance Methods Abstract Methods

Modifier and Type	Method	Description
void	onRetrieve (java.util.ArrayList<SMInAppContent> contents)	

Method Detail

onRetrieve

```
void onRetrieve(java.util.ArrayList<SMInAppContent> contents)
```

Interface SMInAppMessageDisplay

```
public interface SMInAppMessageDisplay
```

Implement this interface if you do not want the SDK to display the In-App Message linked to a push notification and do it yourself

Since:

3.6

Version:

3.9

See Also:

`SManager.checkAndDisplayMessage(Intent, Context, SMInAppMessageDisplay)`

Method Summary

All Methods	Instance Methods	Abstract Methods	
Modifier and Type	Method		Description
boolean	<code>onBeforeDisplay (SMInAppMessage message)</code>		This method is called before displaying an In-App Message linked to a push notification.

Method Detail

onBeforeDisplay

`boolean onBeforeDisplay(SMInAppMessage message)`

This method is called before displaying an In-App Message linked to a push notification.

Parameters:

`message` – the `SMInAppMessage`

Returns:

`true` if you want the SDK to display the In-App Message, return `false` if you want to do it yourself.

Interface SMInAppMessageReturn

```
public interface SMInAppMessageReturn
```

Interface used to implement the code that will be executed after retrieving the In-App contents

Since:

3.4

Version:

3.9

See Also:

SManager, SManager.getInAppMessages(SMInAppMessageReturn)

Method Summary

All Methods Instance Methods Abstract Methods

Modifier and Type	Method	Description
void	onRetrieve (java.util.ArrayList<SMInAppMessage> messages)	

Method Detail

onRetrieve

```
void onRetrieve(java.util.ArrayList<SMInAppMessage> messages)
```

Interface SMNotificationCallback

```
@Deprecated  
public interface SMNotificationCallback
```

Deprecated.

Since 1.3, a broadcast is performed at the click on a button (cf. `SMManager.BROADCAST_EVENT_BUTTON_CLICKED`)

Interface used to allow code to be called when clicking on a button received from a push and displayed in an alert dialog, menu of a web view, etc. It will be called no matter the action the button is supposed to perform.

Since:

1.0

Version:

3.9

Method Summary

All Methods	Instance Methods	Abstract Methods	Deprecated Methods
Modifier and Type	Method		Description
void	<code>onButtonClick (java.lang.String buttonId, int buttonAction, java.lang.String buttonValue)</code>		Deprecated.

Method Detail

onButtonClick

```
void onButtonClick(java.lang.String buttonId,  
                  int buttonAction,  
                  java.lang.String buttonValue)
```

Deprecated.

Interface SMWebViewNavigationOverride

```
public interface SMWebViewNavigationOverride
```

Implement this interface if you want to override the navigation in the WebViews displayed by the In-App messages. It will allow you to check the link clicked by the user and decide how the SDK will handle (or not) the navigation.

Since:

3.6.0

Version:

3.9

Method Summary

All Methods	Instance Methods	Abstract Methods	
Modifier and Type	Method	Description	
<code>SMWebViewNavigationOption</code>	<code>shouldHandleURL (android.content.Context context, java.lang.String url)</code>	This method will be called when a link is clicked in a WebView displayed by an In-App message.	

Method Detail

shouldHandleURL

```
SMWebViewNavigationOption shouldHandleURL(android.content.Context context,
                                         java.lang.String url)
```

This method will be called when a link is clicked in a WebView displayed by an In-App message. How the navigation proceeds will depend on the value returned.

Parameters:

`context` - The Context of the call (the Activity containing the WebView)

`url` - The URL String that was clicked

Returns:

the `SMWebViewNavigationOption` indicating how the SDK must handles the navigation

Package com.selligent.sdk

Class SMAApplication

```
java.lang.Object
    android.content.Context
        android.content.ContextWrapper
            android.app.Application
                com.selligent.sdk.SMAApplication
```

All Implemented Interfaces:

```
android.content.ComponentCallbacks, android.content.ComponentCallbacks2
```

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

The Application class of the SDK from which the app custom Application should be extended.

Since:

1.0

Version:

3.9

Nested Class Summary

Nested classes/interfaces inherited from class android.app.Application

```
android.app.Application.ActivityLifecycleCallbacks,
android.app.Application.OnProvideAssistDataListener
```

Field Summary

Fields inherited from class android.content.Context

```
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE,
APP_OPS_SERVICE, APP_SEARCH_SERVICE, APPWIDGET_SERVICE, AUDIO_SERVICE,
BATTERY_SERVICE, BIND_ABOVE_CLIENT, BIND_ADJUST_WITH_ACTIVITY,
BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE, BIND_DEBUG_UNBIND,
BIND_EXTERNAL_SERVICE, BIND_IMPORTANT, BIND_INCLUDE_CAPABILITIES,
BIND_NOT_FOREGROUND, BIND_NOT_PERCEPTIBLE, BIND_WAIVE_PRIORITY, BIOMETRIC_SERVICE,
BLOB_STORE_SERVICE, BLUETOOTH_SERVICE, BUGREPORT_SERVICE, CAMERA_SERVICE,
CAPTIONING_SERVICE, CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE,
COMPANION_DEVICE_SERVICE, CONNECTIVITY_DIAGNOSTICS_SERVICE, CONNECTIVITY_SERVICE,
CONSUMER_IR_SERVICE, CONTEXT_IGNORE_SECURITY, CONTEXT_INCLUDE_CODE,
CONTEXT_RESTRICTED, CROSS_PROFILE_APPS_SERVICE, DEVICE_POLICY_SERVICE,
DISPLAY_HASH_SERVICE, DISPLAY_SERVICE, DOMAIN_VERIFICATION_SERVICE,
DOWNLOAD_SERVICE, DROPBOX_SERVICE, EUICC_SERVICE, FILE_INTEGRITY_SERVICE,
```

```
FINGERPRINT_SERVICE, GAME_SERVICE, HARDWARE_PROPERTIES_SERVICE,  
INPUT_METHOD_SERVICE, INPUT_SERVICE, IPSEC_SERVICE, JOB_SCHEDULER_SERVICE,  
KEYGUARD_SERVICE, LAUNCHER_APPS_SERVICE, LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE,  
MEDIA_COMMUNICATION_SERVICE, MEDIA_METRICS_SERVICE, MEDIA_PROJECTION_SERVICE,  
MEDIA_ROUTER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE, MODE_APPEND,  
MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_NO_LOCALIZED_COLLATORS,  
MODE_PRIVATE, MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, NETWORK_STATS_SERVICE,  
NFC_SERVICE, NOTIFICATION_SERVICE, NSD_SERVICE, PEOPLE_SERVICE,  
PERFORMANCE_HINT_SERVICE, POWER_SERVICE, PRINT_SERVICE,  
RECEIVER_VISIBLE_TO_INSTANT_APPS, RESTRICTIONS_SERVICE, ROLE_SERVICE,  
SEARCH_SERVICE, SENSOR_SERVICE, SHORTCUT_SERVICE, STORAGE_SERVICE,  
STORAGE_STATS_SERVICE, SYSTEM_HEALTH_SERVICE, TELECOM_SERVICE,  
TELEPHONY_IMS_SERVICE, TELEPHONY_SERVICE, TELEPHONY_SUBSCRIPTION_SERVICE,  
TEXT_CLASSIFICATION_SERVICE, TEXT_SERVICES_MANAGER_SERVICE, TV_INPUT_SERVICE,  
UI_MODE_SERVICE, USAGE_STATS_SERVICE, USB_SERVICE, USER_SERVICE,  
VIBRATOR_MANAGER_SERVICE, VIBRATOR_SERVICE, VPN_MANAGEMENT_SERVICE,  
WALLPAPER_SERVICE, WIFI_AWARE_SERVICE, WIFI_P2P_SERVICE, WIFI_RTT_RANGING_SERVICE,  
WIFI_SERVICE, WINDOW_SERVICE
```

Fields inherited from interface android.content.ComponentCallbacks2

```
TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE,  
TRIM_MEMORY_RUNNING_CRITICAL, TRIM_MEMORY_RUNNING_LOW, TRIM_MEMORY_RUNNING_MODERATE,  
TRIM_MEMORY_UI_HIDDEN
```

Constructor Summary

Constructors

Constructor	Description
<code>SMAplication()</code>	

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	<code>onCreate()</code>	

Methods inherited from class android.app.Application

```
getProcessName, onConfigurationChanged, onLowMemory, onTerminate, onTrimMemory,  
registerActivityLifecycleCallbacks, registerComponentCallbacks,  
registerOnProvideAssistDataListener, unregisterActivityLifecycleCallbacks,
```

```
unregisterComponentCallbacks, unregisterOnProvideAssistDataListener
```

Methods inherited from class android.content.ContextWrapper

```
bindIsolatedService, bindService, bindService, bindServiceAsUser,  
checkCallingOrSelfPermission, checkCallingOrSelfPermission,  
checkCallingOrSelfPermission, checkCallingPermission, checkCallingUriPermission,  
checkCallingUriPermissions, checkPermission, checkSelfPermission,  
checkUriPermission, checkUriPermission, checkUriPermissions, clearWallpaper,  
createAttributionContext, createConfigurationContext, createContext,  
createContextForSplit, createDeviceProtectedStorageContext, createDisplayContext,  
createPackageContext, createWindowContext, createWindowContext, databaseList,  
deleteDatabase, deleteFile, deleteSharedPreferences, enforceCallingOrSelfPermission,  
enforceCallingOrSelfPermission, enforceCallingPermission,  
enforceCallingUriPermission, enforcePermission, enforceUriPermission,  
enforceUriPermission, fileList, getApplicationContext, getApplicationInfo,  
getAssets, getAttributionSource, getAttributionTag, getBaseContext, getCacheDir,  
getClassLoader, getCodeCacheDir, getContentResolver, getDatabasePath, getDataDir,  
getDir, getDisplay, getExternalCacheDir, getExternalCacheDirs, getExternalFilesDir,  
getExternalFilesDirs, getExternalMediaDirs, getFilesDir, getFileStreamPath,  
getMainExecutor, getMainLooper, getNoBackupFilesDir, getObbDir, getObbDirs,  
getOpPackageName, getPackageCodePath, getPackageManager, getPackageName,  
getPackageResourcePath, getParams, getResources, getSharedPreferences,  
getSystemService, getSystemServiceName, getTheme, getWallpaper,  
getWallpaperDesiredMinimumHeight, getWallpaperDesiredMinimumWidth,  
grantUriPermission, isDeviceProtectedStorage, isRestricted, isUiContext,  
moveDatabaseFrom, moveSharedPreferencesFrom, openFileInput, openFileOutput,  
openOrCreateDatabase, openOrCreateDatabase, peekWallpaper, registerReceiver,  
registerReceiver, registerReceiver, registerReceiver, removeStickyBroadcast,  
removeStickyBroadcastAsUser, revokeUriPermission, revokeUriPermission,  
sendBroadcast, sendBroadcast, sendBroadcastAsUser, sendBroadcastAsUser,  
sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast,  
sendOrderedBroadcast, sendOrderedBroadcastAsUser, sendStickyBroadcast,  
sendStickyBroadcast, sendStickyBroadcastAsUser, sendStickyOrderedBroadcast,  
sendStickyOrderedBroadcastAsUser, setTheme, setWallpaper, setWallpaper,  
startActivities, startActivities, startActivity, startActivity,  
startForegroundService, startInstrumentation, startIntentSender, startIntentSender,  
startService, stopService, unbindService, unregisterReceiver, updateServiceGroup
```

Methods inherited from class android.content.Context

```
getColor, getColorStateList, getDrawable, getString, getString, getSystemService,  
getText, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes,  
obtainStyledAttributes, sendBroadcastWithMultiplePermissions
```

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

SMAApplication

```
public SMAApplication()
```

Method Detail

onCreate

```
public void onCreate()
```

Overrides:

onCreate in class android.app.Application

Class SMDeviceInfos

java.lang.Object
 com.selligent.sdk.SMDeviceInfos

```
public class SMDeviceInfos
extends java.lang.Object
```

This class is a set of device properties that can be sent to the platform.

Since:

1.6

Version:

3.9

See Also:

`SManager.sendDeviceInfos (SMDeviceInfos)`

Field Summary

Fields

Modifier and Type	Field	Description
java.lang.String	<code>ExternalId</code>	You can set this with your id for the device.

Constructor Summary

Constructors

Constructor	Description
<code>SMDeviceInfos ()</code>	

Method Summary

Methods inherited from class `java.lang.Object`

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

ExternalId

```
public java.lang.String ExternalId
```

You can set this with your id for the device.

Constructor Detail

SMDeviceInfos

```
public SMDeviceInfos()
```

Class SMEVENT

java.lang.Object
 com.selligent.sdk.SMEVENT

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

Direct Known Subclasses:

SMEVENTUserLogin, SMEVENTUserLogout, SMEVENTUserRegister, SMEVENTUserUnregister

```
public class SMEVENT
extends java.lang.Object
implements java.io.Externalizable
```

Object used to send a custom event to the Selligent platform with SMManager.sendEvent.

Since:

1.0

Version:

3.9

See Also:

SMManager.sendSMEVENT(SMEVENT), Serialized Form

Field Summary

Fields

Modifier and Type	Field	Description
SMCallback	Callback	A SMCallback object containing code to execute after the message is sent
java.util.Hashtable<java.lang.String,java.lang.String>	Data	Custom data

Constructor Summary

Constructors

Constructor	Description
SMEVENT()	Constructs an SMEVENT object without data and callback
SMEVENT(java.util.Hashtable<java.lang.String,java.lang.String> data, SMCallback callback)	Constructs an SMEVENT object with the following:

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
boolean	<code>equals (java.lang.Object o)</code>	Compares this instance with the specified object and indicates if they are equal.
int	<code>hashCode ()</code>	Returns an integer hash code for this object.
void	<code>readExternal (java.io.ObjectInput serializedObject)</code>	This method is called when deserializing the object.
void	<code>writeExternal (java.io.ObjectOutput serializedObject)</code>	This method is called when serializing the object.

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, toString, wait, wait, wait`

Field Detail

Data

`public java.util.Hashtable<java.lang.String,java.lang.String> Data`

Custom data

Callback

`public SMCallback Callback`

A SMCallback object containing code to execute after the message is sent

Constructor Detail

SMEvent

`public SMEvent()`

Constructs an SMEEvent object without data and callback

SMEEvent

```
public SMEEvent(java.util.Hashtable<java.lang.String,java.lang.String> data,  
               SMCallback callback)
```

Constructs an SMEEvent object with the following:

Parameters:

data - a Hashtable<String, String> containing the custom data

callback - a SMCallback object containing code to execute after the message is sent

See Also:

[SMCallback](#)

Method Detail

equals

```
public boolean equals(java.lang.Object o)
```

C.compares this instance with the specified object and indicates if they are equal.

Overrides:

equals in class java.lang.Object

Parameters:

o - the object to compare this instance with.

Returns:

true if the specified object is equal to this Object; false otherwise.

hashCode

```
public int hashCode()
```

Returns an integer hash code for this object. By contract, any two objects for which `equals(Object)` returns true must return the same hash code value. This means that subclasses of Object usually override both methods or neither method.

Overrides:

hashCode in class java.lang.Object

Returns:

this object's hash code.

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

This method is called when deserializing the object. It should not be called manually.

Specified by:

readExternal in interface java.io.Externalizable

Parameters:

serializedObject - the ObjectInput object containing all the values of the SMEvent object to recreate.

Throws:

java.io.IOException

java.lang.ClassNotFoundException

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

This method is called when serializing the object. It should not be called manually.

Specified by:

writeExternal in interface java.io.Externalizable

Parameters:

serializedObject - the ObjectOutputStream object used to store the values of the SMEvent object

Throws:

java.io.IOException

Package com.selligent.sdk

Class SMEVENTUserLogin

java.lang.Object
com.selligent.sdk.SMEVENT
com.selligent.sdk.SMEVENTUserLogin

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

```
public class SMEVENTUserLogin
extends SMEVENT
```

Object used to send a "login" event to the Selligent platform with SMManager.sendEvent.

Since:

1.0

Version:

3.9

See Also:

SMManager.sendSMEVENT(SMEVENT), Serialized Form

Field Summary

Fields

Modifier and Type	Field	Description
java.lang.String	Email	

Fields inherited from class com.selligent.sdk.SMEVENT

Callback, Data

Constructor Summary

Constructors

Constructor	Description
SMEVENTUserLogin()	
SMEVENTUserLogin (java.lang.String email, java.util.Hashtable<java.lang.String,java.lang.String> data, SMCallback callback)	Constructs a new SMEVENTUserLogin
SMEVENTUserLogin (java.util.Hashtable<java.lang.String,java.lang.String> data,	Constructs a new

Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method	Description	
boolean	<code>equals (java.lang.Object o)</code>	Compares this instance with the specified object and indicates if they are equal.	
int	<code>hashCode ()</code>	Returns an integer hash code for this object.	
void	<code>readExternal (java.io.ObjectInput serializedObject)</code>	This method is called when deserializing the object.	
void	<code>writeExternal (java.io.ObjectOutput serializedObject)</code>	This method is called when serializing the object.	

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, toString, wait, wait, wait`

Field Detail

Email

`public java.lang.String Email`

Constructor Detail

SMEventUserLogin

`public SMEventUserLogin()`

SMEventUserLogin

`public SMEventUserLogin(java.lang.String email,
java.util.Hashtable<java.lang.String,java.lang.String> data,
SMCallback callback)`

Constructs a new SMEventUserLogin

Parameters:

email - a String containing the e-mail address of the user.

data - a Hashtable<String, String> containing custom data, can be null.

callback - an SMCallback containing code to perform after the message is sent

See Also:

[SMCallback](#)

SMEventUserLogin

```
public SMEventUserLogin(java.util.Hashtable<java.lang.String,java.lang.String> data,  
                      SMCallback callback)
```

Constructs a new SMEventUserLogin

Parameters:

data - a Hashtable<String, String> containing custom data, can be null.

callback - an SMCallback containing code to perform after the message is sent

Since:

3.2

See Also:

[SMCallback](#)

Method Detail

equals

```
public boolean equals(java.lang.Object o)
```

Description copied from class: [SMEvent](#)

Compares this instance with the specified object and indicates if they are equal.

Overrides:

[equals](#) in class [SMEvent](#)

Parameters:

o - the object to compare this instance with.

Returns:

true if the specified object is equal to this Object; false otherwise.

hashCode

```
public int hashCode()
```

Description copied from class: [SMEvent](#)

Returns an integer hash code for this object. By contract, any two objects for which `SMEvent.equals(Object)` returns true must return the same hash code value. This means that subclasses of Object usually override both methods or neither method.

Overrides:

```
hashCode in class SMEvent
```

Returns:

```
this object's hash code.
```

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.langClassNotFoundException
```

Description copied from class: [SMEvent](#)

This method is called when deserializing the object. It should not be called manually.

Specified by:

```
readExternal in interface java.io.Externalizable
```

Overrides:

```
readExternal in class SMEvent
```

Parameters:

`serializedObject` - the `ObjectInput` object containing all the values of the `SMEvent` object to recreate.

Throws:

```
java.io.IOException
```

```
java.langClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

Description copied from class: [SMEvent](#)

This method is called when serializing the object. It should not be called manually.

Specified by:

```
writeExternal in interface java.io.Externalizable
```

Overrides:

```
writeExternal in class SMEvent
```

Parameters:

serializedObject - the ObjectOutputStream object used to store the values of the SMEvent object

Throws:

java.io.IOException

Package com.selligent.sdk

Class SMEVENTUserLogout

java.lang.Object
com.selligent.sdk.SMEVENT
com.selligent.sdk.SMEVENTUserLogout

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

```
public class SMEVENTUserLogout
extends SMEVENT
```

Object used to send a "logout" event to the Selligent platform with SMManager.sendEvent.

Since:

1.0

Version:

3.9

See Also:

SMManager.sendSMEVENT(SMEVENT), Serialized Form

Field Summary

Fields

Modifier and Type	Field	Description
java.lang.String	Email	

Fields inherited from class com.selligent.sdk.SMEVENT

Callback, Data

Constructor Summary

Constructors

Constructor	Description
SMEVENTUserLogout()	
SMEVENTUserLogout (java.lang.String email, java.util.Hashtable<java.lang.String,java.lang.String> data, SMCallback callback)	Constructs a new SMEVENTUserLogout
SMEVENTUserLogout (java.util.Hashtable<java.lang.String,java.lang.String> data,	Constructs a new

Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method	Description	
boolean	<code>equals (java.lang.Object o)</code>	Compares this instance with the specified object and indicates if they are equal.	
int	<code>hashCode ()</code>	Returns an integer hash code for this object.	
void	<code>readExternal (java.io.ObjectInput serializedObject)</code>	This method is called when deserializing the object.	
void	<code>writeExternal (java.io.ObjectOutput serializedObject)</code>	This method is called when serializing the object.	

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, toString, wait, wait, wait`

Field Detail

Email

`public java.lang.String Email`

Constructor Detail

SMEventUserLogout

`public SMEventUserLogout()`

SMEventUserLogout

`public SMEventUserLogout(java.lang.String email,
java.util.Hashtable<java.lang.String,java.lang.String> data,
SMCallback callback)`

Constructs a new SMEventUserLogout

Parameters:

email - a String containing the e-mail address of the user.

data - a Hashtable<String, String> containing custom data, can be null.

callback - an SMCallback containing code to perform after the message is sent

See Also:

[SMCallback](#)

SMEventUserLogout

```
public SMEventUserLogout(java.util.Hashtable<java.lang.String,java.lang.String> data,  
                        SMCallback callback)
```

Constructs a new SMEventUserLogout

Parameters:

data - a Hashtable<String, String> containing custom data, can be null.

callback - an SMCallback containing code to perform after the message is sent

Since:

3.2

See Also:

[SMCallback](#)

Method Detail

equals

```
public boolean equals(java.lang.Object o)
```

Description copied from class: [SMEvent](#)

Compares this instance with the specified object and indicates if they are equal.

Overrides:

[equals](#) in class [SMEvent](#)

Parameters:

o - the object to compare this instance with.

Returns:

true if the specified object is equal to this Object; false otherwise.

hashCode

```
public int hashCode()
```

Description copied from class: [SMEvent](#)

Returns an integer hash code for this object. By contract, any two objects for which `SMEvent.equals(Object)` returns true must return the same hash code value. This means that subclasses of Object usually override both methods or neither method.

Overrides:

```
hashCode in class SMEvent
```

Returns:

```
this object's hash code.
```

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.langClassNotFoundException
```

Description copied from class: [SMEvent](#)

This method is called when deserializing the object. It should not be called manually.

Specified by:

```
readExternal in interface java.io.Externalizable
```

Overrides:

```
readExternal in class SMEvent
```

Parameters:

`serializedObject` - the `ObjectInput` object containing all the values of the `SMEvent` object to recreate.

Throws:

```
java.io.IOException
```

```
java.langClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

Description copied from class: [SMEvent](#)

This method is called when serializing the object. It should not be called manually.

Specified by:

```
writeExternal in interface java.io.Externalizable
```

Overrides:

```
writeExternal in class SMEvent
```

Parameters:

serializedObject - the ObjectOutputStream object used to store the values of the SMEvent object

Throws:

java.io.IOException

Package com.selligent.sdk

Class SMEVENTUSERREGISTER

java.lang.Object
com.selligent.sdk.SMEVENT
com.selligent.sdk.SMEVENTUSERREGISTER

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

```
public class SMEVENTUSERREGISTER
extends SMEVENT
```

Object used to send a "register" event to the Selligent platform with SMManager.sendEvent.

Since:

1.0

Version:

3.9

See Also:

SMManager.sendSMEVENT(SMEVENT), Serialized Form

Field Summary

Fields

Modifier and Type	Field	Description
java.lang.String	Email	

Fields inherited from class com.selligent.sdk.SMEVENT

Callback, Data

Constructor Summary

Constructors

Constructor	Description
SMEVENTUSERREGISTER()	
SMEVENTUSERREGISTER (java.lang.String email, java.util.Hashtable<java.lang.String,java.lang.String> data, SMCallback callback)	Constructs a new SMEVENTUSERREGISTER
SMEVENTUSERREGISTER (java.util.Hashtable<java.lang.String,java.lang.String> data,	Constructs a new

Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method	Description	
boolean	<code>equals (java.lang.Object o)</code>	Compares this instance with the specified object and indicates if they are equal.	
int	<code>hashCode ()</code>	Returns an integer hash code for this object.	
void	<code>readExternal (java.io.ObjectInput serializedObject)</code>	This method is called when deserializing the object.	
void	<code>writeExternal (java.io.ObjectOutput serializedObject)</code>	This method is called when serializing the object.	

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, toString, wait, wait, wait`

Field Detail

Email

`public java.lang.String Email`

Constructor Detail

SMEventUserRegister

`public SMEventUserRegister()`

SMEventUserRegister

`public SMEventUserRegister(java.lang.String email,
java.util.Hashtable<java.lang.String,java.lang.String> data,
SMCallback callback)`

Constructs a new SMEventUserRegister

Parameters:

email - a String containing the e-mail address of the user.

data - a Hashtable<String, String> containing custom data, can be null.

callback - an SMCallback containing code to perform after the message is sent

See Also:

[SMCallback](#)

SMEventUserRegister

```
public SMEventUserRegister(java.util.Hashtable<java.lang.String,java.lang.String> data,  
                           SMCallback callback)
```

Constructs a new SMEventUserRegister

Parameters:

data - a Hashtable<String, String> containing custom data, can be null.

callback - an SMCallback containing code to perform after the message is sent

Since:

3.2

See Also:

[SMCallback](#)

Method Detail

equals

```
public boolean equals(java.lang.Object o)
```

Description copied from class: [SMEvent](#)

Compares this instance with the specified object and indicates if they are equal.

Overrides:

[equals](#) in class [SMEvent](#)

Parameters:

o - the object to compare this instance with.

Returns:

true if the specified object is equal to this Object; false otherwise.

hashCode

```
public int hashCode()
```

Description copied from class: [SMEvent](#)

Returns an integer hash code for this object. By contract, any two objects for which `SMEvent.equals(Object)` returns true must return the same hash code value. This means that subclasses of Object usually override both methods or neither method.

Overrides:

```
hashCode in class SMEvent
```

Returns:

```
this object's hash code.
```

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.langClassNotFoundException
```

Description copied from class: [SMEvent](#)

This method is called when deserializing the object. It should not be called manually.

Specified by:

```
readExternal in interface java.io.Externalizable
```

Overrides:

```
readExternal in class SMEvent
```

Parameters:

`serializedObject` - the `ObjectInput` object containing all the values of the `SMEvent` object to recreate.

Throws:

```
java.io.IOException
```

```
java.langClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

Description copied from class: [SMEvent](#)

This method is called when serializing the object. It should not be called manually.

Specified by:

```
writeExternal in interface java.io.Externalizable
```

Overrides:

```
writeExternal in class SMEvent
```

Parameters:

serializedObject - the ObjectOutputStream object used to store the values of the SMEvent object

Throws:

java.io.IOException

Package com.selligent.sdk

Class SMEVENTUSERUNREGISTER

java.lang.Object
com.selligent.sdk.SMEVENT
com.selligent.sdk.SMEVENTUSERUNREGISTER

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

```
public class SMEVENTUSERUNREGISTER
extends SMEVENT
```

Object used to send an "unregister" event to the Selligent platform with SMManager.sendEvent.

Since:

1.0

Version:

3.9

See Also:

SMManager.sendSMEVENT(SMEVENT), Serialized Form

Field Summary

Fields

Modifier and Type	Field	Description
java.lang.String	Email	

Fields inherited from class com.selligent.sdk.SMEVENT

Callback, Data

Constructor Summary

Constructors

Constructor	Description
SMEVENTUSERUNREGISTER()	
SMEVENTUSERUNREGISTER (java.lang.String email, java.util.Hashtable<java.lang.String,java.lang.String> data, SMCallback callback)	Constructs a new SMEVENTUSERUNREGISTER
SMEVENTUSERUNREGISTER (java.util.Hashtable<java.lang.String,java.lang.String> data,	Constructs a new

Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method	Description	
boolean	<code>equals (java.lang.Object o)</code>	Compares this instance with the specified object and indicates if they are equal.	
int	<code>hashCode ()</code>	Returns an integer hash code for this object.	
void	<code>readExternal (java.io.ObjectInput serializedObject)</code>	This method is called when deserializing the object.	
void	<code>writeExternal (java.io.ObjectOutput serializedObject)</code>	This method is called when serializing the object.	

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, toString, wait, wait, wait`

Field Detail

Email

`public java.lang.String Email`

Constructor Detail

SMEventUserUnregister

`public SMEventUserUnregister()`

SMEventUserUnregister

`public SMEventUserUnregister(java.lang.String email,
java.util.Hashtable<java.lang.String,java.lang.String> data,
SMCallback callback)`

Constructs a new SMEventUserUnregister

Parameters:

email - a String containing the e-mail address of the user.

data - a Hashtable<String, String> containing custom data, can be null.

callback - an SMCallback containing code to perform after the message is sent

See Also:

[SMCallback](#)

SMEventUserUnregister

```
public SMEventUserUnregister(java.util.Hashtable<java.lang.String,java.lang.String> data,  
                           SMCallback callback)
```

Constructs a new SMEventUserUnregister

Parameters:

data - a Hashtable<String, String> containing custom data, can be null.

callback - an SMCallback containing code to perform after the message is sent

Since:

3.2

See Also:

[SMCallback](#)

Method Detail

equals

```
public boolean equals(java.lang.Object o)
```

Description copied from class: [SMEvent](#)

Compares this instance with the specified object and indicates if they are equal.

Overrides:

[equals](#) in class [SMEvent](#)

Parameters:

o - the object to compare this instance with.

Returns:

true if the specified object is equal to this Object; false otherwise.

hashCode

```
public int hashCode()
```

Description copied from class: [SMEvent](#)

Returns an integer hash code for this object. By contract, any two objects for which `SMEvent.equals(Object)` returns true must return the same hash code value. This means that subclasses of Object usually override both methods or neither method.

Overrides:

```
hashCode in class SMEvent
```

Returns:

```
this object's hash code.
```

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.langClassNotFoundException
```

Description copied from class: [SMEvent](#)

This method is called when deserializing the object. It should not be called manually.

Specified by:

```
readExternal in interface java.io.Externalizable
```

Overrides:

```
readExternal in class SMEvent
```

Parameters:

`serializedObject` - the `ObjectInput` object containing all the values of the `SMEvent` object to recreate.

Throws:

```
java.io.IOException
```

```
java.langClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

Description copied from class: [SMEvent](#)

This method is called when serializing the object. It should not be called manually.

Specified by:

```
writeExternal in interface java.io.Externalizable
```

Overrides:

```
writeExternal in class SMEvent
```

Parameters:

serializedObject - the ObjectOutputStream object used to store the values of the SMEvent object

Throws:

java.io.IOException

Package com.selligent.sdk

Class SMForegroundGcmBroadcastReceiver

```
java.lang.Object  
    android.content.BroadcastReceiver  
        com.selligent.sdk.SMForegroundGcmBroadcastReceiver
```

```
public class SMForegroundGcmBroadcastReceiver  
extends android.content.BroadcastReceiver
```

Class implementing the receiver that will listen to connectivity changes. When the device is back online, it will check if some events were not sent to the Selligent Mobile platform due to lack of connectivity and, in that case, will retry sending them. If you do not extend `SMBaseActivity`, you have to register and unregister this receiver respectively on the `onStart` and `onStop` events of your activities.

Since:

1.0

Version:

3.9

Nested Class Summary

Nested classes/interfaces inherited from class android.content.BroadcastReceiver

```
android.content.BroadcastReceiver.PendingResult
```

Constructor Summary

Constructors

Constructor	Description
<code>SMForegroundGcmBroadcastReceiver (android.content.Context context)</code>	Constructor of the class.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
<code>android.content.IntentFilter</code>	<code>getIntentFilter()</code>	It creates the IntentFilter that must be used when registering the receiver.
<code>void</code>		This method is called

onReceive (android.content.Context context, android.content.Intent intent)

when the BroadcastReceiver is receiving an Intent broadcast.

Methods inherited from class android.content.BroadcastReceiver

abortBroadcast, clearAbortBroadcast, getAbortBroadcast, getDebugUnregister, getResultCode, getResultData, getResultExtras, goAsync, isInitialStickyBroadcast, isOrderedBroadcast, peekService, setDebugUnregister, setOrderedHint, setResult, setResultCode, setResultData, setResultExtras

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

SMForegroundGcmBroadcastReceiver

public SMForegroundGcmBroadcastReceiver(android.content.Context context)

Constructor of the class.

Parameters:

context - the activity in which the receiver is instantiated.

Method Detail

getIntentFilter

public android.content.IntentFilter getIntentFilter()

It creates the IntentFilter that must be used when registering the receiver.

Returns:

the intent filter needed at registration.

onReceive

public void onReceive(android.content.Context context, android.content.Intent intent)

This method is called when the BroadcastReceiver is receiving an Intent broadcast.

Specified by:

onReceive in class android.content.BroadcastReceiver

Parameters:

context - The Context in which the receiver is running

intent - The Intent being received

Class SMInAppContent

java.lang.Object
 com.selligent.sdk.SMInAppContent

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

```
public class SMInAppContent
extends java.lang.Object
implements java.io.Externalizable
```

An In App content

Since:

1.4

Version:

3.9

See Also:

[Serialized Form](#)

Nested Class Summary

Nested Classes

Modifier and Type	Class	Description
static class	SMInAppContent.DisplayMode	Enum listing the different display mode.

Constructor Summary

Constructors

Constructor	Description
SMInAppContent()	Empty constructor of an In App Content
SMInAppContent (java.lang.String json)	Constructor that fills in the In App Content based on the given json

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
-------------------	--------	-------------

boolean	equals (java.lang.Object o)	Compares this instance with the specified object and indicates if they are equal.
java.lang.String	getBody ()	Gets the body of the SMInAppContent
java.lang.String	getCategory ()	Gets the category of the SMInAppContent
long	getCreationDate ()	Gets the creation date of the SMInAppContent
SMInAppContent.DisplayMode	getDisplayMode ()	Gets the display mode of the SMInAppContent
long	getExpirationDate ()	Gets the expiration date of the SMInAppContent
java.lang.String	getId ()	Gets the id of the SMInAppContent
android.graphics.Bitmap	getImage ()	Gets the bitmap of the image if the In App Content is of type Image and marked for its content to be downloaded.
SMLink []	getLinks ()	Gets the links of the SMInAppContent
java.lang.String	getTitle ()	Gets the title of the SMInAppContent
SMContentType	getType ()	Gets the type of the SMInAppContent

boolean	<code>hasBeenFirstSeenInCurrentSession()</code>	Tells if the SMInAppContent has already been seen or not in the current session.
boolean	<code>hasBeenSeen()</code>	Tells if the SMInAppContent has already been seen or not.
int	<code>hashCode()</code>	Returns an integer hash code for this object.
void	<code>readExternal (java.io.ObjectInput serializedObject)</code>	This method is called when deserializing the object.
void	<code>writeExternal (java.io.ObjectOutput serializedObject)</code>	This method is called when serializing the object.

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

SMInAppContent

`public SMInAppContent()`

Empty constructor of an In App Content

SMInAppContent

`public SMInAppContent(java.lang.String jSon)`

Constructor that fills in the In App Content based on the given json

Parameters:

`jSon` - The json String representing an In-App content

Method Detail

equals

```
public boolean equals(java.lang.Object o)
```

Compares this instance with the specified object and indicates if they are equal.

Overrides:

equals in class java.lang.Object

Parameters:

`o` – the object to compare this instance with.

Returns:

true if the specified object is equal to this Object; false otherwise.

hashCode

```
public int hashCode()
```

Returns an integer hash code for this object. By contract, any two objects for which `equals (Object)` returns true must return the same hash code value. This means that subclasses of Object usually override both methods or neither method.

Overrides:

hashCode in class java.lang.Object

Returns:

this object's hash code.

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

This method is called when deserializing the object. It should not be called manually.

Specified by:

readExternal in interface java.io.Externalizable

Parameters:

`serializedObject` – the ObjectInput object containing all the values of the SMIInAppContent object to recreate.

Throws:

`java.io.IOException`

`java.lang.ClassNotFoundException`

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

This method is called when serializing the object. It should not be called manually.

Specified by:

writeExternal in interface java.io.Externalizable

Parameters:

serializedObject - the ObjectOutputStream object used to store the values of the SMInAppContent object

Throws:

java.io.IOException

getId

```
public java.lang.String getId()
```

Gets the id of the SMInAppContent

Returns:

the String representing the id

Since:

1.4

getTitle

```
public java.lang.String getTitle()
```

Gets the title of the SMInAppContent

Returns:

the String representing the title

Since:

1.4

getBody

```
public java.lang.String getBody()
```

Gets the body of the SMInAppContent

Returns:

the String representing the body

Since:

1.4

getLinks

public SMLink[] getLinks()

Gets the links of the SMInAppContent

Returns:

an array of SMLink

Since:

1.4

getType

public SMContentType getType()

Gets the type of the SMInAppContent

Returns:

the SMContentType

Since:

1.4

getCategory

public java.lang.String getCategory()

Gets the category of the SMInAppContent

Returns:

the String representing the category

Since:

1.4

getDisplayMode

public SMInAppContent.DisplayMode getDisplayMode()

Gets the display mode of the SMInAppContent

Returns:

the SMInAppContent.DisplayMode

Since:

1.4

getCreationDate

```
public long getCreationDate()
```

Gets the creation date of the SMInAppContent

Returns:

the long representing the creation date in milliseconds since 01/01/1970

Since:

1.4

getExpirationDate

```
public long getExpirationDate()
```

Gets the expiration date of the SMInAppContent

Returns:

the long representing the expiration date in milliseconds since 01/01/1970

Since:

1.4

hasBeenSeen

```
public boolean hasBeenSeen()
```

Tells if the SMInAppContent has already been seen or not.

Returns:

true if it was seen, false otherwise.

Since:

1.4

hasBeenFirstSeenInCurrentSession

```
public boolean hasBeenFirstSeenInCurrentSession()
```

Tells if the SMInAppContent has already been seen or not in the current session.

Returns:

true if it was seen, false otherwise.

Since:

1.5

getImage

```
@Nullable  
public android.graphics.Bitmap getImage()
```

Gets the bitmap of the image if the In App Content is of type Image and marked for its content to be downloaded.

Returns:

the bitmap of the image if it exists, null otherwise

Class SMInAppContentHtmlFragment

```
java.lang.Object
    androidx.fragment.app.Fragment
        androidx.fragment.app.DialogFragment
            com.selligent.sdk.SMInAppContentHtmlFragment
```

All Implemented Interfaces:

```
android.content.ComponentCallbacks, android.content.DialogInterface.OnCancelListener,
android.content.DialogInterface.OnDismissListener, android.view.View.OnCreateContextMenuListener,
androidx.lifecycle.LifecycleOwner, androidx.lifecycle.ViewModelStoreOwner,
androidx.savedstate.SavedStateRegistryOwner
```

```
public class SMInAppContentHtmlFragment
extends androidx.fragment.app.DialogFragment
```

This class implements a fragment that will display one or several HTML contents. It can either be used as a standard fragment or as a full screen dialog fragment

Since:

1.4

Version:

3.9

Nested Class Summary

Nested classes/interfaces inherited from class androidx.fragment.app.Fragment

```
androidx.fragment.app.Fragment.InstantiationException, androidx.fragment.app.Fragment.SavedState
```

Field Summary

Fields inherited from class androidx.fragment.app.DialogFragment

```
STYLE_NO_FRAME, STYLE_NO_INPUT, STYLE_NO_TITLE, STYLE_NORMAL
```

Constructor Summary

Constructors

Constructor	Description
SMInAppContentHtmlFragment()	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
java.lang.String	getContentCategory()	This method returns the category given at the creation of the instance of SMInAppContentFragment
int		This method returns the number

	<code>getContentCount()</code>	of contents in the fragment
<code>SMContentType</code>	<code>getContentType()</code>	This method returns the type given at the creation of the instance of SMInAppContentFragment
boolean	<code>hasContent()</code>	Tells if the fragment has content or not.
static <code>SMInAppContentHtmlFragment</code>	<code>newInstance (java.lang.String category)</code>	Method used to create a new instance of SMInAppContentHtmlFragment that will display all the HTML In App Contents available
static <code>SMInAppContentHtmlFragment</code>	<code>newInstance (java.lang.String category, int count)</code>	Method used to create a new instance of SMInAppContentHtmlFragment
<code>android.view.View</code>	<code>onCreateView (android.view.LayoutInflater inflater, android.view.ViewGroup container, android.os.Bundle savedInstanceState)</code>	
void	<code>onSaveInstanceState (android.os.Bundle savedInstanceState)</code>	
void	<code>onViewCreated (android.view.View view, android.os.Bundle savedInstanceState)</code>	
void	<code>refresh()</code>	This method will refresh the content by getting it from the cache again and then visually refresh the component if the fragment is visible.
void	<code>show (androidx.fragment.app.FragmentManager fragmentManager, java.lang.String tag)</code>	Display the fragment as a full screen dialog, adding the fragment to the given FragmentManager.
int	<code>show (androidx.fragment.app.FragmentTransaction fragmentTransaction, java.lang.String tag)</code>	Display the fragment as a full screen dialog, adding the fragment using an existing transaction and then committing the transaction.

Methods inherited from class androidx.fragment.app.DialogFragment

dismiss, dismissAllowingStateLoss, getDialog, getShowsDialog, getTheme, isCancelable, onActivityCreated, onAttach, onCancel, onCreate, onCreateDialog, onDestoryView, onDetach, onDismiss, onGetLayoutInflater, onStart, onStop, requireDialog, setCancelable, setShowsDialog, setStyle, setupDialog, showNow

Methods inherited from class androidx.fragment.app.Fragment

dump, equals, getActivity, getAllowEnterTransitionOverlap, getAllowReturnTransitionOverlap, getArguments, getChildFragmentManager, getContext, getEnterTransition, getExitTransition, getFragmentManager, getHost, getId, getLayoutInflater, getLayoutInflater, getLifecycle, getLoaderManager, getParentFragment, getReenterTransition, getResources, getRetainInstance, getReturnTransition, getSavedStateRegistry, getSharedElementEnterTransition, getSharedElementReturnTransition, getString, getString, getTag, getTargetFragment, getTargetRequestCode, getText, getUserVisibleHint, getView, getViewLifecycleOwner, getViewLifecycleOwnerLiveData, getModelStore, hashCode, hasOptionsMenu, instantiate, instantiate, isAdded, isDetached, isHidden, isInLayout, isMenuVisible, isRemoving, isResumed, isStateSaved, isVisible, onActivityResult, onAttach, onAttachFragment, onConfigurationChanged, onContextItemSelected, onCreateAnimation, onCreateAnimator, onCreateContextMenu, onCreateOptionsMenu, onDestroy, onDestoryOptionsMenu, onHiddenChanged, onInflate, onInflate, onLowMemory, onMultiWindowModeChanged,

```
onOptionsItemSelected, onOptionsMenuClosed, onPause, onPictureInPictureModeChanged, onPrepareOptionsMenu,
onPrimaryNavigationFragmentChanged, onRequestPermissionsResult, onResume, onViewStateRestored,
postponeEnterTransition, postponeEnterTransition, registerForContextMenu, requestPermissions,
requireActivity, requireArguments, requireContext, requireFragmentManager, requireHost,
requireParentFragment, requireView, setAllowEnterTransitionOverlap, setAllowReturnTransitionOverlap,
setArguments, setEnterSharedElementCallback, setEnterTransition, setExitSharedElementCallback,
setExitTransition, setHasOptionsMenu, setInitialSavedState, setMenuVisibility, setReenterTransition,
setRetainInstance, setReturnTransition, setSharedElementEnterTransition, setSharedElementReturnTransition,
setTargetFragment, setUserVisibleHint, shouldShowRequestPermissionRationale, startActivity, startActivity,
startActivityForResult, startActivityForResult, startIntentSenderForResult, startPostponedEnterTransition,
toString, unregisterForContextMenu
```

Methods inherited from class java.lang.Object

```
getClass, notify, notifyAll, wait, wait, wait
```

Constructor Detail

SMInAppContentHtmlFragment

```
public SMInAppContentHtmlFragment()
```

Method Detail

newInstance

```
public static SMInAppContentHtmlFragment newInstance(java.lang.String category)
```

Method used to create a new instance of SMInAppContentHtmlFragment that will display all the HTML In App Contents available

Parameters:

category - String specifying the category of the content

Returns:

the new instance of SMInAppContentImageFragment or null if one of the parameters is null.

Since:

1.4

newInstance

```
public static SMInAppContentHtmlFragment newInstance(java.lang.String category,
                                                    int count)
```

Method used to create a new instance of SMInAppContentHtmlFragment

Parameters:

category - String specifying the category of the content

count - number of In App contents to display. A value of -1 will display all contents available. Values 0 and inferior to -1 are invalid.

Returns:

the new instance of SMInAppContentImageFragment or null if one of the parameters is null.

Since:

1.4

onCreateView

```
public android.view.View onCreateView(@NonNull  
        android.view.LayoutInflater inflater,  
        android.view.ViewGroup container,  
        android.os.Bundle savedInstanceState)
```

Overrides:

onCreateView in class androidx.fragment.app.Fragment

onViewCreated

```
public void onViewCreated(@NonNull  
        android.view.View view,  
        android.os.Bundle savedInstanceState)
```

Overrides:

onViewCreated in class androidx.fragment.app.Fragment

onSaveInstanceState

```
public void onSaveInstanceState(@NonNull  
        android.os.Bundle savedInstanceState)
```

Overrides:

onSaveInstanceState in class androidx.fragment.app.DialogFragment

show

```
public void show(@NonNull  
        androidx.fragment.app.FragmentManager fragmentManager,  
        java.lang.String tag)
```

Display the fragment as a full screen dialog, adding the fragment to the given FragmentManager. This is a convenience for explicitly creating a transaction, adding the fragment to it with the given tag, and committing it. This does not add the transaction to the back stack. When the fragment is dismissed, a new transaction will be executed to remove it from the activity. Does nothing if there is no content.

Overrides:

show in class androidx.fragment.app.DialogFragment

Parameters:

fragmentManager - The FragmentManager this fragment will be added to.

tag - String to identify the fragment

Since:

1.4

show

```
public int show(@NonNull  
        androidx.fragment.app.FragmentTransaction fragmentTransaction,  
        java.lang.String tag)
```

Display the fragment as a full screen dialog, adding the fragment using an existing transaction and then committing the transaction. Does nothing if there is no content.

Overrides:

show in class androidx.fragment.app.DialogFragment

Parameters:

fragmentTransaction - An existing transaction in which to add the fragment.

```
tag - String to identify the fragment
```

Returns:

the identifier of the committed transaction, -1 if there is no content.

Since:

1.4 WARNING: If you set `SMSettings.LoadCacheAsynchronously` to true, there might not be any content yet when this is executed. Prefer the method `DialogFragment.show(FragmentManager, String)` which makes sure the content is retrieved before showing the dialog.

hasContent

```
public boolean hasContent()
```

Tells if the fragment has content or not. This is useful to know if the fragment can be displayed or not.

Returns:

true if there is content, false otherwise

Since:

1.4 WARNING: If you set `SMSettings.LoadCacheAsynchronously` to true, there might not be any content yet when this is executed.

getContentCategory

```
public java.lang.String getContentCategory()
```

This method returns the category given at the creation of the instance of `SMInAppContentFragment`

Returns:

a String representing the category

Since:

1.4

getContentType

```
public SMContentType getContentType()
```

This method returns the type given at the creation of the instance of `SMInAppContentFragment`

Returns:

the `SMContentType`

getContentCount

```
public int getContentCount()
```

This method returns the number of contents in the fragment

Returns:

an int corresponding to the number of contents in the fragment

Since:

1.4

refresh

```
public void refresh()
```

This method will refresh the content by getting it from the cache again and then visually refresh the component if the fragment is visible.

Class SMInAppContentImageFragment

```
java.lang.Object
    androidx.fragment.app.Fragment
        androidx.fragment.app.DialogFragment
            com.selligent.sdk.SMInAppContentImageFragment
```

All Implemented Interfaces:

```
android.content.ComponentCallbacks, android.content.DialogInterface.OnCancelListener,
android.content.DialogInterface.OnDismissListener, android.view.View.OnCreateContextMenuListener,
android.view.View.OnTouchListener, androidx.lifecycle.LifecycleOwner, androidx.lifecycle.ViewModelStoreOwner,
androidx.savedstate.SavedStateRegistryOwner
```

```
public class SMInAppContentImageFragment
extends androidx.fragment.app.DialogFragment
implements android.view.View.OnTouchListener
```

This class implements a fragment that will display In App Content containing an image. It can either be used as a standard fragment or as a full screen dialog fragment

Since:

1.4

Version:

3.9

Nested Class Summary

Nested classes/interfaces inherited from class androidx.fragment.app.Fragment

```
androidx.fragment.app.Fragment.InstantiationException, androidx.fragment.app.Fragment.SavedState
```

Field Summary

Fields inherited from class androidx.fragment.app.DialogFragment

```
STYLE_NO_FRAME, STYLE_NO_INPUT, STYLE_NO_TITLE, STYLE_NORMAL
```

Constructor Summary

Constructors

Constructor	Description
SMInAppContentImageFragment()	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
java.lang.String	getContentCategory()	This method returns the category given at the creation of the instance of SMInAppContentFragment
int		This method returns the number

	<code>getContentCount()</code>	of contents in the fragment
<code>SMContentType</code>	<code>getContentType()</code>	This method returns the type given at the creation of the instance of SMInAppContentFragment
boolean	<code>hasContent()</code>	Tells if the fragment has content or not.
static <code>SMInAppContentImageFragment</code>	<code>newInstance (java.lang.String category)</code>	Method used to create a new instance of SMInAppContentImageFragment
<code>android.view.View</code>	<code>onCreateView (android.view.LayoutInflater inflater, android.view.ViewGroup container, android.os.Bundle savedInstanceState)</code>	
void	<code>onSaveInstanceState (android.os.Bundle savedInstanceState)</code>	
boolean	<code>onTouch (android.view.View view, android.view.MotionEvent motionEvent)</code>	
void	<code>onViewCreated (android.view.View view, android.os.Bundle savedInstanceState)</code>	
void	<code>refresh()</code>	This method will refresh the content by getting it from the cache again and then visually refresh the component if the fragment is visible.
void	<code>show (androidx.fragment.app.FragmentManager fragmentManager, java.lang.String tag)</code>	Display the fragment as a full screen dialog, adding the fragment to the given FragmentManager.
int	<code>show (androidx.fragment.app.FragmentTransaction fragmentTransaction, java.lang.String tag)</code>	Display the fragment as a full screen dialog, adding the fragment using an existing transaction and then committing the transaction.

Methods inherited from class androidx.fragment.app.DialogFragment

dismiss, dismissAllowingStateLoss, getDialog, getShowsDialog, getTheme, isCancelable, onActivityCreated, onAttach, onCancel, onCreate, onCreateDialog, onDestroyView, onDetach, onDismiss, onGetLayoutInflater, onStart, onStop, requireDialog, setCancelable, setShowsDialog, setStyle, setupDialog, showNow

Methods inherited from class androidx.fragment.app.Fragment

dump, equals, getActivity, getAllowEnterTransitionOverlap, getAllowReturnTransitionOverlap, getArguments, getChildFragmentManager, getContext, getEnterTransition, getExitTransition, getFragmentManager, getHost, getId, getLayoutInflater, getLayoutInflater, getLifecycle, getLoaderManager, getParentFragment, getReenterTransition, getResources, getRetainInstance, getReturnTransition, getSavedStateRegistry, getSharedElementEnterTransition, getSharedElementReturnTransition, getString, getString, getTag, getTargetFragment, getTargetRequestCode, getText, getUserVisibleHint, getView, getViewLifecycleOwner, getViewLifecycleOwnerLiveData, getViewModelStore, hashCode, hasOptionsMenu, instantiate, instantiate, isAdded, isDetached, isHidden, isInLayout, isMenuVisible, isRemoving, isResumed, isStateSaved, isVisible, onActivityResult, onAttach, onAttachFragment, onConfigurationChanged, onContextItemSelected, onCreateAnimation, onCreateAnimator, onCreateOptionsMenu, onCreateOptionsMenu, onDestory, onDestoryOptionsMenu, onHiddenChanged, onInflate, onInflate, onLowMemory, onMultiWindowModeChanged, onOptionsItemSelected, onOptionsMenuClosed, onPause, onPictureInPictureModeChanged, onPrepareOptionsMenu, onPrimaryNavigationFragmentChanged, onRequestPermissionsResult, onResume, onViewStateRestored, postponeEnterTransition, postponeEnterTransition, registerForContextMenu, requestPermissions, requireActivity, requireArguments, requireContext, requireFragmentManager, requireHost, requireParentFragment, requireView, setAllowEnterTransitionOverlap, setAllowReturnTransitionOverlap,

```
setArguments, setEnterSharedElementCallback, setEnterTransition, setExitSharedElementCallback,  
setExitTransition, setHasOptionsMenu, setInitialSavedState, setMenuVisibility, setReenterTransition,  
setRetainInstance, setReturnTransition, setSharedElementEnterTransition, setSharedElementReturnTransition,  
setTargetFragment, setUserVisibleHint, shouldShowRequestPermissionRationale, startActivity, startActivity,  
startActivityForResult, startActivityForResult, startIntentSenderForResult, startPostponedEnterTransition,  
toString, unregisterForContextMenu
```

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, wait, wait, wait`

Constructor Detail

SMInAppContentImageFragment

```
public SMInAppContentImageFragment()
```

Method Detail

newInstance

```
public static SMInAppContentImageFragment newInstance(java.lang.String category)
```

Method used to create a new instance of SMInAppContentImageFragment

Parameters:

`category` - String specifying the category of the content

Returns:

the new instance of SMIInAppContentImageFragment or null if one of the parameters is null.

Since:

1.4

onTouch

Specified by:

onTouch in interface android.view.View.OnTouchListener

onCreateView

```
public android.view.View onCreateView(@NonNull  
        android.view.LayoutInflater inflater,  
        android.view.ViewGroup container,  
        android.os.Bundle savedInstanceState)
```

Overrides:

onCreateView in class androidx.fragment.app.Fragment

onViewCreated

```
public void onViewCreated(@NotNull  
        android.view.View view,  
        android.os.Bundle savedInstanceState)
```

Overrides:

onViewCreated in class androidx.fragment.app.Fragment

onSaveInstanceState

```
public void onSaveInstanceState(@NonNull  
        android.os.Bundle savedInstanceState)
```

Overrides:

onSaveInstanceState in class androidx.fragment.app.DialogFragment

show

```
public void show(@NonNull  
        androidx.fragment.app.FragmentManager fragmentManager,  
        java.lang.String tag)
```

Display the fragment as a full screen dialog, adding the fragment to the given FragmentManager. This is a convenience for explicitly creating a transaction, adding the fragment to it with the given tag, and committing it. This does not add the transaction to the back stack. When the fragment is dismissed, a new transaction will be executed to remove it from the activity. Does nothing if there is no content.

Overrides:

show in class androidx.fragment.app.DialogFragment

Parameters:

fragmentManager - The FragmentManager this fragment will be added to.

tag - String to identify the fragment

Since:

1.4

show

```
public int show(@NonNull  
        androidx.fragment.app.FragmentTransaction fragmentTransaction,  
        java.lang.String tag)
```

Display the fragment as a full screen dialog, adding the fragment using an existing transaction and then committing the transaction. Does nothing if there is no content.

Overrides:

show in class androidx.fragment.app.DialogFragment

Parameters:

fragmentTransaction - An existing transaction in which to add the fragment.

tag - String to identify the fragment

Returns:

the identifier of the committed transaction, -1 if there is no content.

Since:

1.4 WARNING: If you set SMSettings.LoadCacheAsynchronously to true, there might not be any content yet when this is executed. Prefer the method DialogFragment.show(FragmentManager, String) which makes sure the content is retrieved before showing the dialog.

hasContent

```
public boolean hasContent()
```

Tells if the fragment has content or not. This is useful to know if the fragment can be displayed or not.

Returns:

true if there is content, false otherwise

Since:

1.4 WARNING: If you set `SMSettings.LoadCacheAsynchronously` to true, there might not be any content yet when this is executed.

getContentCategory

```
public java.lang.String getContentCategory()
```

This method returns the category given at the creation of the instance of `SMInAppContentFragment`

Returns:

a String representing the category

Since:

1.4

getContentType

```
public SMContentType getContentType()
```

This method returns the type given at the creation of the instance of `SMInAppContentFragment`

Returns:

the `SMContentType`

getContentCount

```
public int getContentCount()
```

This method returns the number of contents in the fragment

Returns:

an int corresponding to the number of contents in the fragment

Since:

1.4

refresh

```
public void refresh()
```

This method will refresh the content by getting it from the cache again and then visually refresh the component if the fragment is visible.

Class SMInAppContentUrlFragment

```
java.lang.Object
    androidx.fragment.app.Fragment
        androidx.fragment.app.DialogFragment
            com.selligent.sdk.SMInAppContentUrlFragment
```

All Implemented Interfaces:

```
android.content.ComponentCallbacks, android.content.DialogInterface.OnCancelListener,
android.content.DialogInterface.OnDismissListener, android.view.View.OnCreateContextMenuListener,
androidx.lifecycle.LifecycleOwner, androidx.lifecycle.ViewModelStoreOwner,
androidx.savedstate.SavedStateRegistryOwner
```

```
public class SMInAppContentUrlFragment
extends androidx.fragment.app.DialogFragment
```

This class implements a fragment that will display In App Content containing an url. It can either be used as a standard fragment or as a full screen dialog fragment

Since:

1.4

Version:

3.9

Nested Class Summary

Nested classes/interfaces inherited from class androidx.fragment.app.Fragment

```
androidx.fragment.app.Fragment.InstantiationException, androidx.fragment.app.Fragment.SavedState
```

Field Summary

Fields inherited from class androidx.fragment.app.DialogFragment

```
STYLE_NO_FRAME, STYLE_NO_INPUT, STYLE_NO_TITLE, STYLE_NORMAL
```

Constructor Summary

Constructors

Constructor	Description
<code>SMInAppContentUrlFragment ()</code>	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
<code>java.lang.String</code>	<code>getContentCategory ()</code>	This method returns the category given at the creation of the instance of SMInAppContentFragment
<code>int</code>		

	<code>getContentCount()</code>	This method returns the number of contents in the fragment
<code>SMContentType</code>	<code>getContentType()</code>	This method returns the type given at the creation of the instance of <code>SMInAppContentFragment</code>
<code>boolean</code>	<code>hasContent()</code>	Tells if the fragment has content or not.
<code>static SMInAppContentUrlFragment</code>	<code>newInstance (java.lang.String category)</code>	Method used to create a new instance of <code>SMInAppContentImageFragment</code>
<code>android.view.View</code>	<code>onCreateView (android.view.LayoutInflater inflater, android.view.ViewGroup container, android.os.Bundle savedInstanceState)</code>	
<code>void</code>	<code>onSaveInstanceState (android.os.Bundle savedInstanceState)</code>	
<code>void</code>	<code>onViewCreated (android.view.View view, android.os.Bundle savedInstanceState)</code>	
<code>void</code>	<code>refresh()</code>	This method will refresh the content by getting it from the cache again and then visually refresh the component if the fragment is visible.
<code>void</code>	<code>show (androidx.fragment.app.FragmentManager fragmentManager, java.lang.String tag)</code>	Display the fragment as a full screen dialog, adding the fragment to the given <code>FragmentManager</code> .
<code>int</code>	<code>show (androidx.fragment.app.FragmentTransaction fragmentTransaction, java.lang.String tag)</code>	Display the fragment as a full screen dialog, adding the fragment using an existing transaction and then committing the transaction.

Methods inherited from class androidx.fragment.app.DialogFragment

`dismiss, dismissAllowingStateLoss, getDialog, getShowsDialog, getTheme, isCancelable, onActivityCreated, onAttach, onCancel, onCreate, onCreateDialog, onDestroyView, onDetach, onDismiss, onGetLayoutInflater, onStart, onStop, requireDialog, setCancelable, setShowsDialog, setStyle, setupDialog, showNow`

Methods inherited from class androidx.fragment.app.Fragment

`dump, equals, getActivity, getAllowEnterTransitionOverlap, getAllowReturnTransitionOverlap, getArguments, getChildFragmentManager, getContext, getEnterTransition, getExitTransition, getFragmentManager, getHost, getId, getLayoutInflater, getLayoutInflater, getLifecycle, getLoaderManager, getParentFragment, getReenterTransition, getResources, getRetainInstance, getReturnTransition, getSavedStateRegistry, getSharedElementEnterTransition, getSharedElementReturnTransition, getString, getString, getTag, getTargetFragment, getTargetRequestCode, getText, getUserVisibleHint, getView, getViewLifecycleOwner, getViewLifecycleOwnerLiveData, getViewModelStore, hashCode, hasOptionsMenu, instantiate, instantiate, isAdded, isDetached, isHidden, isInLayout, isMenuVisible, isRemoving, isResumed, isStateSaved, isVisible, onActivityResult, onAttach, onAttachFragment, onConfigurationChanged, onContextItemSelected, onCreateAnimation, onCreateAnimator, onCreateContextMenu, onCreateOptionsMenu, onDestroy, onDestroyOptionsMenu, onHiddenChanged, onInflate, onInflate, onLowMemory, onMultiWindowModeChanged, onOptionsItemSelected, onOptionsMenuClosed, onPause, onPictureInPictureModeChanged, onPrepareOptionsMenu, onPrimaryNavigationFragmentChanged, onRequestPermissionsResult, onResume, onViewStateRestored, postponeEnterTransition, postponeEnterTransition, registerForContextMenu, requestPermissions, requireActivity, requireArguments, requireContext, requireFragmentManager, requireHost, requireParentFragment, requireView, setAllowEnterTransitionOverlap, setAllowReturnTransitionOverlap,`

```
setArguments, setEnterSharedElementCallback, setEnterTransition, setExitSharedElementCallback,
setExitTransition, setHasOptionsMenu, setInitialSavedState, setMenuVisibility, setReenterTransition,
setRetainInstance, setReturnTransition, setSharedElementEnterTransition, setSharedElementReturnTransition,
setTargetFragment, setUserVisibleHint, shouldShowRequestPermissionRationale, startActivity, startActivityForResult,
startActivityForResult, startActivityForResultForResult, startIntentSenderForResult, startPostponedEnterTransition,
toString, unregisterForContextMenu
```

Methods inherited from class java.lang.Object

```
getClass, notify, notifyAll, wait, wait, wait
```

Constructor Detail

SMInAppContentUrlFragment

```
public SMInAppContentUrlFragment()
```

Method Detail

newInstance

```
public static SMInAppContentUrlFragment newInstance(java.lang.String category)
```

Method used to create a new instance of SMInAppContentImageFragment

Parameters:

category - String specifying the category of the content

Returns:

the new instance of SMInAppContentImageFragment or null if one of the parameters is null.

Since:

1.4

onCreateView

```
public android.view.View onCreateView(@NonNull
        android.view.LayoutInflater inflater,
        android.view.ViewGroup container,
        android.os.Bundle savedInstanceState)
```

Overrides:

onCreateView in class androidx.fragment.app.Fragment

onViewCreated

```
public void onViewCreated(@NonNull
        android.view.View view,
        android.os.Bundle savedInstanceState)
```

Overrides:

onViewCreated in class androidx.fragment.app.Fragment

onSaveInstanceState

```
public void onSaveInstanceState(@NonNull
```

```
        android.os.Bundle savedInstanceState)
```

Overrides:

```
onSaveInstanceState in class androidx.fragment.app.DialogFragment
```

show

```
public void show(@NonNull  
        androidx.fragment.app.FragmentManager fragmentManager,  
        java.lang.String tag)
```

Display the fragment as a full screen dialog, adding the fragment to the given FragmentManager. This is a convenience for explicitly creating a transaction, adding the fragment to it with the given tag, and committing it. This does not add the transaction to the back stack. When the fragment is dismissed, a new transaction will be executed to remove it from the activity. Does nothing if there is no content.

Overrides:

```
show in class androidx.fragment.app.DialogFragment
```

Parameters:

fragmentManager - The FragmentManager this fragment will be added to.

tag - String to identify the fragment

Since:

1.4

show

```
public int show(@NonNull  
        androidx.fragment.app.FragmentTransaction fragmentTransaction,  
        java.lang.String tag)
```

Display the fragment as a full screen dialog, adding the fragment using an existing transaction and then committing the transaction. Does nothing if there is no content.

Overrides:

```
show in class androidx.fragment.app.DialogFragment
```

Parameters:

fragmentTransaction - An existing transaction in which to add the fragment.

tag - String to identify the fragment

Returns:

the identifier of the committed transaction, -1 if there is no content.

Since:

1.4 WARNING: If you set `SMSettings.LoadCacheAsynchronously` to true, there might not be any content yet when this is executed. Prefer the method `DialogFragment.show(FragmentManager, String)` which makes sure the content is retrieved before showing the dialog.

hasContent

```
public boolean hasContent()
```

Tells if the fragment has content or not. This is useful to know if the fragment can be displayed or not.

Returns:

true if there is content, false otherwise

Since:

1.4 WARNING: If you set `SMSettings.LoadCacheAsynchronously` to true, there might not be any content yet when this is executed.

getContentCategory

```
public java.lang.String getContentCategory()
```

This method returns the category given at the creation of the instance of SMInAppContentFragment

Returns:

a String representing the category

Since:

1.4

getContentType

```
public SMContentType getContentType()
```

This method returns the type given at the creation of the instance of SMInAppContentFragment

Returns:

the SMContentType

getContentCount

```
public int getContentCount()
```

This method returns the number of contents in the fragment

Returns:

an int corresponding to the number of contents in the fragment

Since:

1.4

refresh

```
public void refresh()
```

This method will refresh the content by getting it from the cache again and then visually refresh the component if the fragment is visible.

Package com.selligent.sdk

Class SMInAppMessage

java.lang.Object
com.selligent.sdk.SMInAppMessage

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

```
public class SMInAppMessage
extends java.lang.Object
implements java.io.Externalizable
```

An In App message

Since:

1.3

Version:

3.9

See Also:

[Serialized Form](#)

Field Summary

Fields

Modifier and Type	Field	Description
java.lang.String	id	
java.lang.String	title	

Constructor Summary

Constructors

Constructor	Description
SMInAppMessage ()	
SMInAppMessage (java.lang.String json)	

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
boolean	<code>equals (java.lang.Object otherMessage)</code>	Compares this instance with the specified object and indicates if they are equal.
java.lang.String	<code>getBody ()</code>	Gets the body of the SMInAppMessage
<code>SMNotificationButton[]</code>	<code>getButtons ()</code>	Gets the buttons of the SMInAppMessage
long	<code>getCreationDate ()</code>	Gets the creation date of the SMInAppMessage
long	<code>getExpirationDate ()</code>	Gets the expiration date of the SMInAppMessage
java.lang.String	<code>getId ()</code>	Gets the id of the SMInAppMessage
<code>SMMapMarker[]</code>	<code>getMarkers ()</code>	Gets the list of the markers for an In-app message of type Map
long	<code>getReceptionDate ()</code>	Gets the reception date of the SMInAppMessage
java.lang.String	<code>getTitle ()</code>	Gets the title of the SMInAppMessage
<code>SMMessageType</code>	<code>getType ()</code>	Gets the type of the SMInAppMessage
boolean	<code>hasBeenSeen ()</code>	Tells if the SMInAppContent has already been seen or not.
int	<code>hashCode ()</code>	Returns an integer hash code for this object.
void	<code>readExternal (java.io.ObjectInput serializedObject)</code>	This method is called when deserializing the object.

```
void
```

```
writeExternal (java.io.ObjectOutput serializedObject)
```

This method is called when serializing the object.

Methods inherited from class java.lang.Object

```
getClass, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

id

```
public java.lang.String id
```

title

```
public java.lang.String title
```

Constructor Detail

SMInAppMessage

```
public SMInAppMessage()
```

SMInAppMessage

```
public SMInAppMessage(java.lang.String json)
```

Method Detail

equals

```
public boolean equals(java.lang.Object otherMessage)
```

Compares this instance with the specified object and indicates if they are equal.

Overrides:

```
equals in class java.lang.Object
```

Parameters:

otherMessage - the object to compare this instance with.

Returns:

true if the specified object is equal to this Object; false otherwise.

hashCode

```
public int hashCode()
```

Returns an integer hash code for this object. By contract, any two objects for which `equals (Object)` returns true must return the same hash code value. This means that subclasses of Object usually override both methods or neither method.

Overrides:

hashCode in class java.lang.Object

Returns:

this object's hash code.

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

This method is called when deserializing the object. It should not be called manually.

Specified by:

readExternal in interface java.io.Externalizable

Parameters:

serializedObject - the ObjectInput object containing all the values of the SMIInAppMessage object to recreate.

Throws:

java.io.IOException

java.lang.ClassNotFoundException

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

This method is called when serializing the object. It should not be called manually.

Specified by:

writeExternal in interface java.io.Externalizable

Parameters:

serializedObject - the ObjectOutputStream object used to store the values of the

SMInAppMessage object

Throws:

java.io.IOException

getId

public java.lang.String getId()

Gets the id of the SMInAppMessage

Returns:

the String representing the id

Since:

3.4

getTitle

public java.lang.String getTitle()

Gets the title of the SMInAppMessage

Returns:

the String representing the title

Since:

3.4

getBody

public java.lang.String getBody()

Gets the body of the SMInAppMessage

Returns:

the String representing the body

Since:

3.4

getMarkers

public SMMarker[] getMarkers()

Gets the list of the markers for an In-app message of type Map

Returns:

an array of SMMarker

Since:

3.4

getButtons

```
public SMNotificationButton[] getButtons()
```

Gets the buttons of the SMInAppMessage

Returns:

an array of SMLink

Since:

3.4

getType

```
public SMMimeType getType()
```

Gets the type of the SMInAppMessage

Returns:

the SMMimeType

Since:

3.4

getReceptionDate

```
public long getReceptionDate()
```

Gets the reception date of the SMInAppMessage

Returns:

the long representing the reception date in milliseconds since 01/01/1970

Since:

3.4

getCreationDate

```
public long getCreationDate()
```

Gets the creation date of the SMInAppMessage

Returns:

the long representing the creation date in milliseconds since 01/01/1970

Since:

3.4

getExpirationDate

public long getExpirationDate()

Gets the expiration date of the SMInAppMessage

Returns:

the long representing the expiration date in milliseconds since 01/01/1970

Since:

3.4

hasBeenSeen

public boolean hasBeenSeen()

Tells if the SMInAppContent has already been seen or not.

Returns:

true if it was seen, false otherwise.

Since:

3.4

Package com.selligent.sdk

Class SMLink

java.lang.Object
com.selligent.sdk.SMNotificationButton
com.selligent.sdk.SMLink

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

```
public class SMLink
extends SMNotificationButton
```

A link of an SMInAppContent.

Since:

1.4

Version:

3.8

See Also:

Serialized Form

Field Summary

Fields inherited from class com.selligent.sdk.SMNotificationButton

action, data, id, label, type, value

Constructor Summary

Constructors

Constructor	Description
SMLink()	

Method Summary

Methods inherited from class com.selligent.sdk.SMNotificationButton

getAction, getId, getLabel, getValue, readExternal, writeExternal

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

SMLink

```
public SMLink()
```

Class SMManger

`java.lang.Object`
`com.selligent.sdk.SMManger`

```
public class SMManger
extends java.lang.Object
```

Singleton object used to interact with the Selligent Mobile SDK.

Since:

1.0

Version:

3.9

Field Summary**Fields**

Modifier and Type	Field	Description
static java.lang.String	<code>BROADCAST_DATA_BUTTON</code>	Deprecated. <i>LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated.</i>
static java.lang.String	<code>BROADCAST_DATA_GCM_TOKEN</code>	Deprecated. <i>LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated.</i>
static java.lang.String	<code>BROADCAST_DATA_IN_APP_CONTENTS</code>	Deprecated. <i>LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated.</i>
static java.lang.String	<code>BROADCAST_DATA_IN_APP_MESSAGES</code>	Deprecated. <i>LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated.</i>
static java.lang.String	<code>BROADCAST_EVENT_BUTTON_CLICKED</code>	Deprecated. <i>LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated.</i>
static java.lang.String	<code>BROADCAST_EVENT_RECEIVED_GCM_TOKEN</code>	Deprecated. <i>LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated.</i>
static java.lang.String	<code>BROADCAST_EVENT_RECEIVED_IN_APP_CONTENTS</code>	Deprecated. <i>LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated.</i>
static java.lang.String	<code>BROADCAST_EVENT_RECEIVED_IN_APP_MESSAGE</code>	Deprecated. <i>LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated.</i>
static java.lang.String	<code>BROADCAST_EVENT_RECEIVED_REMOTE_NOTIFICATION</code>	Deprecated. <i>Since listening to broadcast in background is no longer possible under Android O, this broadcast is now deprecated (it is still sent though, so if you don't target Android O, you will still be able to listen to it).</i>
static java.lang.String	<code>BROADCAST_EVENT_WILL_DISMISS_NOTIFICATION</code>	Deprecated. <i>LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated.</i>
static java.lang.String		Deprecated.

	<code>BROADCAST_EVENT_WILL_DISPLAY_NOTIFICATION</code>	<i>LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated.</i>
<code>static boolean DEBUG</code>		When true, different actions and informations will be logged by the SDK and visible in the logcat in Android Studio (tag "SDK").
<code>static boolean DISPLAY_ERROR_MESSAGE</code>		Deprecated. <i>This value is of no use anymore</i>
<code>static boolean IS_LOCATIONTRACKER_ACTIF</code>		Deprecated. <i>This is not used</i>
<code>static java.lang.Class<? extends android.app.Activity> MAIN_ACTIVITY</code>		The main activity of your app.
<code>static float MIN_DISTANCE_CHANGE_FOR_UPDATES</code>		Deprecated. <i>This is not used</i>
<code>static long MIN_TIME_BW_UPDATES</code>		Deprecated. <i>This is not used</i>
<code>static java.lang.Class<? extends android.app.Activity> NOTIFICATION_ACTIVITY</code>		The activity that will be called when opening a notification.
<code>static java.lang.String VERSION_LIB</code>		The version of the Selligent Mobile SDK library

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods	Deprecated Methods	Description
Modifier and Type	Method				
<code>boolean areInAppMessagesEnabled()</code>					This method tells if the management of In App messages is enabled or not
<code>boolean areNotificationEnabled()</code>					This method tells if the reception of notifications is enabled or not
<code>boolean areNotificationsAllowedByOS()</code>					This method tells if the notifications are allowed at OS level
<code>void checkAndDisplayMessage (android.content.Intent intent, android.content.Context context)</code>					This method will check if there is a message to be displayed in the given intent and will display it.
<code>void checkAndDisplayMessage (android.content.Intent intent, android.content.Context context, SMInAppMessageDisplay inAppMessageDisplay)</code>					This method will check if there is a message to be displayed in the given intent and will display it.
<code>void deleteInAppMessage (java.lang.String messageId)</code>					This method deletes one In-App Message.
<code>void deleteInAppMessages (java.lang.String[] messageIds)</code>					This method deletes several In-App Messages at once.
<code>void disableGeolocation()</code>					Disables the geolocation functionality.
<code>void disableInAppMessages()</code>					This method disables the management of "in app" messages.
<code>void disableNotifications()</code>					This method disables the reception of push notifications from the server.
<code>void displayLastReceivedRemotePushNotification (android.app.Activity activity)</code>					Display the latest received remote notification This is mostly used in conjunction with RemoteMessageDisplayType set to None
<code>void displayMessage (java.lang.String messageId, android.app.Activity activity)</code>					Display the given message (remote or In App) whose id is given, like if clicking on the notification
<code>void displayNotification (android.content.Context context, android.content.Intent intent)</code>					This method is to be used only if you set <code>SMSettings.DoNotListenToThePush</code> to true.
<code>void enableGeolocation()</code>					Enables the geolocation functionality.
<code>void enableInAppMessages (SMInAppRefreshType refreshType)</code>					This method enables the management of "in app" messages.

void	<code>enableNotifications()</code>	This method enables the reception of push notifications from the server.
void	<code>executeButtonAction (android.content.Context context, SMNotificationButton button, SMInAppMessage message)</code>	This method will execute the action attached to the given <code>SMNotificationButton</code> of the given <code>SMInAppMessage</code> .
void	<code>executeLinkAction (android.content.Context context, SMLink link, SMInAppContent content)</code>	This method will execute the action attached to the given <code>SMLink</code> of the given <code>SMInAppContent</code> .
java.lang.String	<code>getDeviceId()</code>	This methods returns the device id stored by the SDK after registering to the Selligent Mobile platform.
java.lang.String	<code>getGCMToken()</code>	This methods returns the GCM token stored by the SDK after calling <code>registerDevice (Context)</code> .
java.util.ArrayList< <code>SMInAppContent</code> >	<code>getInAppContents (java.lang.String category, SMContentType type, int max)</code>	Gets the list of valid <code>SMInAppContent</code> for the given type and category.
void	<code>getInAppContents (java.lang.String category, SMContentType type, int max, SMInAppContentReturn callbackEvent)</code>	Gets the list of valid <code>SMInAppContent</code> for the given type and category.
void	<code>getInAppMessages (SMInAppMessageReturn callbackEvent)</code>	Gets the list of all <code>SMInAppMessage</code> currently stored by the SDK, unfiltered.
static <code>SMManager</code>	<code>getInstance()</code>	
java.util.HashMap<java.lang.String,java.lang.String>	<code>getLastRemotePushNotification()</code>	Get the id and the title of the latest received remote notification This is mostly used in conjunction with <code>RemoteMessageDisplayType</code> set to None
int	<code>getNotificationIconColor()</code>	Gets the icon color set by <code>setNotificationIconColor</code> .
android.graphics.Bitmap	<code>getNotificationLargeIcon()</code>	Gets the icon set by <code>setNotificationLargeIcon</code> .
int	<code>getNotificationSmallIcon()</code>	This returns the resource id set by <code>setNotificationSmallIcon</code> .
<code>SMServerManager</code>	<code>getObserverManager()</code>	This method will return the instance of <code>SMServerManager</code> needed for you to listen to the SDK events.
<code>SMRemoteMessageDisplayType</code>	<code>getRemoteMessagesDisplayType()</code>	Gets the way remote messages are displayed when the app is in foreground.
boolean	<code>isGeolocationEnabled()</code>	Tells if the geolocation is enabled or not.
void	<code>registerDevice (android.content.Context context)</code>	Deprecated. <i>Use the json file given by the Firebase Console when activating Cloud Messaging on your project.</i>
void	<code>reload (SMSecureSettings settings, android.app.Activity activity)</code>	Deprecated. <i>Use <code>reload(SMSecureSettings, SMCallback)</code></i>
void	<code>reload (SMSecureSettings settings, SMCallback callback)</code>	This method is used in the special case of the Selligent demo app Parana and should not be needed.
void	<code>retrieveNotificationMessage (android.content.Intent intent, OnSMNotificationMessageRetrieved notificationMessageRetrieved)</code>	It will trigger the <code>OnSMNotificationMessageRetrieved.onSuccess (SMNotificationMessage)</code> when the <code>SMNotificationMessage</code> is retrieved from the intent.
void	<code>sendDeviceInfos()</code>	Deprecated.
void	<code>sendDeviceInfos (SMDeviceInfos deviceInfos)</code>	This method will send a "SetInfo" event to the platform ONLY if the properties of the <code>SMDeviceInfos</code> object changed since the last call
void	<code>sendEvent (SMEvent event)</code>	Deprecated. <i>use <code>sendSMEEvent (com.selligent.sdk.SMEvent)</code> instead</i>
void	<code>sendSMEEvent (SMEvent event)</code>	Use this method to send an event to the Selligent platform.
void	<code>setApplication (android.app.Application app)</code>	This gives to the SDK a pointer to the Application instance.
void	<code>setButtonAsClicked (SMNotificationMessage notificationMessage, SMNotificationButton notificationButton)</code>	It sends a ButtonClick event to the Selligent Mobile Platform.

void	<code>setFirebaseToken (java.lang.String token)</code>	This method is to be used only if you set <code>SMSettings.DoNotFetchTheToken</code> to true.
void	<code>setInAppContentAsSeen (SMInAppContent inAppContent)</code>	Set the given <code>SMInAppContent</code> as seen.
void	<code>setInAppMessageAsSeen (SMInAppMessage inAppMessage)</code>	Set the given <code>SMInAppMessage</code> as seen.
void	<code>setInAppMessageAsUnseen (SMInAppMessage inAppMessage)</code>	Set the given <code>SMInAppMessage</code> as unseen.
void	<code>setNotificationCallback (SMNotificationCallback callback)</code>	Deprecated. <i>Since 1.3, a broadcast is performed at the click of a button (cf.</i>
void	<code>setNotificationIconColor (int argb)</code>	This allows the SDK to set a specific color to the icon for the notifications.
void	<code>setNotificationLargeIcon (int iconResource)</code>	This allows the SDK to use a specific icon for the notifications.
void	<code>setNotificationMessageAsReceived (SMNotificationMessage notificationMessage)</code>	It sends a PushReceived event to the Selligent Mobile Platform.
void	<code>setNotificationMessageAsSeen (SMNotificationMessage notificationMessage)</code>	It sends a PushOpened event to the Selligent Mobile Platform.
void	<code>setNotificationSmallIcon (int iconResource)</code>	This allows the SDK to use a specific icon for the notifications.
void	<code>start (SMSettings settings)</code>	Mandatory method used to setup the Selligent Mobile SDK.
void	<code>start (SMSettings settings, android.app.Application application)</code>	Mandatory method used to setup the Selligent Mobile SDK.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

BROADCAST_EVENT_RECEIVED_REMOTE_NOTIFICATION

@Deprecated

public static final java.lang.String BROADCAST_EVENT_RECEIVED_REMOTE_NOTIFICATION

Deprecated.

Since listening to broadcast in background is no longer possible under Android O, this broadcast is now deprecated (it is still sent though, so if you don't target Android O, you will still be able to listen to it).

String representing a broadcast name you can listen to. It is broadcast shortly after receiving a remote-notification Primary-application may use this notification to decide when to display any remote-notification This broadcast can be sent while the app is in background and, therefore, is sent using a normal `Context.sendBroadcast(Intent)` To listen to this broadcast, you also have to set the package name of your app as a category to your `IntentFilter`

Since:

1.3

See Also:

`Context.sendBroadcast(Intent), IntentFilter, Constant Field Values`

BROADCAST_EVENT_RECEIVED_IN_APP_MESSAGE

@Deprecated

public static final java.lang.String BROADCAST_EVENT_RECEIVED_IN_APP_MESSAGE

Deprecated.

LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use `SMObserverManager.observeInAppMessages(LifecycleOwner, Observer)` instead.

String representing a broadcast name you can listen to. It is broadcasted shortly after receiving InApp messages Primary-application may use this notification to manage the received InApp messages This broadcast is sent locally using `LocalBroadcastManager.sendBroadcast(Intent)`

Since:

1.3

See Also:

`LocalBroadcastManager.sendBroadcast(Intent), Constant Field Values`

BROADCAST_EVENT_RECEIVED_IN_APP_CONTENTS

@Deprecated
public static final java.lang.String BROADCAST_EVENT_RECEIVED_IN_APP_CONTENTS

Deprecated.

LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use SMOObserverManager.observeInAppContents(LifecycleOwner, Observer) instead.

String representing a broadcast name you can listen to. It is broadcasted shortly after receiving InApp contents Primary-application may use this notification to manage the received InApp contents This broadcast is sent locally using LocalBroadcastManager.sendBroadcast(Intent)

Since:

1.4

See Also:

LocalBroadcastManager.sendBroadcast(Intent), Constant Field Values

BROADCAST_EVENT_BUTTON_CLICKED

@Deprecated
public static final java.lang.String BROADCAST_EVENT_BUTTON_CLICKED

Deprecated.

LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use SMOObserverManager.observeClickedButton(LifecycleOwner, Observer) instead.

String representing a broadcast name you can listen to. It is broadcasted when the user interacts with a remote-notification Usefull to retrieve user's actions on a received remote-notification. This broadcast is sent locally using LocalBroadcastManager.sendBroadcast(Intent)

Since:

1.3

See Also:

LocalBroadcastManager.sendBroadcast(Intent), Constant Field Values

BROADCAST_EVENT_WILL_DISPLAY_NOTIFICATION

@Deprecated
public static final java.lang.String BROADCAST_EVENT_WILL_DISPLAY_NOTIFICATION

Deprecated.

LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use SMOObserverManager.observeDisplayedMessage(LifecycleOwner, Observer) instead.

String representing a broadcast name you can listen to. It is broadcasted shortly before displaying a remote-notification Primary-application may use this broadcast to pause any ongoing work before the remote-notification is displayed. This broadcast is also triggered even if you disable shouldDisplayRemoteNotification (see SMSSettings). This broadcast is sent locally using LocalBroadcastManager.sendBroadcast(Intent)

Since:

1.3

See Also:

LocalBroadcastManager.sendBroadcast(Intent), BROADCAST_EVENT_WILL_DISMISS_NOTIFICATION, Constant Field Values

BROADCAST_EVENT_WILL_DISMISS_NOTIFICATION

@Deprecated
public static final java.lang.String BROADCAST_EVENT_WILL_DISMISS_NOTIFICATION

Deprecated.

LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use SMOObserverManager.observeDismissedMessage(LifecycleOwner, Observer) instead.

String representing a broadcast name you can listen to. It is broadcasted shortly before dismissing the current remote-notification Primary-application may use this broadcast to resume any paused work. (see BROADCAST_EVENT_WILL_DISPLAY_NOTIFICATION) This broadcast is sent locally using LocalBroadcastManager.sendBroadcast(Intent)

Since:

1.3

See Also:

LocalBroadcastManager.sendBroadcast(Intent), BROADCAST_EVENT_WILL_DISPLAY_NOTIFICATION, Constant Field Values

BROADCAST_EVENT_RECEIVED_GCM_TOKEN

@Deprecated

```
public static final java.lang.String BROADCAST_EVENT_RECEIVED_GCM_TOKEN
```

Deprecated.

LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use SMObservableManager.observeToken(LifecycleOwner, Observer) instead.

String representing a broadcast name you can listen to. It is broadcasted after receiving the GCM token and only if it changed. Primary-application may use this broadcast to retrieve the GCM token. This broadcast is sent locally using LocalBroadcastManager.sendBroadcast(Intent)

Since:

1.3

See Also:

```
LocalBroadcastManager.sendBroadcast(Intent), Constant Field Values
```

BROADCAST_DATA_IN_APP_MESSAGES

@Deprecated

```
public static final java.lang.String BROADCAST_DATA_IN_APP_MESSAGES
```

Deprecated.

LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use SMObservableManager.observeInAppMessages(LifecycleOwner, Observer) instead.

String representing a key to retrieve an object inside an intent Use this key to retrieve an array of SMInAppMessage from the intent received from the broadcast BROADCAST_EVENT_RECEIVED_IN_APP_MESSAGE. This broadcast is sent locally using LocalBroadcastManager.sendBroadcast(Intent)

Since:

1.3

See Also:

```
LocalBroadcastManager.sendBroadcast(Intent), SMInAppMessage, BROADCAST_EVENT_RECEIVED_IN_APP_MESSAGE, Constant Field Values
```

BROADCAST_DATA_IN_APP_CONTENTS

@Deprecated

```
public static final java.lang.String BROADCAST_DATA_IN_APP_CONTENTS
```

Deprecated.

LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use SMObservableManager.observeInAppContents(LifecycleOwner, Observer) instead.

String representing a key to retrieve an object inside an intent Use this key to retrieve an dictionary containing the number of contents for each category from the broadcast BROADCAST_EVENT_RECEIVED_IN_APP_MESSAGE. This broadcast is sent locally using LocalBroadcastManager.sendBroadcast(Intent)

Since:

1.4

See Also:

```
LocalBroadcastManager.sendBroadcast(Intent), BROADCAST_EVENT_RECEIVED_IN_APP_CONTENTS, Constant Field Values
```

BROADCAST_DATA_BUTTON

@Deprecated

```
public static final java.lang.String BROADCAST_DATA_BUTTON
```

Deprecated.

LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use SMObservableManager.observeClickedButton(LifecycleOwner, Observer) instead.

String representing a key to retrieve an object inside an intent Use this key to retrieve the object SMNotificationButton from the intent received from the broadcast BROADCAST_EVENT_BUTTON_CLICKED. This broadcast is sent locally using LocalBroadcastManager.sendBroadcast(Intent)

Since:

1.3

See Also:

```
LocalBroadcastManager.sendBroadcast(Intent), SMNotificationButton, BROADCAST_EVENT_BUTTON_CLICKED, Constant Field Values
```

BROADCAST_DATA_GCM_TOKEN

@Deprecated

```
public static final java.lang.String BROADCAST_DATA_GCM_TOKEN
```

Deprecated.

LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use SMObservableManager.observeToken(LifecycleOwner, Observer) instead.

String representing a key to retrieve an object inside an intent Use this key to retrieve the GCM token from the intent received from the broadcast BROADCAST_EVENT_RECEIVED_GCM_TOKEN. This broadcast is sent locally using LocalBroadcastManager.sendBroadcast(Intent)

Since:

1.3

See Also:

```
LocalBroadcastManager.sendBroadcast(Intent), BROADCAST_EVENT_RECEIVED_GCM_TOKEN, Constant Field Values
```

MIN_TIME_BW_UPDATES

@Deprecated

```
public static long MIN_TIME_BW_UPDATES
```

Deprecated.

This is not used

MIN_DISTANCE_CHANGE_FOR_UPDATES

@Deprecated

```
public static float MIN_DISTANCE_CHANGE_FOR_UPDATES
```

Deprecated.

This is not used

IS_LOCATIONTRACKER_ACTIF

@Deprecated

```
public static boolean IS_LOCATIONTRACKER_ACTIF
```

Deprecated.

This is not used

DISPLAY_ERROR_MESSAGE

@Deprecated

```
public static boolean DISPLAY_ERROR_MESSAGE
```

Deprecated.

This value is of no use anymore

DEBUG

```
public static boolean DEBUG
```

When true, different actions and informations will be logged by the SDK and visible in the logcat in Android Studio (tag "SDK"). This is for debug only and should never be set to true in a release. Default value is false.

NOTIFICATION_ACTIVITY

```
public static java.lang.Class<? extends android.app.Activity> NOTIFICATION_ACTIVITY
```

The activity that will be called when opening a notification. Default value is SMNotificationActivity

MAIN_ACTIVITY

```
public static java.lang.Class<? extends android.app.Activity> MAIN_ACTIVITY
```

The main activity of your app. The goal is to restrict the display of potential error resolution dialogs to that activity and avoid having it on a splash screen. If not defined, the dialog will be displayed on any activity.

VERSION_LIB

```
public static final java.lang.String VERSION_LIB
```

The version of the Selligent Mobile SDK library

See Also:

[Constant Field Values](#)

Method Detail

getObserverManager

```
public SMObservableManager getObserverManager()
```

This method will return the instance of `SMObserverManager` needed for you to listen to the SDK events.

Returns:

the instance of `SMObserverManager`

Since:

3.0

See Also:

[SMObserverManager](#)

getInstance

```
public static SMManager getInstance()
```

Returns:

the `SMManager` instance

start

```
public void start(SMSettings settings)
```

Mandatory method used to setup the Selligent Mobile SDK. It can only be called once (usually in a class extending `SMAplication`). Any later call with different values would be ineffective. Use this version ONLY if you extend `SMAplication`

Parameters:

`settings` - an `SMSettings` object containing the Google application id, the web service URL, the Selligent client id and the Selligent private key.

See Also:

[SMSettings](#)

start

```
public void start(SMSettings settings,  
                  android.app.Application application)
```

Mandatory method used to setup the Selligent Mobile SDK. It can only be called once (usually in a class extending `SMAplication`). Any later call with different values would be ineffective. You must use this version if you do not extend `SMAplication`

Parameters:

settings - an SMSettings object containing the Google application id, the web service URL, the Selligent client id and the Selligent private key.

application - the instance of the class extending Application

Since:

1.1

See Also:

[SMSettings](#)

reload

@Deprecated

```
public void reload(SMSettings settings,  
                  android.app.Activity activity)
```

Deprecated.

Use [reload\(SMSettings, SMCallback\)](#)

This method is used in the special case of the Selligent demo app Parana and should not be needed. It allows to change the settings given at startup by the start method.

Parameters:

settings - an SMSettings object

activity - the activity calling the method

reload

```
public void reload(SMSettings settings,  
                  SMCallback callback)
```

This method is used in the special case of the Selligent demo app Parana and should not be needed. It allows to change the settings given at startup by the start method.

Parameters:

settings - an SMSettings object containing the Google application id, the web service URL, the Selligent client id and the Selligent private key.

callback - an SMCallback object to execute code after the reload is finished and the device id for the new environment is retrieved

See Also:

[SMSettings](#)

setApplication

```
public void setApplication(android.app.Application app)
```

This gives to the SDK a pointer to the Application instance. You should only call this if you do not extend SMAplication.

Parameters:

app - the Application instance

See Also:

[SMAplication](#)

getDeviceId

```
public java.lang.String getDeviceId()
```

This methods returns the device id stored by the SDK after registering to the Selligent Mobile platform. As the registration is asynchronous, the token returned may be empty if the registration process is not completed.

Returns:

the stored device id

Since:

3.6

`SMInAppMessageDisplay inAppMessageDisplay`

This method will check if there is a message to be displayed in the given intent and will display it. It has to be used in the `onCreate` (after `registerDevice(Context)`) and `onNewIntent` events if your activity does not extend `SMB BaseActivity`. This overload allows you to manage the display of the In-App Message linked to a push notification yourself

Parameters:

`intent` - the intent containing the message

`context` - the activity in which this method is called

`inAppMessageDisplay` - the `SMInAppMessageDisplay` interface instance that allows you to manage the display of an In-App Message

Since:

3.6.0

`enableNotifications`

```
public void enableNotifications()
```

This method enables the reception of push notifications from the server. It sends an "opt-in" message to the web service to allow it to send push notifications to the device.

`disableNotifications`

```
public void disableNotifications()
```

This method disables the reception of push notifications from the server. It sends an "opt-out" message to the web service to tell it to stop sending push notifications to the device. Any still received will be ignored by the SDK.

`areNotificationEnabled`

```
public boolean areNotificationEnabled()
```

This method tells if the reception of notifications is enabled or not

Returns:

true if enabled, false otherwise

`areNotificationsAllowedByOS`

```
public boolean areNotificationsAllowedByOS()
```

This method tells if the notifications are allowed at OS level

Returns:

true if allowed, false otherwise

Since:

3.7

`setNotificationSmallIcon`

```
public void setNotificationSmallIcon(int iconResource)
```

This allows the SDK to use a specific icon for the notifications. This small icon will be visible in the status bar at the top of the device.

Parameters:

`iconResource` - the resource id of the icon (`R.id.your_icon`)

`getNotificationSmallIcon`

```
public int getNotificationSmallIcon()
```

This returns the resource id set by `setNotificationSmallIcon`.

Returns:

the resource id.

See Also:

[setNotificationSmallIcon\(int\)](#)

setNotificationLargeIcon

```
public void setNotificationLargeIcon(int iconResource)
```

This allows the SDK to use a specific icon for the notifications. This large icon will be visible in the notification view.

Parameters:

iconResource - the resource id of the icon (`R.id.your_icon`)

getNotificationLargeIcon

```
public android.graphics.Bitmap getNotificationLargeIcon()
```

Gets the icon set by `setNotificationLargeIcon`.

Returns:

the `Bitmap` of the icon

See Also:

[setNotificationLargeIcon\(int\)](#)

setNotificationIconColor

```
public void setNotificationIconColor(int argb)
```

This allows the SDK to set a specific color to the icon for the notifications.

Parameters:

argb - int representing color : (`ResourcesCompat.getColor(getResources(), R.color.color_id, null)`) or `Color.parseColor("#AARRGGBB")`

getNotificationIconColor

```
public int getNotificationIconColor()
```

Gets the icon color set by `setNotificationIconColor`.

Returns:

the resource int value of the color

setNotificationCallback

@Deprecated

```
public void setNotificationCallback(SMNotificationCallback callback)
```

Deprecated.

Since 1.3, a broadcast is performed at the click of a button (cf. `BROADCAST_EVENT_BUTTON_CLICKED`)

Sets the callback object containing the code that will be executed whenever a button received in a push notification is clicked

Parameters:

callback - a `SMNotificationCallback` object

See Also:

[SMNotificationCallback](#)

displayLastReceivedRemotePushNotification

```
public void displayLastReceivedRemotePushNotification(android.app.Activity activity)
```

Display the latest received remote notification This is mostly used in conjunction with RemoteMessageDisplayType set to None

Parameters:

activity - the current activity

Since:

1.3

getLastRemotePushNotification

```
public java.util.HashMap<java.lang.String,java.lang.String> getLastRemotePushNotification()
```

Get the id and the title of the latest received remote notification This is mostly used in conjunction with RemoteMessageDisplayType set to None

Returns:

a HashMap containing the "id" and "title" of the last remote notification

Since:

1.3

getRemoteMessagesDisplayType

```
public SMRemoteMessageDisplayType getRemoteMessagesDisplayType()
```

Gets the way remote messages are displayed when the app is in foreground. If Automatic, no notification will be created, the message will be displayed right away. If Notification, a notification will be created and the message will be displayed only after clicking on it. If None, nothing will happen, the app will need to manage the display of the message (cf. `getLastRemotePushNotification()`, `displayLastReceivedRemotePushNotification(android.app.Activity)`)

Returns:

the SMRemoteMessageDisplayType

Since:

1.3

displayNotification

```
public void displayNotification(android.content.Context context,  
                               android.content.Intent intent)
```

This method is to be used only if you set `SMSettings.DoNotListenToThePush` to true. It will create a notification based on the information present in the intent and will display it. If the app is in foreground and the setting `SMSettings.RemoteMessageDisplayType` is set to 'Automatic', the SDK will display the in-app message linked to the push right away without going through a notification.

Parameters:

context -

intent - it contains all the push information. If you retrieve the push using the `FirebaseMessagingService` method `onMessageReceived`, simply call `toIntent()` on the `RemoteMessage` object received.

Since:

2.1.0

retrieveNotificationMessage

```
public void retrieveNotificationMessage(@NonNull  
                                       android.content.Intent intent,  
                                       @NonNull  
                                       OnSMNotificationMessageRetrieved notificationMessageRetrieved)
```

It will trigger the `OnSMNotificationMessageRetrieved.onSuccess(SMNotificationMessage)` when the `SMNotificationMessage` is retrieved from the intent. If there is none in the intent, neither `onSuccess` nor `onError` will be triggered. If, for some reason, an error occurs, `onError` will be triggered.

Parameters:

intent - the Intent containing the push

notificationMessageRetrieved - the callback that will be triggered after retrieving the SMNotificationMessage

Since:

3.7

See Also:

[OnSMNotificationMessageRetrieved](#)

setNotificationMessageAsReceived

```
public void setNotificationMessageAsReceived(@NonNull  
    SMNotificationMessage notificationMessage)
```

It sends a PushReceived event to the Selligent Mobile Platform. Make sure this is only called once per push as it will send the event everytime.

Parameters:

notificationMessage - the SMNotificationMessage to mark as received

Since:

3.7

See Also:

[SMNotificationMessage](#)

setNotificationMessageAsSeen

```
public void setNotificationMessageAsSeen(@NonNull  
    SMNotificationMessage notificationMessage)
```

It sends a PushOpened event to the Selligent Mobile Platform. It will send the event everytime the method is called.

Parameters:

notificationMessage - the SMNotificationMessage to mark as seen

Since:

3.7

See Also:

[SMNotificationMessage](#)

setButtonAsClicked

```
public void setButtonAsClicked(@NonNull  
    SMNotificationMessage notificationMessage,  
    @NonNull  
    SMNotificationButton notificationButton)
```

It sends a ButtonClick event to the Selligent Mobile Platform. It will send the event everytime the method is called.

Parameters:

notificationMessage - the SMNotificationMessage containing the button

notificationButton - the SMNotificationButton that was clicked

Since:

3.7

See Also:

[SMNotificationMessage](#), [SMNotificationButton](#)

enableInAppMessages

```
public void enableInAppMessages(SMInAppRefreshType refreshType)
```

This method enables the management of "in app" messages.

Parameters:

refreshType - How often the SDK will fetch the new messages.

Since:

1.3

areInAppMessagesEnabled

```
public boolean areInAppMessagesEnabled()
```

This method tells if the management of In App messages is enabled or not

Returns:

true if enabled, false otherwise

Since:

1.3

disableInAppMessages

```
public void disableInAppMessages()
```

This method disables the management of "in app" messages.

Since:

1.3

displayMessage

```
public void displayMessage(java.lang.String messageId,  
                           android.app.Activity activity)
```

Display the given message (remote or In App) whose id is given, like if clicking on the notification

Parameters:

messageId - the string id of the message to display

activity - the current activity

Since:

1.3

getInAppMessages

```
public void getInAppMessages(SMInAppMessageReturn callbackEvent)
```

Gets the list of all `SMInAppMessage` currently stored by the SDK, unfiltered. Use this method if you want to display the In-App messages yourself. Note: the messages retrieved using this method are the exact content of the In-App message cache. If you started the SDK with the value None for `SMSettings.ClearCacheIntervalValue`, it means there is no cache for the In-App messages and, therefore, this will return an empty array. In that case, you must rely only on the observer to retrieve the In-App messages.

Parameters:

callbackEvent - The event that will be called when the messages are retrieved. It will be passed an `ArrayList` of `SMInAppMessage`

Since:

3.4.0

setInAppMessageAsSeen

```
public void setInAppMessageAsSeen(SMInAppMessage inAppMessage)
```

Set the given `SMInAppMessage` as seen. It will also send an event to the platform to inform it. Use this method if you want to display the In-App messages yourself.

Parameters:

inAppMessage - the SMInAppMessage

Since:

3.4

setInAppMessageAsUnseen

```
public void setInAppMessageAsUnseen(SMInAppMessage inAppMessage)
```

Set the given [SMInAppMessage](#) as unseen. Use this method if you want to display the In-App messages yourself.

Parameters:

inAppMessage - the SMInAppMessage

Since:

3.5

executeButtonAction

```
public void executeButtonAction(android.content.Context context,  
                               SMNotificationButton button,  
                               SMInAppMessage message)
```

This method will execute the action attached to the given [SMNotificationButton](#) of the given [SMInAppMessage](#). It will also send an event to the platform to inform the button was clicked. Use this method if you do not want to implement our Fragments.

Parameters:

context - the current Activity.

button - the [SMNotificationButton](#) containing the action to execute.

message - the corresponding [SMInAppMessage](#).

Since:

3.4

deleteInAppMessage

```
public void deleteInAppMessage(java.lang.String messageId)
```

This method deletes one In-App Message.

Parameters:

messageId - The id of the message

Since:

3.5

deleteInAppMessages

```
public void deleteInAppMessages(java.lang.String[] messageIds)
```

This method deletes several In-App Messages at once.

Parameters:

messageIds - an array containing the ids of the messages

Since:

3.5

getInAppContents

```
public java.util.ArrayList<SMInAppContent> getInAppContents (java.lang.String category,
    SMContentType type,
    int max)
```

Gets the list of valid `SMInAppContent` for the given type and category. Use this method if you do not want to implement our Fragments.

Parameters:

`category` - The category of the `SMInAppContent`.

`type` - The `SMContentType` of the `SMInAppContent`.

`max` - The number of contents to get. -1 to get them all.

Returns:

An `ArrayList` of `SMInAppContent`.

Since:

1.4 Warning: if the cache is loaded asynchronously, there might not be any content yet when you execute this call. In order to be sure to retrieve them, call `getInAppContents(String, SMContentType, int, SMInAppContentReturn)`

getInAppContents

```
public void getInAppContents(java.lang.String category,
    SMContentType type,
    int max,
    SMInAppContentReturn callbackEvent)
```

Gets the list of valid `SMInAppContent` for the given type and category. Use this method if you do not want to implement our Fragments.

Parameters:

`category` - The category of the `SMInAppContent`.

`type` - The `SMContentType` of the `SMInAppContent`.

`max` - The number of contents to get. -1 to get them all.

`callbackEvent` - The event that will be called when the contents are retrieved. It will be passed an `ArrayList` of `SMInAppContent`

Since:

2.1.0

setInAppContentAsSeen

```
public void setInAppContentAsSeen(SMInAppContent inAppContent)
```

Set the given `SMInAppContent` as seen. It will also send an event to the platform to inform it. Use this method if you do not want to implement our Fragments.

Parameters:

`inAppContent` - the `SMInAppContent`

Since:

1.4

executeLinkAction

```
public void executeLinkAction(android.content.Context context,
    SMLink link,
    SMInAppContent content)
```

This method will execute the action attached to the given `SMLink` of the given `SMInAppContent`. It will also send an event to the platform to inform the link was clicked. Use this method if you do not want to implement our Fragments.

Parameters:

`context` - the current Activity.

`link` - the `SMLink` containing the action to execute.

`content` - the corresponding `SMInAppContent`.

Since:

enableGeolocation

```
public void enableGeolocation()
```

Enables the geolocation functionality. It will be enabled until `disableGeolocation()` is called. It will keep going even if the app or the device is restarted. The goal is to provide users with an opt-in. **WARNING:** this method should only be called if "enableOnFirstRun" is set to "false" in the `plotconfig.json` file. **With the default configuration, you do not have to call it.**

Since:

1.7

disableGeolocation

```
public void disableGeolocation()
```

Disables the geolocation functionality. No geolocation related notification will be sent anymore to the user. It will be disabled until `enableGeolocation()` is called even if the app or the device is restarted. The goal is to provide users with an opt-out.

Since:

1.7

isGeolocationEnabled

```
public boolean isGeolocationEnabled()
```

Tells if the geolocation is enabled or not.

Since:

1.7

sendEvent

@Deprecated

```
public void sendEvent(SMEvent event)
```

Deprecated.

use `sendSMEEvent(com.selligent.sdk.SMEvent)` instead

Use this method to send an event to the Selligent platform.

Parameters:

event - an SMEvent object

See Also:

`SMEEvent`, `SMEEventUserLogin`, `SMEEventUserLogout`, `SMEEventUserRegister`, `SMEEventUserUnregister`

sendSMEEvent

```
public void sendSMEEvent(SMEvent event)
```

Use this method to send an event to the Selligent platform. If you are sending a simple `SMEvent`, then we will check if the values are different from the last time you sent them. If they are not, the event won't be sent.

Parameters:

event - an SMEvent object

Since:

1.2

See Also:

`SMEEvent`, `SMEEventUserLogin`, `SMEEventUserLogout`, `SMEEventUserRegister`, `SMEEventUserUnregister`

sendDeviceInfos

```
@Deprecated  
public void sendDeviceInfos()
```

Deprecated.

This method is deprecated as of 1.6 and does not do anything anymore. Use `sendDeviceInfos(SMDeviceInfos)` instead

Since:

1.4.1

sendDeviceInfos

```
public void sendDeviceInfos(SMDeviceInfos deviceInfos)
```

This method will send a "SetInfo" event to the platform ONLY if the properties of the `SMDeviceInfos` object changed since the last call

Parameters:

`deviceInfos` - Wrapper over a few properties of the device.

Package com.selligent.sdk

Class SMMarker

java.lang.Object
com.selligent.sdk.SMMarker

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

```
public class SMMarker
extends java.lang.Object
implements java.io.Externalizable
```

This class represents a point on a map.

Since:

3.4

Version:

3.9

See Also:

[Serialized Form](#)

Constructor Summary

Constructors	
Constructor	Description
SMMarker()	

Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method		Description
java.lang.String	getDescription()		
double	getLatitude()		
double	getLongitude()		
java.lang.String	getTitle()		
void	readExternal(java.io.ObjectInput serializedObject)		
void	writeExternal(java.io.ObjectOutput serializedObject)		

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

SMMarker

public SMMarker()

Method Detail

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

Specified by:

readExternal in interface java.io.Externalizable

Throws:

java.io.IOException

java.lang.ClassNotFoundException

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

Specified by:

writeExternal in interface java.io.Externalizable

Throws:

java.io.IOException

getLatitude

public double getLatitude()

Returns:

the double representing the latitude of the marker

getLongitude

```
public double getLongitude()
```

Returns:

the double representing the longitude of the marker

getDescription

```
public java.lang.String getDescription()
```

Returns:

the description of the marker

getTitle

```
public java.lang.String getTitle()
```

Returns:

the title of the marker

Package com.selligent.sdk

Class SMNotificationButton

java.lang.Object
com.selligent.sdk.SMNotificationButton

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

Direct Known Subclasses:

SMLink

```
public class SMNotificationButton
extends java.lang.Object
implements java.io.Externalizable
```

Object containing all the data used to implement a button in a notification/message.

Since:

1.3

Version:

3.9

See Also:

Serialized Form

Field Summary

Fields

Modifier and Type	Field	Description
int	action	
java.util.Hashtable<java.lang.String,java.lang.String>	data	
java.lang.String	id	
java.lang.String	label	
int	type	
java.lang.String	value	

Constructor Summary

Constructors

Constructor	Description
-------------	-------------

Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method	Description	
int	<code>getAction()</code>		The action that will be executed when clicking on the button.
java.lang.String	<code>getId()</code>		
java.lang.String	<code>getLabel()</code>		
java.lang.String	<code>getValue()</code>		
void	<code>readExternal (java.io.ObjectInput serializedObject)</code>		This method is called when deserializing the object.
void	<code>writeExternal (java.io.ObjectOutput serializedObject)</code>		This method is called when serializing the object.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

label

public java.lang.String label

id

public java.lang.String id

type

public int type

action

```
public int action
```

value

```
public java.lang.String value
```

data

```
public java.util.Hashtable<java.lang.String,java.lang.String> data
```

Constructor Detail

SMNotificationButton

```
public SMNotificationButton()
```

Method Detail

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

This method is called when deserializing the object. It should not be called manually.

Specified by:

readExternal in interface java.io.Externalizable

Parameters:

serializedObject - the ObjectInput object containing all the values of the SMNotificationButton object to recreate.

Throws:

java.io.IOException

java.lang.ClassNotFoundException

Since:

1.3

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

This method is called when serializing the object. It should not be called manually.

Specified by:

writeExternal in interface java.io.Externalizable

Parameters:

serializedObject - the ObjectOutputStream object used to store the values of the SMNotificationButton object

Throws:

java.io.IOException

Since:

1.3

getId

```
public java.lang.String getId()
```

Returns:

the string id of the button

Since:

3.4

getAction

```
public int getAction()
```

The action that will be executed when clicking on the button. Possible values are: 0 - default (does nothing, if in a dialog, closes it) 1 - Phone 2 - SMS 3 - E-Mail 4 - Open browser/Deep link 5 - Open app 6 - Open store 7 - Event 11 - Passbook 12 - Journey map

Returns:

the int representing the type of the button

Since:

3.4

getLabel

```
public java.lang.String getLabel()
```

Returns:

the label of the button

Since:

3.4

getValue

public java.lang.String getValue()

Returns:

the value of the button that will be used to perform the action

Since:

3.4

Package com.selligent.sdk

Class SMNotificationMessage

java.lang.Object
com.selligent.sdk.SMNotificationMessage

All Implemented Interfaces:

java.io.Externalizable, java.io.Serializable

```
public class SMNotificationMessage
extends java.lang.Object
```

A notification message.

Since:

3.7

Version:

3.9

See Also:

Serialized Form

Constructor Summary

Constructors

Constructor	Description
<code>SMNotificationMessage()</code>	

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
long	<code>getCreationDate()</code>	Gets the creation date of the notification
long	<code>getExpirationDate()</code>	Gets the expiration date of the notification
java.lang.String	<code>getIAMBody()</code>	Gets the body of the In-app message linked to the notification
<code>SMNotificationButton[]</code>	<code>getIAMButtons()</code>	Gets the buttons of the In-app message linked

		to the notification
SMMapMarker[]	getIAMMarkers()	Gets the list of the markers for the In-app message linked to the notification
java.lang.String	getIAMTitle()	Gets the title of the In-app message linked to the notification
SMMessageType	getIAMType()	Gets the type of the In-App message linked to the notification
java.lang.String	getId()	Gets the id of the notification
SMNotificationButton	getMainAction()	Gets the main action of the notification
java.lang.String	getMediaType()	Gets the type of the media to display in the notification.
java.lang.String	getMediaUrl()	Gets the URL of the media to display in the notification
java.lang.String	getNotificationBody()	Gets the body of the notification
SMNotificationButton[]	getNotificationButtons()	Gets the buttons of the notification
java.lang.String	getNotificationTitle()	Gets the title of the notification
long	getReceptionDate()	Gets the reception date of the notification
void	readExternal (java.io.ObjectInput serializedObject)	
void	writeExternal (java.io.ObjectOutput serializedObject)	

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

SMNotificationMessage

```
public SMNotificationMessage()
```

Method Detail

getId

```
public java.lang.String getId()
```

Gets the id of the notification

Returns:

the String representing the id

Since:

3.7

getNotificationTitle

```
public java.lang.String getNotificationTitle()
```

Gets the title of the notification

Returns:

the String representing the title

Since:

3.7

getNotificationBody

```
public java.lang.String getNotificationBody()
```

Gets the body of the notification

Returns:

the String representing the body

Since:

3.7

getNotificationButtons

```
public SMNotificationButton[] getNotificationButtons()
```

Gets the buttons of the notification

Returns:

an array of SMLink

Since:

3.7

getMainAction

```
public SMNotificationButton getMainAction()
```

Gets the main action of the notification

Returns:

an SMNotificationButton representing the main action

Since:

3.7

getMediaType

```
public java.lang.String getMediaType()
```

Gets the type of the media to display in the notification. At the moment, only "3" (image) is supported

Returns:

the string representing the media type

Since:

3.7

getMediaUrl

```
public java.lang.String getMediaUrl()
```

Gets the URL of the media to display in the notification

Returns:

the URL

Since:

3.7

getIAMTitle

```
public java.lang.String getIAMTitle()
```

Gets the title of the In-app message linked to the notification

Returns:

the String representing the title

Since:

3.7

getIAMBody

```
public java.lang.String getIAMBody()
```

Gets the body of the In-app message linked to the notification

Returns:

the String representing the body

Since:

3.7

getIAMButtons

```
public SMNotificationButton[] getIAMButtons()
```

Gets the buttons of the In-app message linked to the notification

Returns:

an array of SMLink

Since:

3.7

getIAMMarkers

```
public SMMapMarker[] getIAMMarkers()
```

Gets the list of the markers for the In-app message linked to the notification

Returns:

an array of SMMapMarker

Since:

3.7

getIAMType

```
public SMMessageType getIAMType()
```

Gets the type of the In-App message linked to the notification

Returns:

the `SMMimeType`

Since:

3.7

getReceptionDate

`public long getReceptionDate()`

Gets the reception date of the notification

Returns:

the long representing the reception date in milliseconds since 01/01/1970

Since:

3.7

getCreationDate

`public long getCreationDate()`

Gets the creation date of the notification

Returns:

the long representing the creation date in milliseconds since 01/01/1970

Since:

3.7

getExpirationDate

`public long getExpirationDate()`

Gets the expiration date of the notification

Returns:

the long representing the expiration date in milliseconds since 01/01/1970

Since:

3.7

readExternal

`public void readExternal(java.io.ObjectInput serializedObject)`

Specified by:

`readExternal` in interface `java.io.Externalizable`

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

Specified by:

writeExternal in interface java.io.Externalizable

Throws:

java.io.IOException

Class SMOObserverManager

```
java.lang.Object
    com.selligent.sdk.SMOObserverManager
```

```
public class SMOObserverManager
extends java.lang.Object
```

This class is used to observe some events of the SDK. The triggers are executed once when the event happens. If several observers are listening, each one will be triggered once. Do not create a new instance of this class, instead, use `SMManager.getObserverManager()`

Since:

3.0

Version:

3.9

Constructor Summary

Constructors	
Constructor	Description
<code>SMOObserverManager ()</code>	

Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method		Description
void			
		<code>observeClickedButton (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<SMNotificationButton> observer)</code>	Use this method to get notified when a button of an notification or of an In-App message is clicked or when the main action of a push is executed.
void			Use this method to get notified

```
observeClickedButton      (androidx.lifecycle.LifecycleOwner lifecycleOwner,  
    androidx.lifecycle.Observer<SMNotificationButton> observer,  
    boolean triggerEveryTime)
```

when a button of an notification or of an In-App message is clicked or when teh main action of a push is executed.

void

```
observeDeviceId      (androidx.lifecycle.LifecycleOwner lifecycleOwner,  
    androidx.lifecycle.Observer<java.lang.String> observer)
```

Use this method to get notified when the SDK receives the device id from the Selligent Mobile platform.

void

```
observeDeviceId      (androidx.lifecycle.LifecycleOwner lifecycleOwner,  
    androidx.lifecycle.Observer<java.lang.String> observer, boolean triggerEveryTime)
```

Use this method to get notified when the SDK receives the device id from the Selligent Mobile platform.

void

```
observeDismissedMessage      (androidx.lifecycle.LifecycleOwner lifecycleOwner,  
    androidx.lifecycle.Observer<java.lang.Void> observer)
```

Use this method to get notified when an In-App message is about to be dismissed.

void

```
observeDismissedMessage      (androidx.lifecycle.LifecycleOwner lifecycleOwner,  
    androidx.lifecycle.Observer<java.lang.Void> observer, boolean triggerEveryTime)
```

Use this method to get notified when an In-App message is about to be dismissed.

void	observeDisplayedMessage (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<java.lang.Void> observer)	Use this method to get notified when an In-App message is about to be displayed.
void	observeDisplayedMessage (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<java.lang.Void> observer, boolean triggerEveryTime)	Use this method to get notified when an In-App message is about to be displayed.
void	observeEvent (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<java.lang.String> observer)	Use this method to get notified when a specific event is triggered from a button in a notification, In-App message or In-app content or from the main action of a notification.
void	observeEvent (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<java.lang.String> observer, boolean triggerEveryTime)	Use this method to get notified when a specific event is triggered from a button in a notification, In-App message or In-app content or from the main action of a

		notification.
void	observeInAppContents (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<java.util.HashMap<java.lang.String,java.lang.Integer>> observer)	Use this method to get notified when In-App contents are fetched.
void	observeInAppContents (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<java.util.HashMap<java.lang.String,java.lang.Integer>> observer, boolean triggerEveryTime)	Use this method to get notified when In-App contents are fetched.
void	observeInAppMessages (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<SMInAppMessage []> observer)	Use this method to get notified when new In-App messages were fetched.
void	observeInAppMessages (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<SMInAppMessage []> observer, boolean triggerEveryTime)	Use this method to get notified when new In-App messages were fetched.
void	observeToken (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<java.lang.String> observer)	Use this method to get notified when a new Firebase token was received by the SDK.
void	observeToken (androidx.lifecycle.LifecycleOwner lifecycleOwner, androidx.lifecycle.Observer<java.lang.String> observer, boolean triggerEveryTime)	Use this method to get notified when a new Firebase token was received by the SDK.

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

SMSMObserverManager

```
public SMSMObserverManager()
```

Method Detail

observeToken

```
@MainThread  
public void observeToken(@NonNull  
    androidx.lifecycle.LifecycleOwner lifecycleOwner,  
    @NonNull  
    androidx.lifecycle.Observer<java.lang.String> observer)
```

Use this method to get notified when a new Firebase token was received by the SDK. It replaces the broadcast BROADCAST_EVENT_RECEIVED_GCM_TOKEN. It must be called on the main thread. Observing using this method will trigger the onChanged event of the observer only when the value changes after the observer is created. If there is already a value at that moment, the event will not get triggered. To change that behaviour, use the overload observeToken(LifecycleOwner, Observer, boolean)

Parameters:

lifecycleOwner - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

observer - The Observer that will listen to the event and get triggered when it happens

observeToken

```
@MainThread  
public void observeToken(@NonNull  
    androidx.lifecycle.LifecycleOwner lifecycleOwner,  
    @NonNull  
    androidx.lifecycle.Observer<java.lang.String> observer,  
    boolean triggerEveryTime)
```

Use this method to get notified when a new Firebase token was received by the SDK. It replaces the broadcast BROADCAST_EVENT_RECEIVED_GCM_TOKEN. It must be called on the main thread.

Parameters:

lifecycleOwner - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

observer - The Observer that will listen to the event and get triggered when it happens
triggerEveryTime - if true, the onChanged event of the observer will be triggered every time the observer is (re)created, as long as there is a value set. If false, it will only be triggered when the value is changed after the creation of the observer.

Since:

3.2

observeDeviceId

```
@MainThread
public void observeDeviceId(@NonNull
    androidx.lifecycle.LifecycleOwner lifecycleOwner,
    @NonNull
    androidx.lifecycle.Observer<java.lang.String> observer)
```

Use this method to get notified when the SDK receives the device id from the Selligent Mobile platform. It must be called on the main thread. Observing using this method will trigger the onChanged event of the observer only when the value changes after the observer is created. If there is already a value at that moment, the event will not get triggered. To change that behaviour, use the overload `observeToken(LifecycleOwner, Observer, boolean)`

Parameters:

lifecycleOwner - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.
observer - The Observer that will listen to the event and get triggered when it happens

observeDeviceId

```
@MainThread
public void observeDeviceId(@NonNull
    androidx.lifecycle.LifecycleOwner lifecycleOwner,
    @NonNull
    androidx.lifecycle.Observer<java.lang.String> observer,
    boolean triggerEveryTime)
```

Use this method to get notified when the SDK receives the device id from the Selligent Mobile platform. It must be called on the main thread.

Parameters:

lifecycleOwner - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.
observer - The Observer that will listen to the event and get triggered when it happens
triggerEveryTime - if true, the onChanged event of the observer will be triggered every time the observer is (re)created, as long as there is a value set. If false, it will only be triggered when the value is changed after the creation of the observer.

Since:

3.6

observeInAppMessages

```
@MainThread  
public void observeInAppMessages(@NonNull  
        androidx.lifecycle.LifecycleOwner lifecycleOwner,  
        @NonNull  
        androidx.lifecycle.Observer<SMInAppMessage[]> observer)
```

Use this method to get notified when new In-App messages were fetched. It replaces the broadcast BROADCAST_EVENT_RECEIVED_IN_APP_MESSAGE. It must be called on the main thread. Observing using this method will trigger the onChanged event of the observer only when the value changes after the observer is created. If there is already a value at that moment, the event will not get triggered. To change that behaviour, use the overload observeInAppMessages(LifecycleOwner, Observer, boolean)

Parameters:

lifecycleOwner - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

observer - The Observer that will listen to the event and get triggered when it happens

observeInAppMessages

```
@MainThread  
public void observeInAppMessages(@NonNull  
        androidx.lifecycle.LifecycleOwner lifecycleOwner,  
        @NonNull  
        androidx.lifecycle.Observer<SMInAppMessage[]> observer,  
        boolean triggerEveryTime)
```

Use this method to get notified when new In-App messages were fetched. It replaces the broadcast BROADCAST_EVENT_RECEIVED_IN_APP_MESSAGE. It must be called on the main thread.

Parameters:

lifecycleOwner - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

observer - The Observer that will listen to the event and get triggered when it happens

triggerEveryTime - if true, the onChanged event of the observer will be triggered every time the observer is (re)created, as long as there is a value set. If false, it will only be triggered when the value is changed after the creation of the observer.

Since:

3.2

observeInAppContents

```
@MainThread  
public void observeInAppContents(@NonNull  
        androidx.lifecycle.LifecycleOwner lifecycleOwner,  
        @NonNull  
        androidx.lifecycle.Observer<java.util.HashMap<java.lang.String,java.lang.Integer>> observer)
```

Use this method to get notified when In-App contents are fetched. It replaces the broadcast

BROADCAST_EVENT_RECEIVED_IN_APP_CONTENTS. It must be called on the main thread. Observing using this method will trigger the onChanged event of the observer only when the value changes after the observer is created. If there is already a value at that moment, the event will not get triggered. To change that behaviour, use the overload `observeInAppContents(LifecycleOwner, Observer, boolean)`

Parameters:

`lifecycleOwner` - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

`observer` - The Observer that will listen to the event and get triggered when it happens

observeInAppContents

`@MainThread`

```
public void observeInAppContents(@NonNull  
        androidx.lifecycle.LifecycleOwner lifecycleOwner,  
        @NonNull  
        androidx.lifecycle.Observer<java.util.HashMap<java.lang.String,java.lang.Integer>> observer,  
        boolean triggerEveryTime)
```

Use this method to get notified when In-App contents are fetched. It replaces the broadcast

BROADCAST_EVENT_RECEIVED_IN_APP_CONTENTS. It must be called on the main thread.

Parameters:

`lifecycleOwner` - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

`observer` - The Observer that will listen to the event and get triggered when it happens

`triggerEveryTime` - if true, the onChanged event of the observer will be triggered every time the observer is (re)created, as long as there is a value set. If false, it will only be triggered when the value is changed after the creation of the observer.

Since:

3.2

observeClickedButton

`@MainThread`

```
public void observeClickedButton(@NonNull  
        androidx.lifecycle.LifecycleOwner lifecycleOwner,  
        @NonNull  
        androidx.lifecycle.Observer<SMNotificationButton> observer)
```

Use this method to get notified when a button of an notification or of an In-App message is clicked or when the main action of a push is executed. It replaces the broadcast BROADCAST_EVENT_BUTTON_CLICKED. It must be called on the main thread. Observing using this method will trigger the onChanged event of the observer only when the value changes after the observer is created. If there is already a value at that moment, the event will not get triggered. To change that behaviour, use the overload `observeClickedButton(LifecycleOwner, Observer, boolean)`

Parameters:

`lifecycleOwner` - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

observer - The Observer that will listen to the event and get triggered when it happens

observeClickedButton

```
@MainThread  
public void observeClickedButton(@NonNull  
        androidx.lifecycle.LifecycleOwner lifecycleOwner,  
        @NonNull  
        androidx.lifecycle.Observer<SMNotificationButton> observer,  
        boolean triggerEveryTime)
```

Use this method to get notified when a button of an notification or of an In-App message is clicked or when teh main action of a push is executed. It replaces the broadcast BROADCAST_EVENT_BUTTON_CLICKED. It must be called on the main thread.

Parameters:

lifecycleOwner - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

observer - The Observer that will listen to the event and get triggered when it happens

triggerEveryTime - if true, the onChanged event of the observer will be triggered every time the observer is (re)created, as long as there is a value set. If false, it will only be triggered when the value is changed after the creation of the observer.

Since:

3.2

observeDisplayedMessage

```
@MainThread  
public void observeDisplayedMessage(@NonNull  
        androidx.lifecycle.LifecycleOwner lifecycleOwner,  
        @NonNull  
        androidx.lifecycle.Observer<java.lang.Void> observer)
```

Use this method to get notified when an In-App message is about to be displayed. It replaces the broadcast BROADCAST_EVENT_WILL_DISPLAY_NOTIFICATION. It must be called on the main thread. Observing using this method will trigger the onChanged event of the observer only when the value changes after the observer is created. If there is already a value at that moment, the event will not get triggered. To change that behaviour, use the overload `observeDisplayedMessage(LifecycleOwner, Observer, boolean)`

Parameters:

lifecycleOwner - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

observer - The Observer that will listen to the event and get triggered when it happens

observeDisplayedMessage

```
@MainThread  
public void observeDisplayedMessage(@NonNull
```

```
        androidx.lifecycle.LifecycleOwner lifecycleOwner,  
        @NonNull  
        androidx.lifecycle.Observer<java.lang.Void> observer,  
        boolean triggerEveryTime)
```

Use this method to get notified when an In-App message is about to be displayed. It replaces the broadcast BROADCAST_EVENT_WILL_DISPLAY_NOTIFICATION. It must be called on the main thread.

Parameters:

lifecycleOwner - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

observer - The Observer that will listen to the event and get triggered when it happens

triggerEveryTime - if true, the onChanged event of the observer will be triggered every time the observer is (re)created, as long as there is a value set. If false, it will only be triggered when the value is changed after the creation of the observer.

Since:

3.2

observeDismissedMessage

```
@MainThread  
public void observeDismissedMessage(@NonNull  
        androidx.lifecycle.LifecycleOwner lifecycleOwner,  
        @NonNull  
        androidx.lifecycle.Observer<java.lang.Void> observer)
```

Use this method to get notified when an In-App message is about to be dismissed. It replaces the broadcast BROADCAST_EVENT_WILL_DISMISS_NOTIFICATION. It must be called on the main thread. Observing using this method will trigger the onChanged event of the observer only when the value changes after the observer is created. If there is already a value at that moment, the event will not get triggered. To change that behaviour, use the overload [observeDismissedMessage\(LifecycleOwner, Observer, boolean\)](#)

Parameters:

lifecycleOwner - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

observer - The Observer that will listen to the event and get triggered when it happens

observeDismissedMessage

```
@MainThread  
public void observeDismissedMessage(@NonNull  
        androidx.lifecycle.LifecycleOwner lifecycleOwner,  
        @NonNull  
        androidx.lifecycle.Observer<java.lang.Void> observer,  
        boolean triggerEveryTime)
```

Use this method to get notified when an In-App message is about to be dismissed. It replaces the broadcast BROADCAST_EVENT_WILL_DISMISS_NOTIFICATION. It must be called on the main thread.

Parameters:

`lifecycleOwner` - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

`observer` - The Observer that will listen to the event and get triggered when it happens

`triggerEveryTime` - if true, the onChanged event of the observer will be triggered every time the observer is (re)created, as long as there is a value set. If false, it will only be triggered when the value is changed after the creation of the observer.

Since:

3.2

observeEvent

`@MainThread`

```
public void observeEvent(@NonNull
    androidx.lifecycle.LifecycleOwner lifecycleOwner,
    @NonNull
    androidx.lifecycle.Observer<java.lang.String> observer)
```

Use this method to get notified when a specific event is triggered from a button in a notification, In-App message or In-app content or from the main action of a notification. It replaces the broadcast of a specific value. It must be called on the main thread. Observing using this method will trigger the onChanged event of the observer only when the value changes after the observer is created. If there is already a value at that moment, the event will not get triggered. To change that behaviour, use the overload `observeEvent(LifecycleOwner, Observer, boolean)`

Parameters:

`lifecycleOwner` - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

`observer` - The Observer that will listen to the event and get triggered when it happens

observeEvent

`@MainThread`

```
public void observeEvent(@NonNull
    androidx.lifecycle.LifecycleOwner lifecycleOwner,
    @NonNull
    androidx.lifecycle.Observer<java.lang.String> observer,
    boolean triggerEveryTime)
```

Use this method to get notified when a specific event is triggered from a button in a notification, In-App message or In-app content or from the main action of a notification. It replaces the broadcast of a specific value. It must be called on the main thread.

Parameters:

`lifecycleOwner` - The LifecycleOwner that will be used to automatically start and stop listening. Most of the time, it will be the Activity you are in.

`observer` - The Observer that will listen to the event and get triggered when it happens

`triggerEveryTime` - if true, the onChanged event of the observer will be triggered every time the observer is (re)created, as long as there is a value set. If false, it will only be triggered when the value is changed after the creation of the observer.

Since:

3.2

Class SMSettings

```
java.lang.Object
    com.selligent.sdk.SMSettings
```

```
public class SMSettings
extends java.lang.Object
```

Configuration object passed to the 'start' method of SMManager.

Since:

1.0

Version:

3.9

See Also:

[SMManager.start\(SMSettings\)](#)

Field Summary

Fields

Modifier and Type	Field	Description
boolean	AddInAppMessageFromPushToInAppMessageList	If set to true, any In-App Message received through a push notification will be added to the list of In-App Messages retrieved by the SDK and will be available via SMManager.getInAppMessages(SMInAppMessageReturn) .
SMClearCache	ClearCacheIntervalValue	This value tells how often the SDK's inAppMessagesCache mechanism should clear itself.
java.lang.String	ClientId	The client id given by Selligent to contact the web services
boolean	ConfigureGeolocation	This is will tell the SDK to use the geolocation.
boolean	DoNotFetchTheToken	If set to true, the SDK will not try to retrieve the token used for push notification from Firebase.
boolean	DoNotListenToThePush	If set to true, the SDK will not listen for the push anymore.
boolean	EnableNotifications	If set to true, the notifications will be enabled as soon as the token is retrieved (this is the default behaviour).
java.lang.String	GoogleApplicationId	Deprecated. <i>You don't need to set this property as long as you use the JSON file given by Firebase</i>
SMInAppRefreshType	InAppContentRefreshType	This is will tell how often the SDK must get the In App content.
SMInAppRefreshType	InAppMessageRefreshType	This is will tell how often the SDK must get the In App messages.
boolean	LoadCacheAsynchronously	This will make the SDK load the content of its cache asynchronously.
java.lang.String	NotificationChannelDescription	Override this value to set a description of the channel It is visible in the notification channel setting screen of the device.
java.lang.String	NotificationChannelId	Override this value if you already have a channel setup in your app and you want the SDK to use it.
java.lang.String	NotificationChannelName	Override this value to set the name of the channel.
java.lang.String	PrivateKey	The private key given by Selligent to contact the web services
SMRemoteMessageDisplayType	RemoteMessageDisplayType	If set to Automatic, when the app is active, the remote push

		messages will automatically be displayed.
<code>SMThemeCategories</code>	<code>Theme</code>	Deprecated. <i>since 1.4, this value is not used anymore</i>
<code>java.lang.String</code>	<code>WebServiceUrl</code>	The URL of the Selligent web services

Constructor Summary

Constructors	
Constructor	Description
<code>SMSettings()</code>	

Method Summary

Methods inherited from class `java.lang.Object`

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

`GoogleApplicationId`

@Deprecated
 public java.lang.String GoogleApplicationId

Deprecated.

You don't need to set this property as long as you use the JSON file given by Firebase

The application id given by the Firebase Developer Console

`WebServiceUrl`

public java.lang.String WebServiceUrl

The URL of the Selligent web services

`ClientId`

public java.lang.String ClientId

The client id given by Selligent to contact the web services

`PrivateKey`

public java.lang.String PrivateKey

The private key given by Selligent to contact the web services

`Theme`

@Deprecated
 public SMThemeCategories Theme

Deprecated.

since 1.4, this value is not used anymore

The category of theme used (Holo, Material, AppCompat, DeviceDefault or Theme). This will allow the SDK to create the alert dialogs with a correct layout.

If set to DeviceDefault, the layout of the dialog will depend on the version of Android (Lollipop and above will have Material, others will have Holo).

Since:

1.2

ClearCacheIntervalValue

```
public SMClearCache ClearCacheIntervalValue
```

This value tells how often the SDK's inAppMessagesCache mechanism should clear itself. Internally, each notification-messages has a life span. Clearing the inAppMessagesCache stands for deleting notification messages with an expired life span. In other words, only old notification messages are deleted from the inAppMessagesCache. More recent ones are kept in memory until their life span expires and a new clearInAppMessageCache is called. By default, this value is set to Auto. Configuring this value depends how frequently the application will query specific notification messages. In other words, it depends how often you call the API `SMManager.displayMessage(String, Activity)`. In a nutshell: * If the application will never query `SMManager.displayMessage(String, Activity)`, we recommend keeping this value to default. * If the application uses the "In app messages" service, we recommend keeping this value to default. * On the other hand, if the application abuses `SMManager.displayMessage(String, Activity)`, we recommend selecting a value higher than the default one. As soon as IAM-service is enabled, the SDK will consider Week as being the default value. Except if you explicitly override the property. In 99% of the cases, you should not override this property as the SDK is smart enough to handle the inAppMessagesCache mechanism by itself.

Since:

1.3

InAppMessageRefreshType

```
public SMInAppRefreshType InAppMessageRefreshType
```

This will tell how often the SDK must get the In App messages. Setting this property WILL enable the In App messages in the SDK, even if the value is set to "None". If you do not want to enable the In App messages, leave it to null. If you do not set a value, you can still do it later by calling `SMManager.enableInAppMessages(SMInAppRefreshType)`

Since:

1.3

InAppContentRefreshType

```
public SMInAppRefreshType InAppContentRefreshType
```

This will tell how often the SDK must get the In App content. Setting this property WILL enable the In App content in the SDK, even if the value is set to "None". If you do not want to enable the In App content, leave it to null.

Since:

1.4

ConfigureGeolocation

```
public boolean ConfigureGeolocation
```

This will tell the SDK to use the geolocation. Default is false. If you didn't set "enableOnFirstRun" to false in the plotconfig.json file, geolocation will enable right away, asking for the permission if needed, monitoring geofences and displaying notifications. Otherwise, you will need to call `SMManager.enableGeolocation()`. NB: In the plotconfig.json file, if "enableOnFirstRun" or "automaticallyAskLocationPermission" is set to false, you have to ask for the location permission yourself. The default value for these settings is true.

Since:

1.7

RemoteMessageDisplayType

```
public SMRemoteMessageDisplayType RemoteMessageDisplayType
```

If set to Automatic, when the app is active, the remote push messages will automatically be displayed. If set to Notification, the user will have to click on the notification to display them. If set to None, nothing will happen, the app has to manage the display (cf.

```
SMManager.getLastRemotePushNotification(),
```

`SMManger.displayLastReceivedRemotePushNotification(android.app.Activity)`) Default value is Automatic.

Since:

1.3

See Also:

[SMManger](#)

LoadCacheAsynchronously

```
public boolean LoadCacheAsynchronously
```

This will make the SDK load the content of its cache asynchronously. It improves performance at startup as reading the files will be done in a separate thread. This has an impact on the management of In-App contents, so if you are already using them and want to set this to true, you might have to review your code (cf. document "Using the SDK" for more information about this). If you don't already use the In-App contents, feel free to set this to true. Default value is false.

Since:

2.1.0

DoNotFetchTheToken

```
public boolean DoNotFetchTheToken
```

If set to true, the SDK will not try to retrieve the token used for push notification from Firebase. Instead, you will be responsible to retrieve it and give it to the SDK by calling the method `SMManger.setFirebaseToken(java.lang.String)` every time the token changes. Use this if you already have your own way to retrieve the token and do not need the SDK to do it. Default value is false.

Since:

2.1.0

DoNotListenToThePush

```
public boolean DoNotListenToThePush
```

If set to true, the SDK will not listen for the push anymore. Instead, you will be responsible to listen and give it to the SDK by calling the method `SMManger.displayNotification(android.content.Context, android.content.Intent)` every time a push is received. Use this if you already have your own way to listen to the Firebase push and do not want the SDK to do it. Default value is false

Since:

2.1.0

EnableNotifications

```
public boolean EnableNotifications
```

If set to true, the notifications will be enabled as soon as the token is retrieved (this is the default behaviour). If set to false, you will have to call `SMManger.enableNotifications()` to allow notifications from Selligent to be sent to the device. Default value is true.

Since:

3.4.0

AddInAppMessageFromPushToInAppMessageList

```
public boolean AddInAppMessageFromPushToInAppMessageList
```

If set to true, any In-App Message received through a push notification will be added to the list of In-App Messages retrieved by the SDK and will be available via `SMManger.getInAppMessages(SMInAppMessageReturn)`.

Since:

3.6.0

WebViewNavigationOverride

```
public SMWebViewNavigationOverride WebViewNavigationOverride
```

Instantiate this if you want to override the navigation in the WebViews displayed by the In-App messages. It will allow you to check the link clicked by the user and decide how the SDK will handle (or not) the navigation.

Since:

3.6.0

See Also:

[SMWebViewNavigationOverride](#)

NotificationChannelId

```
@NonNull  
public java.lang.String NotificationChannelId
```

Override this value if you already have a channel setup in your app and you want the SDK to use it. If the channel is not found, it will be created. Default is "SMChannel001"

Since:

3.9.0

NotificationChannelName

```
@NonNull  
public java.lang.String NotificationChannelName
```

Override this value to set the name of the channel. It is visible in the notification setting screen of the device. If the channel already exists with a different name and this value is not the default one, the channel will be updated. Default is "SMDefaultChannel"

Since:

3.9.0

NotificationChannelDescription

```
@NonNull  
public java.lang.String NotificationChannelDescription
```

Override this value to set a description of the channel. It is visible in the notification channel setting screen of the device. If the channel already exists with a different description and this value is not the default one, the channel will be updated. Default is empty.

Since:

3.9.0

Constructor Detail

SMSettings

```
public SMSettings()
```

Package com.selligent.sdk

Enum SMClearCache

```
java.lang.Object  
    java.lang.Enum<SMClearCache>  
        com.selligent.sdk.SMClearCache
```

All Implemented Interfaces:

```
java.io.Serializable, java.lang.Comparable<SMClearCache>
```

```
public enum SMClearCache  
extends java.lang.Enum<SMClearCache>
```

Since:

1.3 The lifespan of the items in the cache

Version:

3.9

Enum Constant Summary

Enum Constants	
Enum Constant	Description
<code>Auto</code>	Default value, Selligent Mobile SDK manages the lifespan automatically.
<code>Day</code>	To clear the items that are one day old
<code>Month</code>	To clear the items that are one month old
<code>None</code>	To disable the cache mechanism.
<code>Quarter</code>	To clear the items that are one three months old
<code>Week</code>	To clear the items that are one week old

Method Summary

All Methods	Static Methods	Concrete Methods
Modifier and Type	Method	Description
<code>static SMClearCache</code>	<code>valueOf (java.lang.String name)</code>	Returns the enum constant of this type with the specified name.
<code>static SMClearCache[]</code>	<code>values()</code>	Returns an array containing the constants of this enum type, in the order they are declared.

Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

Auto

public static final SMClearCache Auto

Default value, Selligent Mobile SDK manages the lifespan automatically.

None

public static final SMClearCache None

To disable the cache mechanism.

Day

public static final SMClearCache Day

To clear the items that are one day old

Week

public static final SMClearCache Week

To clear the items that are one week old

Month

public static final SMClearCache Month

To clear the items that are one month old

Quarter

public static final SMClearCache Quarter

To clear the items that are one three months old

Method Detail

values

```
public static SMClearCache[] values()
```

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (SMClearCache c : SMClearCache.values())
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

```
public static SMClearCache valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

name - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

`java.lang.NullPointerException` - if the argument is null

Package com.sellgent.sdk

Enum SMContentType

java.lang.Object
 java.lang.Enum<SMContentType>
 com.sellgent.sdk.SMContentType

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<SMContentType>

```
public enum SMContentType
extends java.lang.Enum<SMContentType>
```

Enum listing the different types of In App Content

Since:

1.4

Version:

3.9

Enum Constant Summary

Enum Constants	
Enum Constant	Description
Html	
Image	
Url	

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description	
static SMContentType	fromInteger (int x)		
int	getValue()		
static SMContentType	valueOf (java.lang.String name)	Returns the enum constant of this type with the specified name.	
static SMContentType[]	values()	Returns an array containing the constants of this enum type, in the order they are declared.	

Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

Html

public static final SMContentType Html

Url

public static final SMContentType Url

Image

public static final SMContentType Image

Method Detail

values

public static SMContentType[] values()

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (SMContentType c : SMContentType.values())
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

public static SMContentType valueOf(java.lang.String name)

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used

to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

name - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

`java.lang.NullPointerException` - if the argument is null

getValue

```
public int getValue()
```

fromInteger

```
public static SMContentType fromInteger(int x)
```

Enum SMInAppContent.DisplayMode

```
java.lang.Object
  java.lang.Enum<SMInAppContent.DisplayMode>
    com.sellgent.sdk.SMInAppContent.DisplayMode
```

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<SMInAppContent.DisplayMode>

Enclosing class:

SMInAppContent

```
public static enum SMInAppContent.DisplayMode
extends java.lang.Enum<SMInAppContent.DisplayMode>
```

Enum listing the different display mode. OnlyOnce means the SMInAppContent will only be displayed once. UntilReplaced means the SMInAppContent will stay visible until a new one replaces it (or it expires).

Enum Constant Summary

Enum Constants	
Enum Constant	Description
OnlyOnce	
UntilReplaced	

Method Summary

All Methods	Static Methods	Concrete Methods	
Modifier and Type	Method	Description	
static SMInAppContent.DisplayMode	valueOf (java.lang.String name)	Returns the enum constant of this type with the specified name.	
static SMInAppContent.DisplayMode []	values ()	Returns an array containing the constants of this enum type, in the order they are declared.	

Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

OnlyOnce

public static final SMInAppContent.DisplayMode OnlyOnce

UntilReplaced

public static final SMInAppContent.DisplayMode UntilReplaced

Method Detail

values

public static SMInAppContent.DisplayMode[] values()

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (SMInAppContent.DisplayMode c : SMInAppContent.DisplayMode.values())
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

public static SMInAppContent.DisplayMode valueOf(java.lang.String name)

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

name - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

java.lang.IllegalArgumentException - if this enum type has no constant with the

specified name

java.lang.NullPointerException - if the argument is null

Package com.sellgent.sdk

Enum SMInAppRefreshType

```
java.lang.Object
  java.lang.Enum<SMInAppRefreshType>
    com.sellgent.sdk.SMInAppRefreshType
```

All Implemented Interfaces:

```
java.io.Serializable, java.lang.Comparable<SMInAppRefreshType>
```

```
public enum SMInAppRefreshType
extends java.lang.Enum<SMInAppRefreshType>
```

Enum with the different values for the refresh of the In App messages and In Ap contents. Messages/contents will be retrieved when the application becomes active (if In App messages/contents are enabled) if the last fetch was older than the refresh type. Minutely is there for testing purposes ONLY. We do NOT recommend using it in production.

Since:

1.3

Version:

3.9

Enum Constant Summary

Enum Constants

Enum Constant	Description
Daily	
Hourly	
Minutely	
None	

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
static SMInAppRefreshType	valueOf (java.lang.String name)	Returns the enum constant of this type with the specified name.
static SMInAppRefreshType[]	values()	Returns an array containing the constants of this enum type, in the order they are declared.

Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

None

public static final SMInAppRefreshType None

Minutely

public static final SMInAppRefreshType Minutely

Hourly

public static final SMInAppRefreshType Hourly

Daily

public static final SMInAppRefreshType Daily

Method Detail

values

public static SMInAppRefreshType[] values()

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (SMInAppRefreshType c : SMInAppRefreshType.values())
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

```
public static SMInAppRefreshType valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

name - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

`java.lang.NullPointerException` - if the argument is null

Package com.selligent.sdk

Enum SMM MessageType

java.lang.Object
 java.lang.Enum<SMM MessageType>
 com.selligent.sdk.SMM MessageType

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<SMM MessageType>

```
public enum SMM MessageType
extends java.lang.Enum<SMM MessageType>
```

Possible values for an In-app message type

Since:

3.5.1

Version:

3.9

Enum Constant Summary

Enum Constants	Description
Alert	
Hidden	
Html	
Image	
Map	
Passbook	
Undefined	
Url	

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description	
static SMM MessageType	fromInteger (int x)		

int	getValue()	
static SMM MessageType	valueOf (java.lang.String name)	Returns the enum constant of this type with the specified name.
static SMM MessageType []	values ()	Returns an array containing the constants of this enum type, in the order they are declared.

Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

Undefined

public static final SMM MessageType Undefined

Hidden

public static final SMM MessageType Hidden

Alert

public static final SMM MessageType Alert

Html

public static final SMM MessageType Html

Url

public static final SMM MessageType Url

Image

public static final SMM MessageType Image

Map

```
public static final SMM MessageType Map
```

Passbook

```
public static final SMM MessageType Passbook
```

Method Detail

values

```
public static SMM MessageType[] values()
```

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (SMM MessageType c : SMM MessageType.values())
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

```
public static SMM MessageType valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

name - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

`java.lang.NullPointerException` - if the argument is null

getValue

```
public int getValue()
```

fromInteger

```
public static SMM MessageType fromInteger(int x)
```

Package com.selligent.sdk

Enum SMRemoteMessageDisplayType

java.lang.Object
 java.lang.Enum<SMRemoteMessageDisplayType>
 com.selligent.sdk.SMRemoteMessageDisplayType

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<SMRemoteMessageDisplayType>

```
public enum SMRemoteMessageDisplayType
extends java.lang.Enum<SMRemoteMessageDisplayType>
```

List the different possibilities to display a remote message received while the app is in foreground - Automatic: the message is automatically displayed - Notification: a notification is created, the user needs to click on it to display the message - None: nothing is done, the message is not displayed, it is up to the app to display it (cf.

SMManager.getLastRemotePushNotification(),

SMManager.displayLastReceivedRemotePushNotification(android.app.Activity))

Since:

1.3

Version:

3.9

Enum Constant Summary

Enum Constants

Enum Constant	Description
Automatic	
None	
Notification	

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
static SMRemoteMessageDisplayType	valueOf (java.lang.String name)	Returns the enum constant of this type with the specified name.
static SMRemoteMessageDisplayType []	values ()	Returns an array containing the constants of this enum

type, in the order they are declared.

Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

Automatic

public static final SMRemoteMessageDisplayType Automatic

Notification

public static final SMRemoteMessageDisplayType Notification

None

public static final SMRemoteMessageDisplayType None

Method Detail

values

public static SMRemoteMessageDisplayType[] values()

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (SMRemoteMessageDisplayType c : SMRemoteMessageDisplayType.values())
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

```
public static SMRemoteMessageDisplayType valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

name - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

java.lang.IllegalArgumentException - if this enum type has no constant with the specified name

java.lang.NullPointerException - if the argument is null

Package com.selligent.sdk

Enum SMThemeCategories

```
java.lang.Object
  java.lang.Enum<SMThemeCategories>
    com.selligent.sdk.SMThemeCategories
```

All Implemented Interfaces:

```
java.io.Serializable, java.lang.Comparable<SMThemeCategories>
```

@Deprecated

```
public enum SMThemeCategories
extends java.lang.Enum<SMThemeCategories>
```

Deprecated.

Since:

1.2

Version:

3.9

Enum Constant Summary

Enum Constants

Enum Constant	Description
AppCompat	Deprecated.
DeviceDefault	Deprecated.
Holo	Deprecated.
Material	Deprecated.
Theme	Deprecated.

Method Summary

All Methods

Static Methods

Concrete Methods

Deprecated Methods

Modifier and Type	Method	Description
-------------------	--------	-------------

<code>static SMThemeCategories</code>	<code>valueOf (java.lang.String name)</code>	Deprecated. Returns the enum constant of this type with the specified name.
<code>static SMThemeCategories[]</code>	<code>values()</code>	Deprecated. Returns an array containing the constants of this enum type, in the order they are declared.

Methods inherited from class java.lang.Enum

`compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf`

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, wait, wait, wait`

Enum Constant Detail

Theme

`public static final SMThemeCategories Theme`

Deprecated.

DeviceDefault

`public static final SMThemeCategories DeviceDefault`

Deprecated.

Holo

`public static final SMThemeCategories Holo`

Deprecated.

Material

`public static final SMThemeCategories Material`

Deprecated.

AppCompat

`public static final SMThemeCategories AppCompat`

Deprecated.

Method Detail

values

```
public static SMThemeCategories[] values()
```

Deprecated.

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (SMThemeCategories c : SMThemeCategories.values())
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

```
public static SMThemeCategories valueOf(java.lang.String name)
```

Deprecated.

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

name - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

`java.lang.NullPointerException` - if the argument is null

Enum SMWebViewNavigationOption

```
java.lang.Object
  java.lang.Enum<SMWebViewNavigationOption>
    com.selligent.sdk.SMWebViewNavigationOption
```

All Implemented Interfaces:

```
java.io.Serializable, java.lang.Comparable<SMWebViewNavigationOption>
```

```
public enum SMWebViewNavigationOption
extends java.lang.Enum<SMWebViewNavigationOption>
```

Enum used by `SMWebViewNavigationOverride.shouldHandleURL(Context, String)` to tell the SDK how to handle the navigation: - `ResumeNavigation`: the navigation will continue in the WebView like usual. - `StopNavigation`: the navigation will stop and the Activity displaying the WebView will close. Use this if you want to handle the navigation yourself. - `StopNavigationAndExecuteDeeplink`: the navigation will stop, the Activity displaying the WebView will close and the SDK will start another Activity to execute the deeplink.

Since:

3.6.0

Version:

3.9

Enum Constant Summary

Enum Constants

Enum Constant	Description
<code>ResumeNavigation</code>	
<code>StopNavigation</code>	
<code>StopNavigationAndExecuteDeeplink</code>	

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
<code>static SMWebViewNavigationOption</code>	<code>valueOf (java.lang.String name)</code>	Returns the enum constant of this type with the specified name.
<code>static SMWebViewNavigationOption[]</code>	<code>values ()</code>	Returns an array containing the constants of this enum

type, in the order they are declared.

Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

ResumeNavigation

public static final SMWebViewNavigationOption ResumeNavigation

StopNavigation

public static final SMWebViewNavigationOption StopNavigation

StopNavigationAndExecuteDeepLink

public static final SMWebViewNavigationOption StopNavigationAndExecuteDeepLink

Method Detail

values

public static SMWebViewNavigationOption[] values()

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (SMWebViewNavigationOption c : SMWebViewNavigationOption.values())
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

```
public static SMWebViewNavigationOption valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

name - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

java.lang.IllegalArgumentException - if this enum type has no constant with the specified name

java.lang.NullPointerException - if the argument is null

Package com.selligent.sdk

Interface Summary

Interface	Description
OnSMNotificationMessageRetrieved	This allows to write codes that will be executed after an event is sent to the Selligent platform
SMCallback	This allows to write codes that will be executed after an event is sent to the Selligent platform
SMinAppContentReturn	Interface used to implement the code that will be executed after retrieving the In-App contents
SMinAppMessageDisplay	Implement this interface if you do not want the SDK to display the In-App Message linked to a push notification and do it yourself
SMinAppMessageReturn	Interface used to implement the code that will be executed after retrieving the In-App contents
SMNotificationCallback	Deprecated. <i>Since 1.3, a broadcast is performed at the click on a button (cf.</i>
SMWebViewNavigationOverride	Implement this interface if you want to override the navigation in the WebViews displayed by the In-App messages.

Class Summary

Class	Description
SMAplication	The Application class of the SDK from which the app custom Application should be extended.
SMDeviceInfos	This class is a set of device properties that can be sent to the platform.
SMEvent	Object used to send a custom event to the Selligent platform with SMManager.sendEvent.
SMEventUserLogin	Object used to send a "login" event to the Selligent platform with SMManager.sendEvent.
SMEventUserLogout	Object used to send a "logout" event to the Selligent platform with SMManager.sendEvent.
SMEventUserRegister	Object used to send a "register" event to the Selligent platform with SMManager.sendEvent.
SMEventUserUnregister	Object used to send an "unregister" event to the Selligent platform with

	SMManager.sendEvent.
SMForegroundGcmBroadcastReceiver	Class implementing the receiver that will listen to connectivity changes.
SIMInAppContent	An In App content
SIMInAppContentHtmlFragment	This class implements a fragment that will display one or several HTML contents.
SIMInAppContentImageFragment	This class implements a fragment that will display In App Content containing an image.
SIMInAppContentUrlFragment	This class implements a fragment that will display In App Content containing an url.
SIMInAppMessage	An In App message
SIMLink	A link of an SIMInAppContent .
SMManager	Singleton object used to interact with the Selligent Mobile SDK.
SIMMapMarker	This class represents a point on a map.
SIMNotificationButton	Object containing all the data used to implement a button in a notification/message.
SIMNotificationMessage	A notification message.
SIMObserverManager	This class is used to observe some events of the SDK.
SIMSettings	Configuration object passed to the 'start' method of SMManager.

Enum Summary

Enum	Description
SIMClearCache	
SIMContentType	Enum listing the different types of In App Content
SIMInAppContent.DisplayMode	Enum listing the different display mode.
SIMInAppRefreshType	Enum with the different values for the refresh of the In App messages and In App contents.
SIMMessageType	Possible values for an In-app message type
SIMRemoteMessageDisplayType	List the different possibilities to display a remote message received while the app is in foreground - Automatic: the message is automatically displayed - Notification: a notification is created, the user needs to click on it to display the message - None: nothing is done, the message is not displayed, it is up to the app to display it (cf.

SMThemeCategories	Deprecated.
SMWebViewNavigationOption	Enum used by <code>shouldHandleURL(Context, String)</code> to tell the SDK how to handle the navigation: - ResumeNavigation: the navigation will continue in the WebView like usual.

Serialized Form

Package com.selligent.sdk

Class *com.selligent.sdk.SMEvent* extends *java.lang.Object* implements *Serializable*

serialVersionUID: 1L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
           java.lang.ClassNotFoundException
```

This method is called when deserializing the object. It should not be called manually.

Throws:

```
java.io.IOException
java.lang.ClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

This method is called when serializing the object. It should not be called manually.

Throws:

```
java.io.IOException
```

Class *com.selligent.sdk.SMEventUserLogin* extends *com.selligent.sdk.SMEventUser* implements *Serializable*

serialVersionUID: 1L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
```

```
throws java.io.IOException,  
       java.lang.ClassNotFoundException
```

Throws:

```
java.io.IOException
```

```
java.lang.ClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)  
    throws java.io.IOException
```

Throws:

```
java.io.IOException
```

Class *com.selligent.sdk.SMEventUserLogout* extends *com.selligent.sdk.SMEventUser* implements Serializable

serialVersionUID: 1L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)  
    throws java.io.IOException,  
           java.lang.ClassNotFoundException
```

Throws:

```
java.io.IOException
```

```
java.lang.ClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)  
    throws java.io.IOException
```

Throws:

```
java.io.IOException
```

Class *com.selligent.sdk.SMEventUserRegister* extends *com.selligent.sdk.SMEventUser*

implements Serializable

serialVersionUID: 1L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

Throws:

```
java.io.IOException
java.lang.ClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

Throws:

```
java.io.IOException
```

Class *com.selligent.sdk.SMEventUserUnregister* extends *com.selligent.sdk.SMEventUser*
implements Serializable

serialVersionUID: 1L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

Throws:

```
java.io.IOException
java.lang.ClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
```

```
throws java.io.IOException
```

Throws:

```
java.io.IOException
```

Class *com.selligent.sdk.SMInAppContent* extends *com.selligent.sdk.BaseMessage* implements *Serializable*

serialVersionUID: 1L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

This method is called when deserializing the object. It should not be called manually.

Throws:

```
java.io.IOException
```

```
java.lang.ClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

This method is called when serializing the object. It should not be called manually.

Throws:

```
java.io.IOException
```

Class *com.selligent.sdk.SMInAppMessage* extends *com.selligent.sdk.InternalInAppMessage* implements *Serializable*

serialVersionUID: 3L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

This method is called when deserializing the object. It should not be called manually.

Throws:

```
java.io.IOException
java.lang.ClassNotFoundException
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

This method is called when serializing the object. It should not be called manually.

Throws:

```
java.io.IOException
```

Class *com.selligent.sdk.SMLink* extends *SMNotificationButton* implements *Serializable*

serialVersionUID: 1L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

This method is called when deserializing the object. It should not be called manually.

Throws:

```
java.io.IOException
java.lang.ClassNotFoundException
```

Since:

1.3

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

This method is called when serializing the object. It should not be called manually.

Throws:

java.io.IOException

Since:

1.3

Class *com.sellgent.sdk.SMMarker* extends java.lang.Object implements Serializable

serialVersionUID: 4L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
        java.lang.ClassNotFoundException
```

Throws:

java.io.IOException

java.lang.ClassNotFoundException

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

Throws:

java.io.IOException

Class *com.sellgent.sdk.SMNotificationButton* extends java.lang.Object implements Serializable

serialVersionUID: 2L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
    throws java.io.IOException,
```

```
java.lang.ClassNotFoundException
```

This method is called when deserializing the object. It should not be called manually.

Throws:

```
java.io.IOException
```

```
java.lang.ClassNotFoundException
```

Since:

```
1.3
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

This method is called when serializing the object. It should not be called manually.

Throws:

```
java.io.IOException
```

Since:

```
1.3
```

**Class *com.sellgent.sdk.SMNotificationMessage* extends
com.sellgent.sdk.NotificationMessage implements Serializable**

serialVersionUID: 1L

Serialization Methods

readExternal

```
public void readExternal(java.io.ObjectInput serializedObject)
```

writeExternal

```
public void writeExternal(java.io.ObjectOutput serializedObject)
    throws java.io.IOException
```

Throws:

```
java.io.IOException
```

Constant Field Values

Contents

com.selligent.*

com.selligent.*

Modifier and Type	Constant Field	Value
public static final java.lang.String	BROADCAST_DATA_BUTTON	"SMDaDataButton"
public static final java.lang.String	BROADCAST_DATA_GCM_TOKEN	"SMDaDataGCMToken"
public static final java.lang.String	BROADCAST_DATA_IN_APP_CONTENTS	"SMDaDataInAppContents"
public static final java.lang.String	BROADCAST_DATA_IN_APP_MESSAGES	"SMDaDataInAppMessages"
public static final java.lang.String	BROADCAST_EVENT_BUTTON_CLICKED	"SMEventButtonClicked"
public static final java.lang.String	BROADCAST_EVENT_RECEIVED_GCM_TOKEN	"SMReceivedGCMToken"
public static final java.lang.String	BROADCAST_EVENT_RECEIVED_IN_APP_CONTENTS	"SMReceivedInAppContent"
public static final java.lang.String	BROADCAST_EVENT_RECEIVED_IN_APP_MESSAGE	"SMReceivedInAppMessage"
public static final java.lang.String	BROADCAST_EVENT_RECEIVED_REMOTE_NOTIFICATION	"SMReceivedRemoteNotification"
public static final java.lang.String	BROADCAST_EVENT_WILL_DISMISS_NOTIFICATION	"SMEventWillDismissNotification"
public static final java.lang.String	BROADCAST_EVENT_WILL_DISPLAY_NOTIFICATION	"SMEventWillDisplayNotification"
public static final java.lang.String	VERSION_LIB	"3.9.0"